AIAA Aviation
16-20 June 2014, Atlanta, GA
15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

An Unsteady Continuous Adjoint Approach for Aerodynamic Design on Dynamic Meshes

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

This article 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 verify 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 pitching airfoil and wing test cases.

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_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
$\bar{\bar{I}}$	Identity matrix
J	Cost function defined as an integral over S

*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.

$\mathcal{J}$	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, $\mu^2_{tot} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t}$
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

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..

Computational cost is paramount for design in unsteady flows. Due to the increased cost of time-accurate calculations, efficient methods for computing sensitivity information are a must. The adjoint approach to sensitivity analysis[1–3] is 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. Solving the governing equations in arbitrary Lagrangian-

Eulerian (ALE) form addresses the 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[4] 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[5] 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[6,7] formulated an unsteady, discrete adjoint for turbulent flows on dynamically deforming, unstructured meshes in both two and three dimensions. More recently, Nielsen et al.[8,9] have demonstrated an unsteady, discrete adjoint approach for design in turbulent flows on dynamic, possibly overset, deforming meshes. Economon et al.[10] 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[11] 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.[12] with the Spalart-Allmaras turbulence model on unstructured meshes and by Dumont et al.[13] with the $k-\omega$ turbulence model and the shear stress transport correction on structured meshes. Economon et al.[14,15] have shown both inviscid and viscous continuous adjoint formulations in a rotating frame.

Another complementary approach for unsteady problems with inherent time-periodicity is the time-spectral method,[16] or similarly, non-linear frequency domain methods.[17,18] These approaches allow for the solution of a periodic steady state directly by introducing the periodicity explicitly in the discretization of the time derivative term of the flow equations. By trading memory cost for calculation time, these spectral methods greatly reduce the time-to-solution for achieving a periodic steady state. Adjoint approaches for these periodic methods have been effectively employed for design, including the design of helicopter rotors.[19–21]

The majority 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 algorithmic differentiation is available, this article 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 article presents details on the derivation of the unsteady continuous adjoint equations, their admissible boundary conditions, and the expressions for surface sensitivity. 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.

From the general scenario of viscous, unsteady flow, the corresponding adjoint formulations for inviscid, rotating frame, or even steady problems can be easily recovered from the general framework. Moreover, as the unsteady continuous adjoint equations are a system of partial differential equations (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.

The key contributions of this article are centered around the detailed derivation, implementation, and

application of the unsteady continuous adjoint formulation for aerodynamic design on dynamic meshes. More specifically and to the author's knowledge, the methodology is the first continuous adjoint surface formulation based on shape calculus for the unsteady, compressible RANS equations in ALE form with a generic source term. 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 unsteady continuous adjoint methodology and shape design framework are demonstrated through two separate test cases in 2D and 3D.

The paper is organized as follows. Section II briefly overviews the unsteady, compressible RANS equations, including the accompanying boundary conditions and turbulence modeling. Section III contains a derivation of the unsteady continuous adjoint formulation for computing surface sensitivities. Section IV 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. Section V presents results for two- and three-dimensional optimal shape design demonstrations using the NACA 64A010 airfoil and ONERA M6 wing as baseline geometries. Lastly, Section VI summarizes the main conclusions of the article.

II. Physical Problem Description

Consider an aerodynamic body or surface S immersed in a fluid represented by a domain Ω as shown in Fig. 1. Throughout the domain in both space and time, the behavior of the fluid is physically modeled by a particular set of governing 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, or $S = S(t)$ and $\Omega = \Omega(t)$.

We are concerned with time-accurate, viscous flow around aerodynamic bodies in arbitrary motion governed by the compressible, unsteady Navier-Stokes equations, which are statements of conservation for mass, momentum, and energy in the fluid. The equations are expressed 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, such as a wing or a full aircraft configuration. These conservation equations along with a generic source term $\mathcal{Q}$ can be expressed in an arbitrary Lagrangian-Eulerian (ALE)[22] differential form as

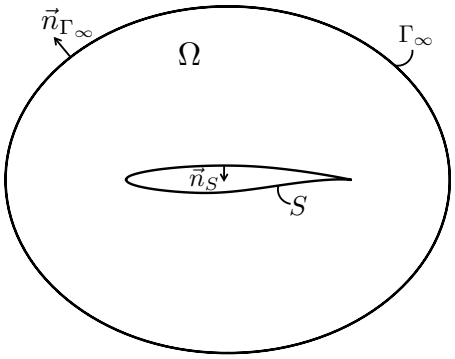


Figure 1. Notional schematic of the flow domain, Ω, the boundaries, Γ_∞ and S, as well as the definition of the surface normals.

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

where the conservative variables are given by

$$U = \begin{Bmatrix} \rho \\ \rho\vec{v} \\ \rho E \end{Bmatrix}, \tag{2}$$

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

$$\vec{F}^c_{ale} = \begin{Bmatrix} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho\vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{\bar{I}}p \\ \rho E(\vec{v} - \vec{u}_\Omega) + p\vec{v} \end{Bmatrix}, \quad \vec{F}^{v1} = \begin{Bmatrix} \cdot \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \vec{v} \end{Bmatrix}, \quad \vec{F}^{v2} = \begin{Bmatrix} \cdot \\ \cdot \\ c_p \nabla T \end{Bmatrix}, \quad \mathcal{Q} = \begin{Bmatrix} q_\rho \\ \vec{q}_{\rho\vec{v}} \\ q_{\rho E} \end{Bmatrix}, \tag{3}$$

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

$$\bar{\bar{\tau}} = \nabla\vec{v} + \nabla\vec{v}^\mathsf{T} - \frac{2}{3}\bar{\bar{I}}(\nabla \cdot \vec{v}). \tag{4}$$

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

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}(\vec{v} \cdot \vec{v})\right], \tag{5}$$

the temperature is given by

$$T = \frac{p}{\rho R}, \tag{6}$$

and

$$c_p = \frac{\gamma R}{(\gamma - 1)}. \tag{7}$$

The second line in equation system (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.[23] Here, W represents the characteristic variables. For problems on fixed grids (i.e., $\vec{u}_\Omega = 0$), the system in (1) reduces to a purely Eulerian formulation.

For unsteady problems, the temporal conditions will be problem dependent, and in this article, 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, which removes transient effects by reaching a periodic steady state.

A. Turbulence Modeling

We are also concerned with obtaining solutions of the unsteady Reynolds-averaged Navier-Stokes equations, which will require the inclusion of a suitable turbulence model. In accord with the standard approach to turbulence modeling based upon the Boussinesq hypothesis,[24] 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[25] 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

$$\mu^1_{tot} = \mu_{dyn} + \mu_{tur}, \qquad \mu^2_{tot} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t}, \tag{8}$$

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

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., $\mu_{tur} = \mu_{tur}(U, \hat{\nu})$. We assume that $\hat{\nu}$ is a single scalar variable obtained from a one-equation turbulence model. The Spalart-Allmaras (S-A) model[26] is one of the most common and widely used turbulence models for the analysis and design of engineering applications affected by turbulent flows, especially applications in external aerodynamics. The S-A model is used for all of the turbulent calculations in this article.

III. Surface Sensitivities Via a Time-Accurate Continuous Adjoint

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. Initially, we will concentrate on a generic functional defined as a time-averaged, integrated quantity on the solid surface which 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.

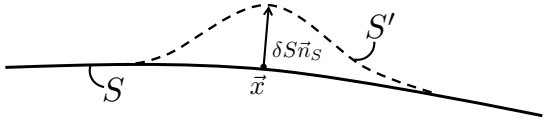


Figure 2. An infinitesimal shape deformation in the local surface normal direction.

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:

$$\min_S J(S) = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, T, \partial_n T, \vec{n})\, ds\, dt$$
$$\text{subject to: } \mathcal{R}(U) = 0 \tag{9}$$

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 vector normal to the surface S. In this work, S is assumed to be continuously differentiable (C^1), and the local shape perturbations applied to S can be described by

$$S' = \{\vec{x} + \delta S(\vec{x})\, \vec{n}(\vec{x}) : \ \vec{x} \in S\}, \tag{10}$$

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. 2.

The minimization of Eqn. 9 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.

A. 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, which gives

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\delta S} j(\vec{f}, T, \partial_n T, \vec{n})\, ds\, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \delta j(\vec{f}, T, \partial_n T, \vec{n})\, ds\, dt, \tag{11}$$

where for conciseness in notation, we use $\int_{\delta S}(\cdot)\, ds = \int_{S'}(\cdot)\, ds - \int_S(\cdot)\, ds$ 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. 11 can be 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[27] 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, the first term on the right hand side of Eqn. 11 becomes:

$$\begin{aligned} \int_{\delta S} j(\vec{f}, T, \partial_n T, \vec{n})\ ds &= \int_S (\partial_n j - 2H_m j)\, \delta S\ ds \\ &= \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T}\, \partial_n T + \frac{\partial j}{\partial (\partial_n T)}\, \partial_n^2 T - 2H_m j \right) \delta S\ ds, \end{aligned} \tag{12}$$

where from the chain rule and our functional definition,

$$\partial_n j = \vec{n} \cdot \nabla j(\vec{f}, T, \partial_n T, \vec{n}) \quad = \quad \frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T}\, \partial_n T + \frac{\partial j}{\partial (\partial_n T)}\, \partial_n^2\, T. \tag{13}$$

The second term on the right hand side of Eqn. 11 can also be further manipulated by focusing on δj:

$$\delta j(\vec{f}, T, \partial_n T, \vec{n}) = \frac{\partial j}{\partial \vec{f}} \cdot \delta \vec{f} + \frac{\partial j}{\partial T}\, \delta T + \frac{\partial j}{\partial (\partial_n T)}\, \delta(\partial_n T) - \frac{\partial j}{\partial \vec{n}} \cdot \nabla_S(\delta S) \tag{14}$$

where we have used $\delta \vec{n} = -\nabla_S(\delta S)$, which holds for small deformations.[28] Here, ∇_S represents the tangential gradient operator on S. Combining results from Eqns. 12 and 14 and introducing them into Eqn. 11 gives an intermediate expression for the variation of the functional:

$$\begin{aligned} \delta J = & \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T}\, \partial_n T + \frac{\partial j}{\partial (\partial_n T)}\, \partial_n^2\, T - 2 H_m j \right) \delta S\, ds\, dt \\ & + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \delta \vec{f} + \frac{\partial j}{\partial T}\, \delta T + \frac{\partial j}{\partial (\partial_n T)}\, \delta(\partial_n T) - \frac{\partial j}{\partial \vec{n}} \cdot \nabla_S(\delta S) \right) ds\, dt. \end{aligned} \tag{15}$$

While other options are possible, we will focus on a force-based objective function that depends only on $\vec{f}$ in the following way

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

such that

$$\frac{\partial j}{\partial \vec{f}} = \vec{d}, \quad \frac{\partial j}{\partial T} = 0, \quad \frac{\partial j}{\partial (\partial_n T)} = 0, \quad \frac{\partial j}{\partial \vec{n}} = \vec{0}, \tag{17}$$

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

$$\vec{d} = \begin{cases} \left(\frac{1}{C_\infty}\right)(\cos\alpha\ \cos\beta, \sin\alpha\ \cos\beta, \sin\beta), & C_D & \text{Drag}, \\ \left(\frac{1}{C_\infty}\right)(-\sin\alpha, \cos\alpha, 0), & C_L & \text{Lift}, \\ \left(\frac{1}{C_\infty}\right)(-\sin\beta\ \cos\alpha, -\sin\beta\ \sin\alpha, \cos\beta), & C_{SF} & \text{Side-force}, \\ \left(\frac{1}{C_\infty C_D}\right)(-\sin\alpha - \frac{C_L}{C_D}\cos\alpha\cos\beta, -\frac{C_L}{C_D}\sin\beta, \cos\alpha - \frac{C_L}{C_D}\sin\alpha\cos\beta), & \frac{C_L}{C_D} & \text{L/D}, \\ \left(\frac{1}{C_\infty}\right)(0, 0, 1), & C_{f_z} & \text{Z-Force}, \\ \left(\frac{1}{C_\infty L_{ref}}\right)(-(y - y_o), (x - x_o), 0), & C_{m_z} & \text{Z-Moment}, \end{cases} \tag{18}$$

where $C_\infty = \frac{1}{2} v_\infty^2 \rho_\infty A_z$, v_∞ is the freestream velocity, ρ_∞ is the freestream density, L_{ref} is a reference length for computing moments, A_z is the reference area, α is the freestream 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 normal surface 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.

After choosing a force-based functional and imposing the relationships in Eqn. 17 above, the variation of the functional in Eqn. 15 is simplified to

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\vec{d} \cdot \partial_n \vec{f} - 2 H_m (\vec{d} \cdot \vec{f}) \right) \delta S\, ds\, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot \delta \vec{f}\, ds\, dt, \tag{19}$$

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

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 $\vec{f} = (\bar{\bar{I}} p - \bar{\bar{\sigma}}) \cdot \vec{n}$, and therefore,

$$\delta \vec{f} = \delta \left[(\bar{\bar{I}} p - \bar{\bar{\sigma}}) \cdot \vec{n} \right] = (\bar{\bar{I}} \delta p - \delta \bar{\bar{\sigma}}) \cdot \vec{n} - (\bar{\bar{I}} p - \bar{\bar{\sigma}}) \cdot \nabla_S(\delta S), \tag{20}$$

where we have again used $\delta\vec{n} = -\nabla_S(\delta S)$. By introducing Eqn. 20, Eqn. 19 can be rearranged as

$$\delta J = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}\delta p - \delta\bar{\bar{\sigma}})\cdot\vec{n}\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left[\vec{d}\cdot\partial_n\vec{f} - 2H_m(\vec{f}\cdot\vec{d})\right]\delta S\,ds\,dt - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\nabla_S(\delta S)\,ds\,dt. \tag{21}$$

The final term of Eqn. 21 can be integrated by parts to give

$$\begin{aligned}\delta J &= \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}\delta p - \delta\bar{\bar{\sigma}})\cdot\vec{n}\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left[\vec{d}\cdot\partial_n\vec{f} - 2H_m(\vec{f}\cdot\vec{d})\right]\delta S\,ds\,dt \\ &\quad - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\delta S\right]\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right]\delta S\,ds\,dt \\ &= \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}\delta p - \delta\bar{\bar{\sigma}})\cdot\vec{n}\,ds\,dt \\ &\quad + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left\{\nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] + \vec{d}\cdot\partial_n\vec{f} - 2H_m(\vec{f}\cdot\vec{d})\right\}\delta S\,ds\,dt,\end{aligned} \tag{22}$$

where we have rearranged and used the identity $\int_S \nabla_S\cdot(\cdot)\,ds = 0$ on a closed surface in going from the first to second lines. Focusing now on the braced portion of the integrand in the final term in Eqn. 22, further simplifications can be made:

$$\begin{aligned}&\nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] + \vec{d}\cdot\partial_n\vec{f} - 2H_m(\vec{f}\cdot\vec{d}) \\ &= \nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] + \partial_n(\vec{d}\cdot\vec{f}) - 2H_m(\vec{d}\cdot\vec{f}) - \vec{f}\cdot\partial_n\vec{d} \\ &= \nabla_S\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] + \partial_n\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\right] - 2H_m\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\right] - (\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\cdot\partial_n\vec{d} \\ &= \nabla\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] - (\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\cdot\partial_n\vec{d}\end{aligned} \tag{23}$$

, where we have used the product rule in going from the first to second lines and the divergence expressed on the surface $\nabla\cdot\vec{q} = \nabla_S\cdot\vec{q} + \partial_n(\vec{q}\cdot\vec{n}) - 2H_m(\vec{q}\cdot\vec{n})$ with $\vec{q}$ being an arbitrary vector that, in this case, is described by $\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})$. Substituting the result of Eqn. 23 back into Eqn. 22 gives a near final expression for the variation of the functional,

$$\delta J = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}\delta p - \delta\bar{\bar{\sigma}})\cdot\vec{n}\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left\{\nabla\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] - (\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\cdot\partial_n\vec{d}\right\}\delta S\,ds\,dt. \tag{24}$$

Lastly, the final term in Eqn. 24 can be simplified for easier calculation (and to remove higher-order derivatives) by using information from the Navier-Stokes equations. Expanding the divergence term gives,

$$\begin{aligned}\nabla\cdot\left[\vec{d}\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}})\right] &= \nabla\vec{d}:(\bar{\bar{I}}p - \bar{\bar{\sigma}}) + \vec{d}\cdot\nabla\cdot(\bar{\bar{I}}p - \bar{\bar{\sigma}}) \\ &= \nabla\vec{d}:(\bar{\bar{I}}p - \bar{\bar{\sigma}}) + \vec{d}\cdot(\nabla p - \nabla\cdot\bar{\bar{\sigma}}) \\ &= \nabla\vec{d}:(\bar{\bar{I}}p - \bar{\bar{\sigma}}) + \vec{d}\cdot\left[\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})\right],\end{aligned} \tag{25}$$

where in going from the third to fourth line of Eqn. 25, 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

$$\begin{aligned}\delta J &= \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(\bar{\bar{I}}\delta p - \delta\bar{\bar{\sigma}})\cdot\vec{n}\,ds\,dt \\ &\quad + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left\{\vec{d}\cdot\left[\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})\right] + \nabla\vec{d}:(\bar{\bar{I}}p - \bar{\bar{\sigma}}) - (\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\cdot\partial_n\vec{d}\right\}\delta S\,ds\,dt.\end{aligned} \tag{26}$$

Note that, for a steady problem without source terms and with a constant force projection vector, the second integral in Eqn. 26 vanishes.

B. The Adjoint Approach to Optimal Design

Following the adjoint approach to optimal design, Eqn. 9 can be transformed into an unconstrained optimization problem by including the inner product of an adjoint state vector $\Psi = \Psi(\vec{x}, t)$ and the governing equations integrated over the domain (space and time) in order to form the Lagrangian:

$$\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, T, \partial_n T, \vec{n})\, ds\, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \mathcal{R}(U)\, d\Omega\, dt, \tag{27}$$

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

$$\Psi = \left\{ \begin{array}{c} \psi_\rho \\ \psi_{\rho v_1} \\ \psi_{\rho v_2} \\ \psi_{\rho v_3} \\ \psi_{\rho E} \end{array} \right\} = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}. \tag{28}$$

Note that, because the flow equations must be satisfied in the domain, or $\mathcal{R}(U) = 0$, the original functional in Eqn. 9 and the Lagrangian in Eqn. 27 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. 27:

$$\delta\mathcal{J} = \delta J - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \delta\mathcal{R}(U)\, d\Omega\, dt, \tag{29}$$

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

C. The Linearized Navier-Stokes Equations

The second term on the right hand side of Eqn. 29 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 δU as well as the gradient of the solution $\delta(\nabla U)$. To complete the linearized system of equations, the boundary conditions corresponding to the original governing system must also be linearized.

In this work, we will 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 to reduce the complexity inherent in including sensitivity information for the viscosity, which may require the treatment of a turbulence model. 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,

$$\left\{ \begin{array}{ll} \delta\mathcal{R}(U) = \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U - \nabla \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta(\nabla U) - \frac{\partial \mathcal{Q}}{\partial U} \delta U = 0 & \text{in } \Omega \quad t > 0 \\ \delta\vec{v} = -\partial_n(\vec{v} - \vec{u}_\Omega)\delta S & \text{on } S \\ \partial_n(\delta T) = \nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T)\delta S & \text{on } S \\ (\delta W)_+ = 0 & \text{on } \Gamma_\infty \end{array} \right. \tag{30}$$

where we have assumed the use of an adiabatic wall condition and introduced the following Jacobian matrices

that can be found in the appendix,

$$\left.\begin{aligned}
&\vec{A}^c = \left(A_x^c, A_y^c, A_z^c\right), && A_i^c = \left.\frac{\partial \vec{F}_i^c}{\partial U}\right|_{U(x,y,z)} \\
&\vec{A}^{vk} = \left(A_x^{vk}, A_y^{vk}, A_z^{vk}\right), && A_i^{vk} = \left.\frac{\partial \vec{F}_i^{vk}}{\partial U}\right|_{U(x,y,z)} \\
&\bar{\bar{D}}^{vk} = \begin{pmatrix} D_{xx}^{vk} & D_{xy}^{vk} & D_{xz}^{vk} \\ D_{yx}^{vk} & D_{yy}^{vk} & D_{yz}^{vk} \\ D_{zx}^{vk} & D_{zy}^{vk} & D_{zz}^{vk} \end{pmatrix}, && D_{ij}^{vk} = \left.\frac{\partial \vec{F}_i^{vk}}{\partial(\partial_j U)}\right|_{U(x,y,z)}
\end{aligned}\right\} \qquad i,j = 1\ldots 3, \quad k = 1,2. \tag{31}$$

D. The Unsteady Continuous Adjoint Equations

After introducing the linearized Navier-Stokes equations, linearized boundary conditions, and δJ from Eqn. 26 into Eqn. 29 and rearranging for clarity, the variation of the functional takes the following form after manipulation and evaluation of boundary integrals by hand,

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

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. 32 can be eliminated by satisfying the adjoint system:

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

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

The remaining terms can be gathered as,

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

One final simplification can be made in order to avoid the need to compute higher order derivatives of the

temperature. Starting from the final term in Eqn. 34,

$$\begin{aligned}
&\psi_{\rho E}\mu_{tot}^2 c_p \left[\nabla T\cdot\nabla_S(\delta S) - \partial_n^2(T)\delta S\right] \\
&= \psi_{\rho E}\mu_{tot}^2 c_p \nabla T\cdot\nabla_S(\delta S) - \psi_{\rho E}\left[\nabla\cdot(\mu_{tot}^2 c_p\nabla T) - \nabla_S\cdot(\mu_{tot}^2 c_p\nabla_S T)\right]\delta S \\
&= \psi_{\rho E}\mu_{tot}^2 c_p \nabla T\cdot\nabla_S(\delta S) + \psi_{\rho E}\delta S\,\nabla_S\cdot(\mu_{tot}^2 c_p\nabla_S T) - \psi_{\rho E}\nabla\cdot(\mu_{tot}^2 c_p\nabla T)\delta S, \qquad (35)
\end{aligned}$$

and the second term on the right hand side of Eqn. 35 can be integrated by parts as

$$\begin{aligned}
\nabla_S\cdot\left[\psi_{\rho E}\delta S\mu_{tot}^2 c_p\nabla_S T\right] &= \nabla_S(\psi_{\rho E}\delta S)\cdot(\mu_{tot}^2 c_p\nabla_S T) + \psi_{\rho E}\delta S\,\nabla_S\cdot\left(\mu_{tot}^2 c_p\nabla_S T\right) \\
\int_{t_o}^{t_f}\int_S \nabla_S\cdot\left[\psi_{\rho E}\delta S\mu_{tot}^2 c_p\nabla_S T\right]\,ds\,dt &= \int_{t_o}^{t_f}\int_S \nabla_S(\psi_{\rho E}\delta S)\cdot(\mu_{tot}^2 c_p\nabla_S T)\,ds\,dt \\
&\quad + \int_{t_o}^{t_f}\int_S \psi_{\rho E}\delta S\,\nabla_S\cdot\left(\mu_{tot}^2 c_p\nabla_S T\right)\,ds\,dt \\
\int_{t_o}^{t_f}\int_S \psi_{\rho E}\delta S\,\nabla_S\cdot\left(\mu_{tot}^2 c_p\nabla_S T\right)\,ds\,dt &= \int_{t_o}^{t_f}\int_S -\nabla_S(\psi_{\rho E}\delta S)\cdot(\mu_{tot}^2 c_p\nabla_S T)\,ds\,dt, \qquad (36)
\end{aligned}$$

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, and rearranged in going to the third line. Using the result of Eqn. 36 in Eqn. 35 gives,

$$\begin{aligned}
&= \psi_{\rho E}\mu_{tot}^2 c_p\nabla T\cdot\nabla_S(\delta S) - \nabla_S(\psi_{\rho E}\delta S)\cdot(\mu_{tot}^2 c_p\nabla_S T) - \psi_{\rho E}\nabla\cdot(\mu_{tot}^2 c_p\nabla T)\delta S \\
&= \mu_{tot}^2 c_p\nabla_S T\cdot[\psi_{\rho E}\cdot\nabla_S(\delta S) - \nabla_S(\psi_{\rho E}\delta S)] - \psi_{\rho E}\nabla\cdot(\mu_{tot}^2 c_p\nabla T)\delta S \\
&= -\mu_{tot}^2 c_p\nabla_S T\cdot\nabla_S(\psi_{\rho E})\delta S - \psi_{\rho E}\nabla\cdot(\mu_{tot}^2 c_p\nabla T)\delta S, \qquad (37)
\end{aligned}$$

and by expressing the energy equation on the surface (i.e., imposing the no-slip and heat flux boundary conditions) and substituting the result into Eqn. 37, we find that

$$\begin{aligned}
&\psi_{\rho E}\mu_{tot}^2 c_p\left[\nabla T\cdot\nabla_S(\delta S) - \partial_n^2(T)\delta S\right] \\
&= -\mu_{tot}^2 c_p\nabla_S(\psi_{\rho E})\cdot\nabla_S(T) - \psi_{\rho E}\left[p(\nabla\cdot\vec{v}) - \bar{\bar{\sigma}}:\nabla\vec{v} + \partial t(\rho E) + (\vec{q}_{\rho\vec{v}} - \partial t(\rho\vec{v}))\cdot\vec{v} - q_{\rho E}\right]. \qquad (38)
\end{aligned}$$

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

$$\begin{aligned}
\delta\mathcal{J} &= \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \{\vec{d}\cdot[\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})] + \nabla\vec{d}:(\bar{\bar{I}}p - \bar{\bar{\sigma}}) - (\bar{\bar{I}}p - \bar{\bar{\sigma}})\cdot\vec{n}\cdot\partial_n\vec{d} + \vartheta\partial_n(\vec{v} - \vec{u}_\Omega)\cdot\vec{n} \\
&\quad - \psi_{\rho E}\partial_n(\vec{v} - \vec{u}_\Omega)\cdot\bar{\bar{\sigma}}\cdot\vec{n} + \vec{n}\cdot\left(\bar{\bar{\Sigma}}^{\varphi} + \bar{\bar{\Sigma}}^{\psi_{\rho E}}\right)\cdot\partial_n(\vec{v} - \vec{u}_\Omega) - \mu_{tot}^2 c_p\nabla_S(\psi_{\rho E})\cdot\nabla_S(T) \\
&\quad - \psi_{\rho E}\left[p(\nabla\cdot\vec{v}) - \bar{\bar{\sigma}}:\nabla\vec{v} + \partial_t(\rho E) + (\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v}))\cdot\vec{v} - q_{\rho E}\right]\}\delta S\,ds\,dt, \\
&= \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left\{\frac{\partial\mathcal{J}}{\partial S}\right\}\delta S\,ds\,dt, \qquad (39)
\end{aligned}$$

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 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.

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.[29] Furthermore, it is important to check the individual contributions from the terms in Eqn. 39 and to simplfy 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 maintain accuracy in the calculated gradients. The expression in Eqn. 39 is a very general result, and for the pitching results in this article with a constant force projection vector, the terms involving the source terms and derivatives of $\vec{d}$ in the surface sensitivity do not appear.

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. The typical optimal shape design loop is depicted in Fig. 3.

All components were implemented within the SU^2 software suite (**S**tanford **U**niversity **U**nstructured).[30,31] 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.

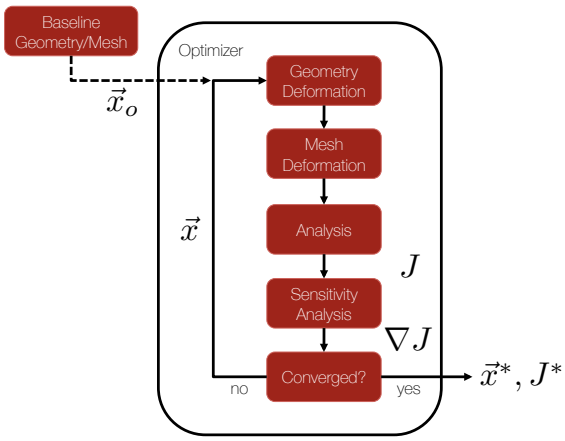


Figure 3. Flow chart for a typical shape optimization problem. J is the objective function, and $\vec{x}$ is the vector of design variables.

A. Numerical Methods for PDE Analysis

Both the governing flow and adjoint PDEs are spatially discretized on unstructured meshes via the Finite Volume Method (FVM) 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.

1. *Spatial Integration*

Convective fluxes for the flow and adjoint problems are discretized using either a centered scheme with Jameson-Schmidt-Turkel (JST)-type scalar artificial dissipation[32] or the upwind scheme of Roe.[33] The adjoint convective flux uses a modified version of the JST scheme that treats the discretization in a non-conservative manner. The convection of the turbulence variable, $\hat{\nu}$, is discretized using an upwind scheme (typically first-order). Second-order accuracy is easily achieved via reconstruction of variables on the cell interfaces by using a MUSCL approach with limitation of gradients.[34] In all cases, viscous fluxes are computed with the node-gradient-based approach due to Weiss et al.,[35] which, apart from 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. The Green-Gauss or weighted least-squares methods are available for approximating the spatial gradients of the flow and the adjoint variables. Source terms are approximated via piecewise reconstruction in the finite-volume cells.

2. *Time 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[36,37] 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. Each physical time step is relaxed in pseudo time using implicit integration. Currently, the following preconditioned Krylov subspace methods are available for solving the resulting linear systems: the Generalized Minimal Residual (GMRES) method,[38] and the Biconjugate Gradient Stabilized (Bi-CGSTAB) method.[39]

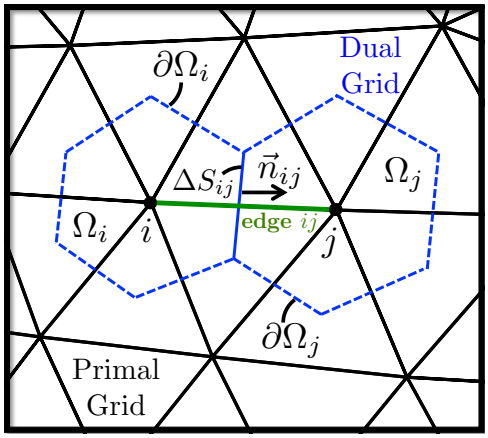


Figure 4. Dual mesh control volumes surrounding two nodes, i and j, in the domain interior.

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. While some techniques do exist that can ease the burden of data storage for the unsteady adjoint, this straightforward approach was chosen for the relatively small numerical experiments in this article.

While not discussed in detail here, a variety of convergence acceleration techniques are also available. An agglomeration multigrid method is implemented that generates effective convergence at all length scales of a problem by employing a sequence of grids of varying resolution.[40,41] Furthermore, the code is fully parallel through use of the Message Passing Interface (MPI) standard which allows for the simulation of large-scale problems on parallel computers through a typical domain partitioning approach.

3. Dynamic Meshes

Apart from solving the governing equations in ALE form, the handling of dynamic meshes forms another major component of calculating unsteady flows with moving surfaces or domains. With each new physical time step, the nodal coordinates and grid velocities must be updated using suitable methods for moving any boundaries and interior nodes of the volume mesh and computing the resulting grid velocities. Two typical strategies involve rigid mesh transformations or dynamically deforming meshes. Rigid mesh transformations have been used for the results in this article, and more detail on the approach is given below.

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

$$\vec{x}_i{}^{n+1} = \mathfrak{R}(\Delta\vec{\theta})\,\vec{r}_i + \Delta\vec{h}, \tag{40}$$

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

$$\mathfrak{R}(\Delta\vec{\theta}) = \begin{bmatrix} \cos\theta_y\cos\theta_z & \sin\theta_x\sin\theta_y\cos\theta_z - \cos\theta_x\sin\theta_z & \cos\theta_x\sin\theta_y\cos\theta_z + \sin\theta_x\sin\theta_z \\ \cos\theta_y\sin\theta_z & \sin\theta_x\sin\theta_y\sin\theta_z + \cos\theta_x\cos\theta_z & \cos\theta_x\sin\theta_y\sin\theta_z - \sin\theta_x\cos\theta_z \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y \end{bmatrix} \tag{41}$$

with $\Delta\vec{\theta} = \{\theta_x, \theta_y, \theta_z\}^{\mathsf{T}}$ 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. 40 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. 40 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 prescribed by

$$\alpha(t) = \alpha_o + \alpha_m \sin(\omega t), \tag{42}$$

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, $\omega_r = \frac{\omega c}{2v_\infty}$, 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. 42, $\Delta\vec{\theta}$ 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 in this case, the mesh velocities that were computed during the direct problem are also written to disk and retrieved with each time step. Note again that the cell volumes will remain fixed for rigid mesh transformations.

Finally, when computing unsteady flows on dynamic meshes with the ALE form of the equations, a Geometric Conservation Law (GCL) should be satisfied. First introduced by Thomas and Lombard,[42] it has been shown mathematically and through numerical experiment[43–45] that satisfying the GCL can improve the accuracy and stability of the chosen scheme. A straightforward technique for the numerical implementation of the GCL[46,47] has been included as part of the dual-time stepping approach.

B. Geometry Parameterization

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 in conjunction with gradient smoothing,[48] for instance, this approach is not pursued here. Instead, we will compute the surface sensitivities $\frac{\partial J}{\partial S}$ 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).

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):

$$\delta\mathcal{J} = \int_S \left\{\frac{\partial\mathcal{J}}{\partial S}\right\} \delta S \, ds. \tag{43}$$

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

$$\frac{\delta\mathcal{J}}{\delta\alpha_j} = \int_S \left\{\frac{\partial\mathcal{J}}{\partial S}\right\} \frac{\delta S}{\delta\alpha_j} \, ds \approx \sum_{i\in\mathcal{N}(S)} \left\{\frac{\partial\mathcal{J}}{\partial S}\right\}_i \frac{\vec{n}_i \cdot \Delta\vec{x_i}}{\Delta\alpha_j} \Delta S_i, \tag{44}$$

where S is the surface being designed, $\mathcal{N}(S)$ represents the set of mesh nodes on the S, $\left\{\frac{\partial J}{\partial S}\right\}_i$ 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 $\left\{\frac{\partial J}{\partial S}\right\}_i$ is given by the surface sensitivity formulas, the remaining terms in Eqn. 44 are geometric in nature and are 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.

1. Bump Functions

In the two-dimensional airfoil calculations that follow, Hicks-Henne bump functions were employed,[49] 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

$$f_n(x) = \sin^3(\pi x^{e_n}), \;\; e_n = \frac{\log(0.5)}{\log(x_n)}, \;\; x \in [0,1], \tag{45}$$

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.

2. Free-Form Deformation

In three dimensions, a Free-Form Deformation (FFD)[50] 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

$$X(u,v,w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w), \tag{46}$$

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. 46.

C. 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. 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[51–53] can help preserve grid quality near solid surfaces where methods based on a spring analogy might fail (resulting in negative cell volumes).

In three dimensions, linear elasticity governs small displacements $\vec{u} = (u_1, u_2, u_3)^\mathsf{T}$ of an elastic solid subject to body forces $\vec{f}$ and surface tractions,

$$\begin{cases} \mathcal{M}(\vec{u}) = \nabla \cdot \bar{\bar{\sigma}} - \vec{f} = 0 & \text{in } \Omega \\ \vec{u} = \vec{g} & \text{on } \Gamma \end{cases} \tag{47}$$

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

$$\bar{\bar{\sigma}} = \lambda Tr(\bar{\bar{\epsilon}}) \bar{\bar{I}} + 2\mu \bar{\bar{\epsilon}}, \tag{48}$$

with the strain-displacement relation written as

$$\bar{\bar{\epsilon}} = \frac{1}{2}(\nabla \vec{u} + \nabla \vec{u}^\mathsf{T}), \tag{49}$$

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

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}, \tag{50}$$

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 the distance to the nearest solid wall), 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 ($\vec{g}$ in the governing equations above). 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.

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[a], 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 values and gradients for any chosen constraints.

V. Numerical Results

A. Pitching Airfoil in Turbulent Flow

In order to validate the implementation of the unsteady RANS equations in ALE form, a comparison was made against the well-known CT6 data set of Davis.[54] The physical experiment measured the unsteady performance for the NACA 64A010 airfoil pitching about the quarter-chord point. The particular experimental case of interest studied pitching motion with a reduced frequency, or w_r, of 0.202, $M_\infty = 0.796$, a mean angle of attack of 0 degrees, and a Reynolds number of 12.5 million.

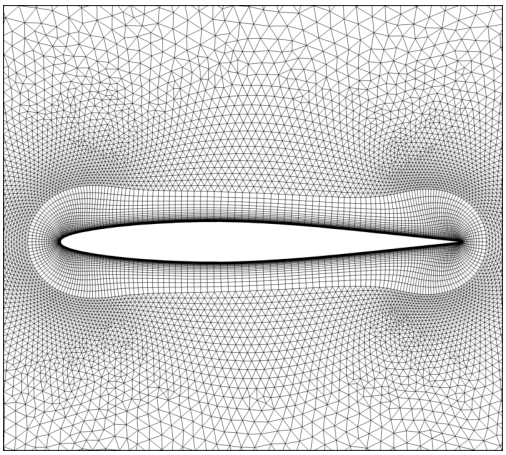


Figure 5. Close-up view of the hybrid mesh around the airfoil geometry.

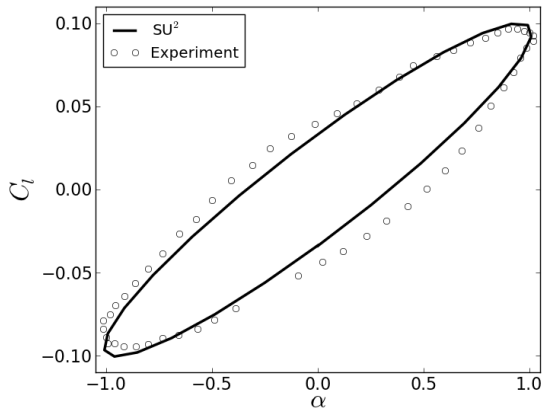
A mixed-element, unstructured mesh was constructed around the airfoil that consisted of 22,904 triangular elements, 12,500 quadrilaterals near the airfoil surface to capture the boundary layer, 24,111 nodes in total, 250 edges along the airfoil surface, and 68 edges along the far-field boundary. The spacing at the wall was chosen to achieve a $y+$ value less than 1. A view of the mesh near the airfoil is presented in Fig. 5.

The unsteady RANS equations with the Spalart-Allmaras turbulence model were solved for the flow around the pitching airfoil. All numerical simulations were performed with 25 time steps per period of oscillation for a total of 10 periods, which afforded adequate time for transient effects to wash away and well-resolved, time-averaged behavior in the periodic steady state. 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.

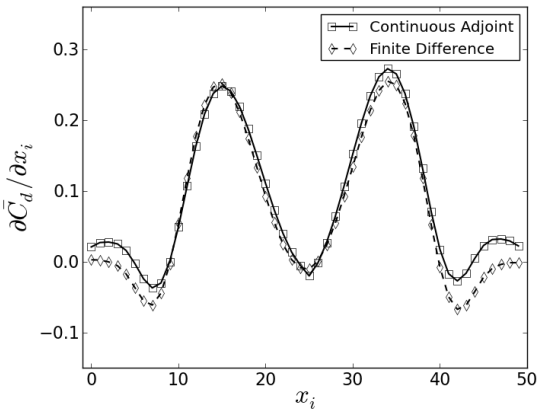
Fig. 6 shows a comparison of the lift coefficient versus angle of attack between numerical results from SU^2 and experimental data during the final period of oscillation. In physical time, the curve is traversed in a counterclockwise fashion. Note that nonlinear behavior corresponding to moving shock waves results in a hysteresis effect. The numerical results agree well with experimentally measured values.

50 bump functions 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 (25 bumps each on the upper and lower surfaces). In order to verify the accuracy of the gradient information obtained via the time-accurate

[a] `http://www.scipy.org`



(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.

Figure 6. Numerical results for a pitching NACA 64A010 in transonic flow.

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 ($1E-6c$). The resulting gradients are compared in Fig. 6 and exhibit good agreement.

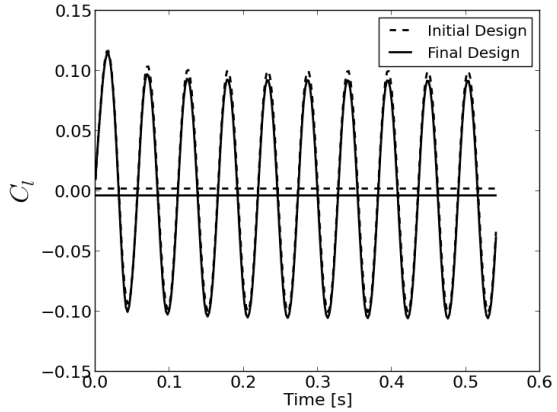
Finally, a redesign of the pitching NACA 64A010 airfoil was performed using the gradient information obtained from the time-accurate viscous adjoint formulation. The specific shape optimization problem was for time-averaged drag minimization using the same 50 bump design variables used above with a constraint 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. 7, along with the optimization history. Profile shapes of the initial and final designs are also compared in Fig. 7. The transonic shocks have been removed from the design by a thinning of the profile shape near the mid-chord while maintaining a constant airfoil area.

B. Pitching Wing in Turbulent Flow

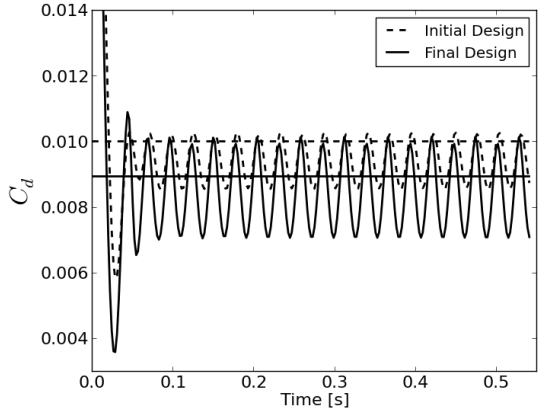
To provide a more realistic test of the time-accurate adjoint capabilities in three dimensions, the ONERA M6 wing was used as a baseline geometry. This unsteady test case was performed at a transonic Mach number with the wing pitching about an axis that passes through its quarter chord location and is perpendicular to the root airfoil section. The specific flow conditions were as follows: $w_r = 0.1682$, $M_\infty = 0.8395$, 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 initial 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 $y+ \approx 1$ 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 a typical far-field boundary. The surface meshes for the wing geometry and symmetry plane are shown in Fig. 8. All numerical simulations of the pitching wing were performed with 25 times steps per period for a total of 7 periods until reaching a periodic steady state.

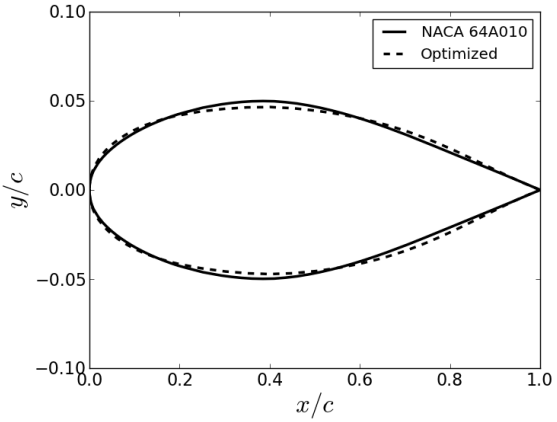
The pitching ONERA M6 wing was redesigned using gradients obtained via the viscous, time-accurate adjoint. Note again that, for the unsteady adjoint, the solution at each time step from the direct problem (including mesh node coordinates and grid velocities) was written to file and then loaded in reverse fashion while integrating the adjoint equations backward in physical time. Three-dimensional design variables were defined using a FFD parameterization. Movement in the vertical direction (z-direction) was allowed for 50 control points in total on the upper and lower surfaces of the FFD box. Fig. 8 contains a view of the FFD box around the wing geometry.



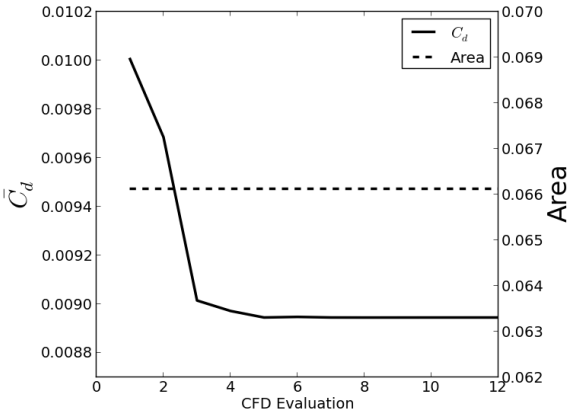
(a) Comparison of lift coefficient versus angle of attack in degrees between SU^2 and experiment using URANS.



(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.

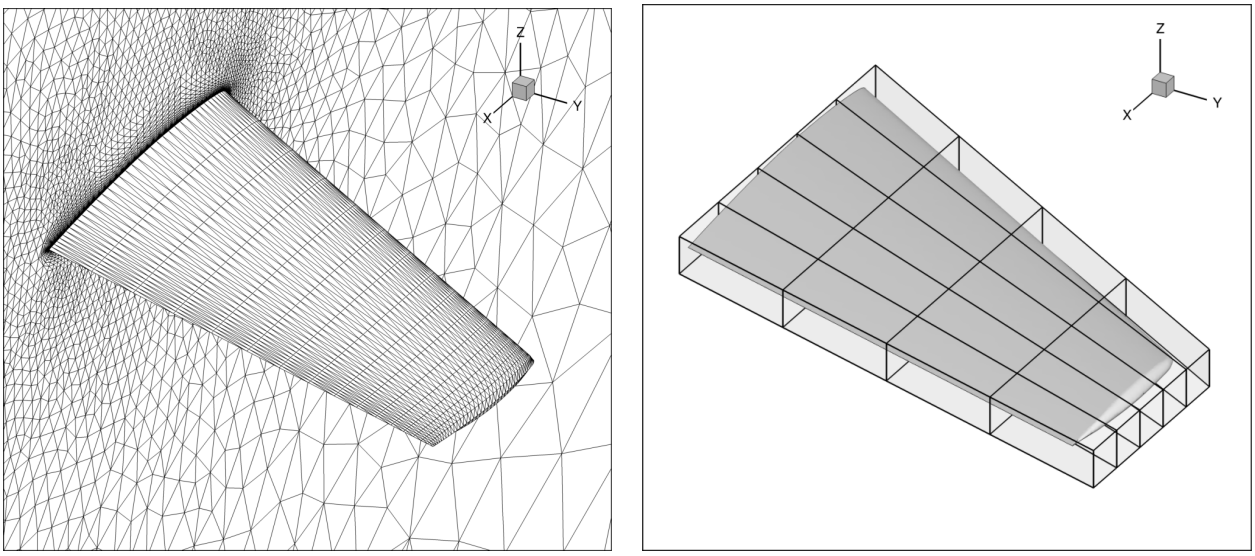


(d) Average drag and area for each CFD evaluation during the optimization process.

Figure 7. Force coefficient histories, shape comparison, and optimization history for the pitching airfoil design.

The specific shape optimization problem was for the minimization of the time-averaged drag with lift and geometric constraints. An aggressive constraint was imposed on the time-averaged coefficient of lift, as it was required to be greater than a value of 0.268 (a 4 % increase over the $\bar{C_L}$ for the initial geometry). The maximum thickness at five spanwise sections of the wing was also constrained to be larger than specified values. After 11 evaluations, the time-averaged drag, $\bar{C_D}$, was successfully reduced by 20.3 %, and the geometric constraints were met. The optimizer was also successfully able to increase the time-averaged lift to a value of 0.266, although this was slightly less that the prescribed constraint value.

Fig. 9 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 been reduced in the optimized design due to a reduction in the shock strengths when the wing is at large incidence angles. Fig. 10 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 greatly reduced. The section profile shapes of the initial and final designs are compared in Fig. 9. The optimized geometry features increased camber, especially near the tip, and a slight thinning of the sections.



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

(b) Wing surface with the surrounding FFD box.

Figure 8. Surface mesh and FFD box details for the pitching ONERA M6 numerical experiment.

VI. Conclusions

A viscous continuous adjoint formulation for optimal shape design in unsteady flows 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. First, a gradient verification study was performed using a pitching NACA 64A010 as a test case. Good agreement was found between the gradients provided by the unsteady continuous adjoint approach and finite differencing. The pitching NACA 64A010 was redesigned for minimum time-averaged drag with a geometric constraint on the internal area of the airfoil. The time-averaged drag of the optimal design was 10.6 % lower than the baseline geometry. A larger, more realistic design case was also performed using a pitching ONERA M6 wing as a baseline. Similarly, a time-averaged drag reduction of 20.3 % was achieved while meeting geometric constraints on wing section thicknesses and maintaining the original time-averaged lift.

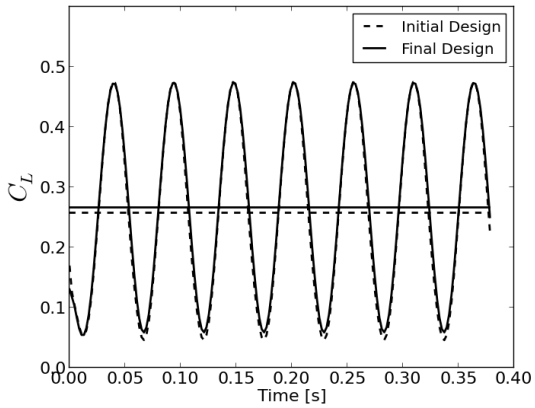
It is important to note that the methodology presented in this article is efficient both computationally and in terms of memory performance. The formulation was derived with generality in mind by treating a general set of governing equations that allow for unsteadiness and surfaces in motion, and the implementation can be used for complex geometries on unstructured meshes. Lastly, while not discussed in this article, the unsteady continuous adjoint formulation presented here 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.

VII. Acknowledgements

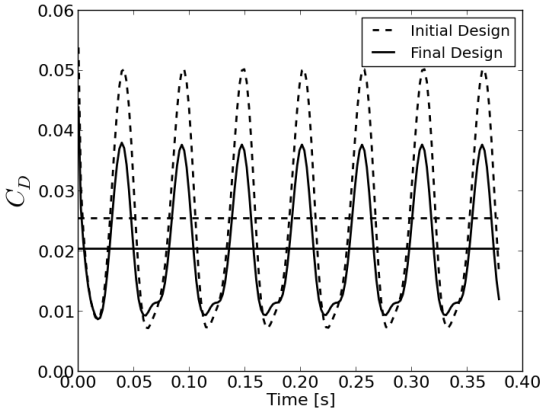
T. Economon would like to acknowledge Prof. Antony Jameson, Prof. Robert MacCormack, Prof. Sanjiva Lele, and Prof. Michael Saunders for helpful discussions, feedback on this work, as well as their willingness to serve on the PhD defense committee.

References

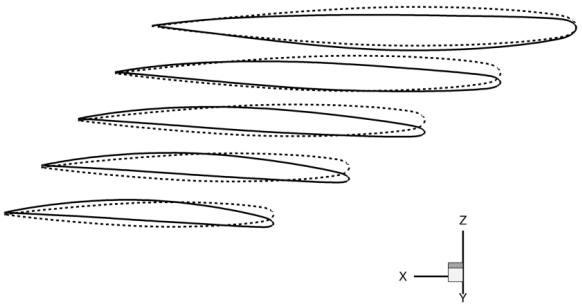
[1] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971.



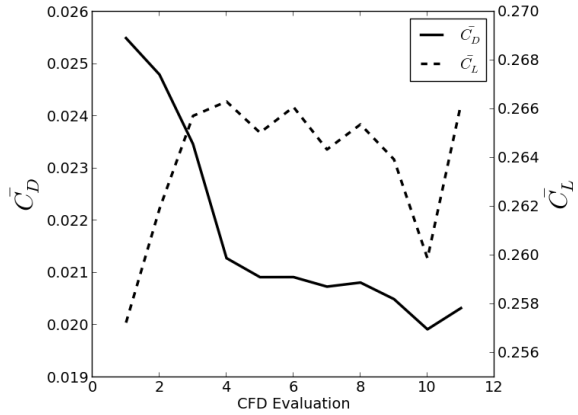
(a) C_L history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average lift increased with the new design.



(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 section of the initial and final wing designs.



(d) Lift and drag values for each CFD evaluation during the optimization process.

Figure 9. Force coefficient histories, section shape comparison, and optimization history for the pitching wing design.

[2]O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.

[3]A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[4]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.

[5]M. P. Rumpfkeil and D. W. Zingg. A general framework for the optimal control of unsteady flows with applications. *AIAA Paper 2007-1128*, 2007.

[6]D. J. Mavriplis. Solution of the unsteady discrete adjoint for three-dimensional problems on dynamically deforming unstructured meshes. *AIAA Paper 2008-727*, 2008.

[7]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.

[8]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.

[9]E. J. Nielsen and B. Diskin. Discrete adjoint-based design for unsteady turbulent flows on dynamic overset unstructured grids. *AIAA Paper 2012-0554*, 2012.

[10]T. D. Economon, F. Palacios, and J. J. Alonso. Unsteady aerodynamic design on unstructured meshes with sliding interfaces. *AIAA Paper 2013-0632*, 2013.

[11]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.

[12]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.

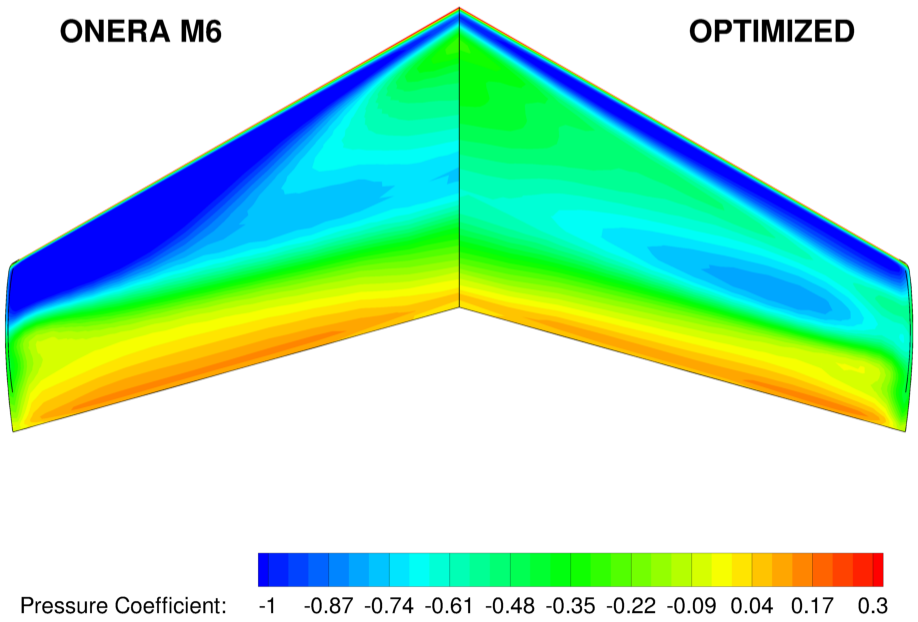


Figure 10. Pressure coefficient comparison between the ONERA M6 and the final design at the incidence of maximum drag for the URANS case.

[13]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.

[14]T. D. Economon, F. Palacios, and J. J. Alonso. Optimal shape design for open rotor blades. *AIAA Paper 2012-3018*, 2012.

[15]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.

[16]A. K. Gopinath and A. Jameson. Time spectral method for periodic unsteady computations over two- and three-dimensional bodies. *AIAA Paper 2005-1220*, 2005.

[17]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.

[18]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.

[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]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.

[21]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.

[22]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.

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

[24]D.C. Wilcox. *Turbulence Modeling for CFD*. 2nd Ed., DCW Industries, Inc., 1998.

[25]F. M. White. *Viscous Fluid Flow*. McGraw Hill Inc., New York, 1974.

[26]P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 1992-0439*, 1992.

[27]O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.

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

[29]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.

[30]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 (SU^2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper 2013-0287*, 2013.

[31]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 (SU^2): Open-source analysis and design technology for turbulent flows. *AIAA Paper 2014-0243*, 2014.

[32]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.

[33]P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[34]V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper 1993-0880*, 1993.

[35]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.

[36]A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 1991-1596*, 1991.

[37]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.

[38]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.

[39]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.

[40]D. J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report 95-27, ICASE, 1995.

[41]D. J. Mavriplis. On convergence acceleration techniques for unstructured meshes. Technical report, Institute for Computer Applications on Science and Engineering (ICASE), 1998.

[42]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.

[43]J. T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8):1381–1388, August 1990.

[44]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.

[45]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.

[46]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.

[47]R. T. Biedron and J. L. Thomas. Recent enhancements to the FUN3D flow solver for moving-mesh applications. *AIAA Paper 2009-1360*, 2009.

[48]A. Jameson, S. Sriram, and L. Martinelli. A continuous adjoint method for unstructured grids. *AIAA Paper*, 2003-3955, 2003.

[49]R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[50]J. A. Samareh. Aerodynamic shape optimization based on free-form deformation. *AIAA Paper 2004-4630*, 2004.

[51]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.

[52]E. J. Nielsen and W. K. Anderson. Recent improvements in aerodynamic design optimization on unstructured meshes. *AIAA Journal*, 40(6), June 2002.

[53]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.

[54]S. S. Davis. NACA 64A010 (NASA Ames model) oscillatory pitching, compendium of unsteady aerodynamic measurements. Technical Report AGARD R-702, AGARD, 1982.

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

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