51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0632

Unsteady Aerodynamic Design on Unstructured Meshes with Sliding Interfaces

Thomas D. Economon[*], Francisco Palacios[†], and Juan J. Alonso[‡]

Stanford University, Stanford, CA 94305, U.S.A.

A time-accurate, continuous adjoint formulation is developed, verified, and applied. This formulation enables the efficient design of aerodynamic surfaces in unsteady, inviscid flow within a gradient-based optimization framework. A systematic method for the derivation of suitable boundary conditions for the adjoint problem is demonstrated, and a new sliding mesh algorithm for unstructured meshes is also developed and tested. The gradient information provided by the time-accurate adjoint formulation is verified for design variables in two dimensions with and without the use of sliding meshes, and in both situations, gradients compare very favorably with those obtained via finite differencing. Optimal shape design results for a pitching airfoil demonstrate the effectiveness of the unsteady adjoint approach. Furthermore, performing shape design with sliding meshes recovers nearly identical results as a non-sliding case, suggesting that their use in conjunction with the unsteady adjoint can provide a stable and accurate design approach. The combination of sliding mesh interfaces and the time-accurate continuous adjoint produces a powerful design tool for unsteady problems with surfaces in relative motion.

Nomenclature

Variable	*Definition*
c	Airfoil chord length
$\vec{d}$	Force projection vector
j_S	Scalar function defined at each point on S
$\vec{n}$	Unit normal vector
p	Static pressure
p_∞	Freestream pressure
t	Time variable
t_o	Initial time
t_f	Final time
$\vec{u}_x$	Local velocity at a point in a moving domain (mesh velocity)
$\vec{v}$	Flow velocity vector
v_∞	Freestream velocity
$\vec{x}$	Position in cartesian space
$\vec{A}$	Inviscid flux Jacobian matrices
A_z	Projected area in the z-direction
C_D	Coefficient of drag
C_L	Coefficient of lift
C_{SF}	Coefficient of side-force
C_p	Coefficient of pressure
E	Total energy per unit mass
$\vec{F}$	Euler convective fluxes

[*] Ph.D. Candidate, Department of Aeronautics & Astronautics, AIAA Student Member.

[†] Engineering Research Associate, Department of Aeronautics & Astronautics, AIAA Member.

[‡] Associate Professor, Department of Aeronautics & Astronautics, AIAA Associate Fellow.

$\vec{F}_{ale}$	Euler convective fluxes in ALE form
H	Stagnation enthalpy
$\bar{\bar{I}}$	Identity matrix
J	Cost function defined as an integral over S
$\mathcal{J}$	Lagrangian
M_∞	Freestream Mach number
$\mathfrak{R}$	Rotation matrix for rigid grid motion
$\mathcal{R}(U)$	System of governing flow equations
S	Solid wall flow domain boundary (design surface)
$\mathbb{T}$	Time interval, $t_f - t_o$
U	Vector of conservative variables
W	Vector of characteristic variables
α	Angle of attack
α_o	Mean pitching angle
α_m	Pitching amplitude
β	Sideslip angle
γ	Ratio of specific heats, $\gamma = 1.4$ for air
ρ	Fluid density
ρ_∞	Freestream density
$\vec{\phi}$	Adjoint velocity vector
ω	Angular frequency
ω_r	Reduced frequency, $\frac{\omega c}{2 v_\infty}$
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, $\vec{n}_S \cdot \nabla(\cdot)$
$\cdot$	Vector inner product
$\times$	Vector cross product
$\otimes$	Vector outer product
B^T	Transpose operation on column vector or matrix B
$\delta(\cdot)$	Denotes first variation of a quantity

I. Introduction and Motivation

Many practical flows of aerodynamic interest are unsteady in nature, and between the increasing power of computational resources and advanced algorithms, accurately predicting and designing for the performance of aerospace systems in an unsteady environment is becoming more tractable. Several examples of engineering applications that could immediately benefit from a truly time-accurate design approach are counter-rotating open rotors, rotorcraft, turbomachinery, wind turbines, or flapping flight, to name a few. An unsteady treatment of these flows will also directly enable multidisciplinary design, analysis, and optimization (MDAO) involving other time-dependent physics associated with these systems, such as their structural or acoustic responses. Active flow control using devices that augment the flow field (jets, actuators, etc.) is another area that could be investigated with time-dependent inputs and included in a design framework.

Most of the examples above require some type of dynamic mesh capability for simulations with surfaces moving through a fluid, e.g. a pitching airfoil in a uniform freestream or an isolated helicopter rotor in forward flight. Grid motion strategies for these cases often include either surface movements followed by volumetric mesh deformation through a spring analogy or a rigid transformation of the entire mesh with each physical time step. In the more complicated case where multiple surfaces move relative to one another within a single computational domain, such as with counter-rotating rotors, rotor-fuselage interactions for rotorcraft or fixed-wing vehicles, or rotor-stator interactions within a turbomachine, then a system of sub-grids, or zones, with independent motion will be needed. Furthermore, communication of solution information between these zones must be achieved through a sliding mesh interface, an overset mesh approach, a mixing plane approximation, or another method that appropriately considers the relative grid motion.[1–3]

In the context of optimal shape design, adjoint formulations as a means of sensitivity analysis have been the subject of a rich volume of research literature over the past two decades. Many advances and extensions have been made during this period, and the effectiveness of these formulations for use in aerodynamic design, especially for steady problems, is well established.[4–6] Less common and more challenging are adjoint formulations for unsteady problems due to potentially prohibitive storage requirements associated with time-accurate data and the need for reverse time integration when solving the corresponding adjoint equations. However, recent work demonstrating the viability of unsteady adjoint approaches across a range of applications[7–13] and the aforementioned improvements in computational power and algorithms suggest a growing interest and capability for design in unsteady flows.

Adjoint formulations are typically classified as either continuous (the governing equations are first linearized then the result is discretized) or discrete (the governing equations are first discretized and the result is linearized). A large amount of the previous work on unsteady adjoints has been discrete in nature, and while a discrete adjoint approach can often be more straightforward to implement, especially if automatic differentiation is available, we pursue advances in the continuous approach with this article. The continuous formulation can offer the advantage of physical insight into the character of the governing flow equations and their adjoint system, and this insight can aid in composing well-behaved numerical solution methods. However, consistent boundary conditions that accompany the continuous adjoint equations must also be derived, and unfortunately, clear strategies for their derivation are less prevalent in the literature. Lastly, while sliding mesh techniques are often applied to unsteady flow simulations, their suitability for the unsteady continuous adjoint problem, which requires reverse time integration of the equations with a non-conservative spatial discretization, is less explored. Robust and accurate boundary conditions and numerical methods for the adjoint problem will be essential when moving to the design of larger, more complex aerospace applications.

Therefore, the contributions of this article are the following: a detailed derivation of a time-accurate, continuous adjoint formulation including an extended discussion of the adjoint boundary conditions, the development of a sliding mesh strategy for unstructured grids, and the application of these components to a two-dimensional pitching airfoil problem. More specifically, the adjoint treatment presented is a systematic methodology centered around finding *surface sensitivities* with the use of differential geometry formulas. This type of surface formulation has no dependence on volume mesh sensitivities and has been successfully applied to full aircraft configurations and even extended to the RANS equations.[14] It is here extended for optimal shape design in unsteady flows with the added flexibility to handle complex, dynamic geometries that is afforded by unstructured meshes with sliding mesh interfaces.

The paper is organized as follows. In Section II, a description of the physical problem in which we are interested is given, including the governing flow equations with appropriate boundary conditions. Section III contains a complete derivation of a time-accurate, continuous adjoint formulation for the unsteady Euler equations. An extended discussion of suitable boundary conditions for the adjoint problem can also be found in this section. Section IV details the numerical methods employed, development of a sliding mesh interface approach, and the shape optimization approach. Lastly, Section V presents results for a verification and validation of the methodology as applied to a pitching NACA 64A010 airfoil and, finally, a redesign of the profile shape for minimum time-averaged drag. Special emphasis is placed on the comparison of results with and without the use of sliding meshes.

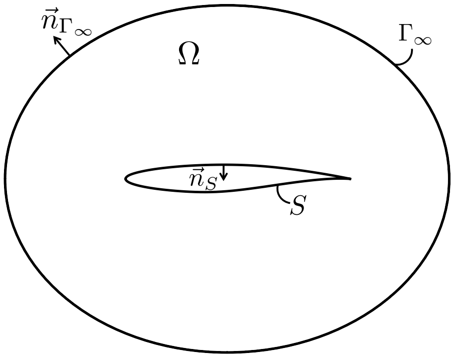


Figure 1. Notional schematic of the flow domain, Ω, the boundaries, Γ_∞ and S, as well as the definition of the surface normals.

II. Description of the Physical Problem

Ideal fluids are governed by the Euler equations. In our particular problem, these equations are considered in a domain, Ω, bounded by a disconnected boundary which is divided into a far-field component, Γ_∞, and a solid wall boundary, S, as seen in Fig. 1. The surface S will also be referred to as the design surface, and it is considered continuously differentiable (C^1). Normal vectors to the boundary surfaces are directed out of the domain by convention.

We are interested in the time-accurate fluid behavior around aerodynamic bodies in arbitrary motion for situations where viscous effects can be considered negligible. The governing partial differential equations (PDEs) in the limit of vanishing viscosity are the unsteady, compressible Euler equations. These conservation equations can be expressed in an arbitrary Lagrangian-Eulerian (ALE)[15] differential form as

$$\left\{\begin{array}{lll} \mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}_{ale} = 0, & \text{in } \Omega, & t_o \leq t \leq t_f \\ (\vec{v} - \vec{u}_x) \cdot \vec{n}_S = 0, & \text{on } S, & t_o \leq t \leq t_f \\ (W)_+ = W_\infty, & \text{on } \Gamma_\infty, & t_o \leq t \leq t_f \end{array}\right. \tag{1}$$

where

$$U = \left\{\begin{array}{c} \rho \\ \rho\vec{v} \\ \rho E \end{array}\right\}, \quad \vec{F}_{ale} = \left\{\begin{array}{c} \rho(\vec{v} - \vec{u}_x) \\ \rho\vec{v} \otimes (\vec{v} - \vec{u}_x) + \bar{\bar{I}}p \\ \rho E(\vec{v} - \vec{u}_x) + p\vec{v} \end{array}\right\}, \tag{2}$$

ρ is the fluid density, $\vec{v} = \{u, v, w\}^\mathsf{T}$ is the flow velocity in a Cartesian coordinate system, $\vec{u}_x$ is the local velocity at a point, $\vec{x}$, in a moving domain (mesh velocity after discretization), E is the total energy per unit mass, and p is the static pressure. The second line of Eqn. 1 represents the flow tangency condition at a solid wall, and the final line represents a characteristic-based boundary condition at the far-field where the fluid state at the boundary is updated using the state at infinity depending on the sign of the eigenvalues. This boundary will be treated generally and its implications on the adjoint problem will be discussed in detail below. The boundary conditions also take into account any velocity due to domain motion. The temporal conditions will be problem dependent, and for purposes of this article, we will be interested in time-periodic flows where the initial and terminal conditions do not affect the time-averaged behavior over the time interval of interest, $\mathbb{T} = t_f - t_o$. In order to close the system of equations after assuming a perfect gas, the pressure is determined from

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}(\vec{v} \cdot \vec{v})\right], \tag{3}$$

and the stagnation enthalpy is given by

$$H = E + \frac{p}{\rho}. \tag{4}$$

III. Surface Sensitivities via a Time-Accurate Adjoint Formulation

A typical shape optimization problem seeks the minimization of a certain cost function, J, with respect to changes in the shape of the boundary, S, with a constraint that the equations governing the system with suitable temporal and boundary conditions, $\mathcal{R}(U)$, be satisfied. We will concentrate on functionals defined as time-averaged, integrated quantities on the solid surface, so the optimization problem can be expressed mathematically as

$$\begin{array}{ll} \textbf{Minimize} & J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f}\!\int_S j_S \, ds \, dt \\ \textbf{such that} & \mathcal{R}(U) = 0, \end{array} \tag{5}$$

where $j_S = \vec{d} \cdot p\,\vec{n}_S$ is a time-dependent, pressure-based functional defined at each point on the surface, as we are under an inviscid flow assumption. The vector $\vec{d}$ is the force projection vector, and it is an arbitrary, constant vector which can be chosen to relate the pressure at the surface to a desired quantity of interest.

For external aerodynamic applications, likely candidates are

$$\vec{d} = \left\{ \begin{array}{lll} \left(\frac{1}{C_\infty}\right)(\cos\,\alpha\;\cos\,\beta, \sin\,\alpha\;\cos\,\beta, \sin\,\beta), & C_D & \text{Drag coefficient} \\ \left(\frac{1}{C_\infty}\right)(-\sin\,\alpha, \cos\,\alpha, 0), & C_L & \text{Lift coefficient} \\ \left(\frac{1}{C_\infty}\right)(-\sin\,\beta\;\cos\,\alpha, -\sin\,\beta\;\sin\,\alpha, \cos\,\beta), & C_{SF} & \text{Side-force coefficient} \\ \left(\frac{1}{C_\infty C_D}\right)(-\sin\alpha - \frac{C_L}{C_D}\cos\alpha\cos\beta, -\frac{C_L}{C_D}\sin\beta, \cos\alpha - \frac{C_L}{C_D}\sin\alpha\cos\beta), & \frac{C_L}{C_D} & \text{L/D} \end{array} \right. \tag{6}$$

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

A. The Continuous Adjoint Approach

The minimization of Eqn. 5 can be considered a problem of optimal control whereby the behavior of the governing flow equation system is controlled by the shape of S with deformations of the surface acting as the control input. Therefore, the goal is to compute the variation of Eqn. 5 caused by arbitrary but small (and multiple) deformations of S and to use this sensitivity information to drive our geometric changes in order to find an optimal shape for S. This leads directly to a gradient-based optimization framework.

Following the continuous adjoint approach to optimal design, Eqn. 5 can be transformed into an unconstrained optimization problem by adding the inner product of an unsteady adjoint variable vector, Ψ, and the continuous governing equations integrated over the domain (space and time) to form the Lagrangian:

$$\mathcal{J} = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot(p\vec{n}_S)\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\mathcal{R}(U)\,d\Omega\,dt, \tag{7}$$

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

$$\Psi = \left\{ \begin{array}{c} \psi_\rho \\ \psi_{\rho u} \\ \psi_{\rho v} \\ \psi_{\rho w} \\ \psi_{\rho E} \end{array} \right\} = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}. \tag{8}$$

Note that because the flow equations must be satisfied in the domain, or $\mathcal{R}(U) = 0$, Eqn. 5 and Eqn. 7 are equivalent. In order to find the sensitivity information needed to minimize the objective function, we now take the first variation of Eqn. 7 with respect to infinitesimal shape deformations applied to S that can be described mathematically by

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

where S has been deformed to a new surface S' by applying an infinitesimal profile deformation, δS, in the local normal direction, $\vec{n}_S$, at a point, $\vec{x}$, on the surface, as shown in Fig. 2. With respect to these arbitrary perturbations of the surface shape and after simplification, the variation of the objective function becomes:

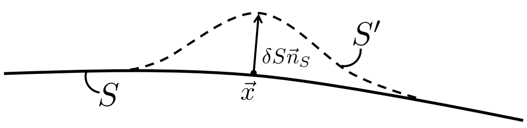


Figure 2. An infinitesimal shape deformation in the local surface normal direction.

$$\delta\mathcal{J} = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S (\vec{d}\cdot\nabla p)\delta S\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S (\vec{d}\cdot\vec{n}_S)\delta p\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\delta\mathcal{R}(U)\,d\Omega\,dt. \tag{10}$$

It is important to note that the first two terms of Eqn. 10 are found by using differential geometry formulas,[16] and this is a key feature differentiating the current formulation from other adjoint approaches. The third term of Eqn. 10 can be expanded by including the linearized version of the governing equations with respect to the small perturbations of the design surface (which induce perturbations in U),

$$\begin{aligned}\delta\mathcal{R}(U) &= \delta\left[\frac{\partial U}{\partial t} + \nabla\cdot\vec{F} - \nabla\cdot(U\otimes\vec{u}_x)\right]\\ &= \frac{\partial}{\partial t}(\delta U) + \nabla\cdot\left(\frac{\partial\vec{F}}{\partial U}\delta U\right) - \nabla\cdot\left[\frac{\partial(U\otimes\vec{u}_x)}{\partial U}\delta U\right]\\ &= \frac{\partial}{\partial t}(\delta U) + \nabla\cdot\left(\vec{A} - \bar{\bar{I}}\vec{u}_x\right)\delta U, \end{aligned} \tag{11}$$

along with the linearized form of the boundary condition at the surface,

$$\delta\vec{v}\cdot\vec{n}_S = -(\vec{v}-\vec{u}_x)\cdot\delta\vec{n}_S - \partial_n(\vec{v}-\vec{u}_x)\cdot\vec{n}_S\delta S, \tag{12}$$

where $\vec{A}$ is the Jacobian of $\vec{F}$ using conservative variables, $\frac{\partial\vec{F}}{\partial U}$. In the first line of Eqn. 11, the terms involving the domain velocity have been separated from the traditional Euler fluxes. Eqn. 11 can now be introduced into Eqn. 10 to produce

$$\begin{aligned}\delta\mathcal{J} = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S (\vec{d}\cdot\nabla p)\delta S\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S (\vec{d}\cdot\vec{n}_S)\delta p\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^{\mathsf{T}}\frac{\partial}{\partial t}(\delta U)\,d\Omega\,dt\\ + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^{\mathsf{T}}\nabla\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U\,d\Omega\,dt.\end{aligned} \tag{13}$$

The key to the adjoint approach is the removal of any dependence on variations of the flow variables (δU and δp in this case) so that the variation of the objective function for multiple surface deformations can be found *without* the need for multiple flow solutions. This results in a computationally efficient method for aerodynamic sensitivity analysis within a large design space, because the computational cost no longer depends on the number of design variables as it does with a direct method like finite differencing. We now perform manipulations to remove this dependence. After changing the order of integration, integrating the third term of Eqn. 13 by parts gives

$$\int_\Omega\int_{t_o}^{t_f}\Psi^{\mathsf{T}}\frac{\partial}{\partial t}(\delta U)\,dt\,d\Omega = \int_\Omega\left[\Psi^{\mathsf{T}}\delta U\right]_{t_o}^{t_f}\,d\Omega - \int_\Omega\int_{t_o}^{t_f}\frac{\partial\Psi^{\mathsf{T}}}{\partial t}\delta U\,dt\,d\Omega. \tag{14}$$

A zero-valued initial condition for the adjoint variables can be imposed, and assuming an unsteady flow with time-periodic behavior, the first term on the right hand side of Eqn. 14 can be eliminated with the following temporal conditions (the cost function does not depend on t_f):

$$\Psi(\vec{x},t_o) = 0, \tag{15}$$

$$\Psi(\vec{x},t_f) = 0. \tag{16}$$

Now, integrating the fourth term of Eqn. 13 by parts yields

$$\int_{t_o}^{t_f}\int_\Omega\Psi^{\mathsf{T}}\nabla\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U\,d\Omega\,dt = \int_{t_o}^{t_f}\int_\Omega\nabla\cdot\left[\Psi^{\mathsf{T}}\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U\right]d\Omega\,dt - \int_{t_o}^{t_f}\int_\Omega\nabla\Psi^{\mathsf{T}}\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U d\Omega\,dt, \tag{17}$$

and applying the divergence theorem to the first term on the right hand side of Eqn. 17, assuming a smooth solution, gives

$$\begin{aligned}\int_{t_o}^{t_f}\int_\Omega\Psi^{\mathsf{T}}\nabla\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U\,d\Omega\,dt = \int_{t_o}^{t_f}\int_S\Psi^{\mathsf{T}}\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\cdot\vec{n}_S\;\delta U ds\,dt + \int_{t_o}^{t_f}\int_{\Gamma_\infty}\Psi^{\mathsf{T}}\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\cdot\vec{n}_\infty\;\delta U ds\,dt\\ - \int_{t_o}^{t_f}\int_\Omega\nabla\Psi^{\mathsf{T}}\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\delta U d\Omega\,dt.\end{aligned} \tag{18}$$

With the appropriate choice of boundary conditions, the integral over the far-field boundary can be forced to vanish, and this will be discussed in detail below. Combining and rearranging the results from Eqns. 13, 14 (after reversing the order of integration again), 15, 16 & 18 yields an intermediate expression for the variation of the cost function,

$$\begin{aligned}\delta\mathcal{J} = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S (\vec{d}\cdot\nabla p)\delta S\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S (\vec{d}\cdot\vec{n}_S)\delta p\,ds\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S \Psi^{\mathsf{T}}\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\cdot\vec{n}_S\;\delta U ds\,dt \\ -\frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_\Omega \left[\frac{\partial\Psi^{\mathsf{T}}}{\partial t} + \nabla\Psi^{\mathsf{T}}\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\right]\delta U\,d\Omega\,dt.\end{aligned} \tag{19}$$

The surface integral in the third term on the right hand side of Eqn. 19 can be evaluated by hand given our knowledge of $\vec{A}$, $\vec{u}_x$, the wall boundary condition, and the linearized wall boundary condition in Eqn. 12. By leveraging previous derivation by the authors[16] with slight modifications (including time integration), it can be shown that evaluating the surface integral and rearranging the variation of the functional gives

$$\begin{aligned}\delta\mathcal{J} =& \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S \left[\vec{d}\cdot\nabla p + (\nabla\cdot\vec{v})\vartheta + (\vec{v}-\vec{u}_x)\cdot\nabla(\vartheta)\right]\delta S\,ds\,dt \\ &+ \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S \left[\vec{d}\cdot\vec{n}_S - \vec{n}_S\cdot\vec{\varphi} - \psi_{\rho E}(\vec{v}\cdot\vec{n}_S)\right]\delta p\,ds\,dt - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_\Omega \left[\frac{\partial\Psi^{\mathsf{T}}}{\partial t} + \nabla\Psi^{\mathsf{T}}\cdot\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\right]\delta U\,d\Omega\,dt,\end{aligned} \tag{20}$$

where $\vartheta = \rho\psi_\rho + \rho\vec{v}\cdot\vec{\varphi} + \rho H\psi_{\rho E}$, as a shorthand. Finally, by satisfying the system of PDEs commonly referred to as the adjoint equations along with the admissible adjoint boundary condition that eliminates the dependence on the fluid flow variation at the surface (δp), both the second and third terms on the right hand side of Eqn. 20 can be eliminated:

$$\begin{cases} \frac{\partial\Psi}{\partial t} + \left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)^{\mathsf{T}}\cdot\nabla\Psi = 0, & \text{in } \Omega, \quad t_o \le t \le t_f \\ \vec{n}_S\cdot\vec{\varphi} = \vec{d}\cdot\vec{n}_S - \psi_{\rho E}(\vec{v}\cdot\vec{n}_S), & \text{on } S, \quad t_o \le t \le t_f \end{cases} \tag{21}$$

where a transpose operation has been performed on the adjoint equations. The variation of the objective function becomes

$$\delta\mathcal{J} = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S \left[\vec{d}\cdot\nabla p + (\nabla\cdot\vec{v})\vartheta + (\vec{v}-\vec{u}_x)\cdot\nabla(\vartheta)\right]\delta S\,ds\,dt = \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\!\!\int_S \frac{\partial\mathcal{J}}{\partial S}\delta S\,ds\,dt, \tag{22}$$

where $\frac{\partial\mathcal{J}}{\partial S} = \vec{d}\cdot\nabla p + (\nabla\cdot\vec{v})\vartheta + (\vec{v}-\vec{u}_x)\cdot\nabla(\vartheta)$ is what we call the *surface sensitivity*. The surface sensitivity provides a measure of the variation of the objective function with respect to infinitesimal variations of the surface shape in the direction of the local surface normal. With each physical time step, this value is computed at every surface node of the numerical grid with negligible computational cost. Apart from the integral over the specified time interval, note that the final expression for the variation involves only a surface integral and has no dependence on the volume mesh.

B. Boundary Conditions for the Continuous Adjoint Problem

Let us return to the integral over the far-field boundary in Eqn. 18 which was assumed to vanish with the appropriate choice of adjoint boundary conditions. This integral will indeed vanish if the scalar integrand is equal to zero at every point on the boundary with each physical time step:

$$\Psi^{\mathsf{T}}\left(\vec{A}-\bar{\bar{I}}\vec{u}_x\right)\cdot\vec{n}_\infty\,\delta U = 0 \tag{23}$$

Therefore, we seek the adjoint state, Ψ, at the boundary that eliminates any contribution from this integral to the variation of the functional through the satisfaction of the preceding expression. A common strategy for the removal of this integral is the imposition of a homogeneous adjoint boundary condition, $\Psi = \mathbf{0}$. While this does force Eqn. 23 to be true, it may be an over-specification and does not take into account the mathematical character of the equations at the boundary. Compatibility between the governing PDEs and the boundary conditions should be maintained in order to avoid issues related to solution accuracy or code convergence. Before further consideration, we will revisit the handling of boundary conditions in the direct problem and the relationship between the direct and adjoint characteristic behavior.

In general for the direct problem, characteristic-based boundary conditions are imposed at the outer boundaries where the fluid states are updated depending on the sign of the characteristic speeds, or eigenvalues, given by the well-known diagonalization of the flux Jacobian matrix, $\vec{A}$. Incoming characteristics correspond to the propagation of information into the flow domain, and at these locations, physical boundary conditions, such as mass flow, stagnation conditions, or back pressure, are prescribed. Outgoing characteristics correspond to information propagation out of the domain, and numerical boundary conditions are imposed at these locations that extrapolate the characteristic variable information from within the domain often using Riemann invariants. The reader is referred to the discussion by Hirsch[17] for further detail on the subject.

It is important to highlight that the sign of the characteristic velocities has been flipped in the adjoint problem, causing characteristic information to propagate in the reverse direction. This sign reversal occurs due to the integration by parts procedure during the adjoint derivation. The result is that the required type of boundary conditions for the direct and adjoint problem are also reversed, i.e. physical conditions in the direct problem become numerical ones in the adjoint problem and vice-versa, and the reversal implies the existence of a complementary set of well-posed conditions for the adjoint problem. This relationship is summarized for inflow and outflow boundaries in Table 1.

Table 1. Number of physical (P) and numerical (N) boundary conditions required for the direct and adjoint problems in three dimensions (the state vector, U, has five components).

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

Based on this idea, Giles and Pierce[18] proposed suitable physical adjoint boundary conditions involving the characteristic speeds based on a manipulation of Eqn. 23 after introducing the diagonalization of the flux Jacobian. Recent work by Hayashi et al.[19] advances the notion of characteristic-based boundary conditions for the adjoint equations by connecting the direction of information propagation to the imposition of physical conditions at the boundaries in the direct problem. More specifically, rather than impose conditions using the characteristic variables, one should consider the realizable perturbations in the flow solution, δU, allowed at the boundaries given the particular choice of physical boundary conditions in the direct problem. The latter derivation approach will now be further detailed and extended in systematic fashion.

Consider the far-field boundary, Γ_∞, to be split into a subsonic inflow and subsonic outflow region for the direct problem in three dimensions. At the inflow, 4 physical conditions can be prescribed, and for clarity in the derivation, we will choose to specify mass flow (ρ and $\vec{v}$) at this boundary. At the outlet boundary, only 1 physical condition can be imposed, and we will choose the typical imposition of back pressure (p). The type of physical conditions chosen for the direct problem has therefore constrained the allowable perturbations at the boundaries: we have prescribed $\delta\rho = \delta\vec{v} = 0$ at the inlet and $\delta p = 0$ at the outlet. Therefore, to derive suitable adjoint boundary conditions for this situation, we introduce these primitive variable perturbations directly into Eqn. 23:

$$\Psi^{\mathsf{T}}\left(\vec{A} - \bar{\bar{I}}\vec{u}_x\right)\cdot\vec{n}_\infty\, M\delta V = 0, \tag{24}$$

where $M = \frac{\partial U}{\partial V}$ is the transformation matrix from conservative to primitive variables and $\delta V = \{\delta\rho, \delta\vec{v}, \delta p\}^{\mathsf{T}}$ is the vector of primitive variable perturbations. After taking the transpose of Eqn. 24 and evaluating $(\vec{A} - \bar{\bar{I}}\vec{u}_x)\cdot\vec{n}_\infty\, M$ in three dimensions, the expanded result can be written as

$$\left\{\begin{array}{c} \delta\rho \\ \delta u \\ \delta v \\ \delta w \\ \delta p \end{array}\right\}^{\mathsf{T}} \left[\begin{array}{ccccc} v_n & uv_n & vv_n & wv_n & \frac{\vec{v}^{\,2}}{2}v_n \\ \rho n_x & \rho u n_x + \rho v_n & \rho v n_x & \rho w n_x & \rho H n_x + \rho u v_n \\ \rho n_y & \rho u n_y & \rho v n_y + \rho v_n & \rho w n_y & \rho H n_y + \rho v v_n \\ \rho n_z & \rho u n_z & \rho v n_z & \rho w n_z + \rho v_n & \rho H n_z + \rho w v_n \\ 0 & n_x & n_y & n_z & \frac{1}{\gamma-1}[\gamma(\vec{v}\cdot\vec{n}) - (\vec{u}_x\cdot\vec{n})] \end{array}\right] \left\{\begin{array}{c} \psi_\rho \\ \psi_{\rho u} \\ \psi_{\rho v} \\ \psi_{\rho w} \\ \psi_{\rho E} \end{array}\right\} = 0, \tag{25}$$

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

For example, at the inlet, the imposition of mass flow forces $\delta\rho = \delta\vec{v} = 0$, and therefore, the terms associated with the products in the first four rows of Eqn. 25 are all zero. As δp is the only allowable perturbation in the flow variables remaining, in order to completely remove contributions from the boundary integral, the following expression must be satisfied,

$$\delta p \left\{ \vec{\phi} \cdot \vec{n}_\infty + \frac{\psi_{\rho E}}{\gamma - 1} [\gamma(\vec{v} \cdot \vec{n}_\infty) - (\vec{u}_x \cdot \vec{n}_\infty)] \right\} = 0, \tag{26}$$

and because δp is arbitrary, we recover the admissible physical boundary condition at the mass flow inlet:

$$\psi_{\rho E} = -\frac{\gamma - 1}{\gamma(\vec{v} \cdot \vec{n}_\infty) - (\vec{u}_x \cdot \vec{n}_\infty)} \, \vec{\phi} \cdot \vec{n}_\infty. \tag{27}$$

Using the same derivation approach, physical boundary conditions for the adjoint variables at the outlet can be found. At the outlet, only $\delta p = 0$, and all other flow variable perturbations are allowable. This eliminates any contributions from the final row of the expression in Eqn. 25, and leaves a system of four equations with five unknowns from the first four rows of Eqn. 25 that must be set equal to zero in order to remove any dependence on the remaining perturbations. Choosing $\psi_{\rho E}$ as the free variable and solving the other four equations in terms of it gives the following four physical boundary conditions:

$$\psi_\rho = \psi_{\rho E} \left[\frac{\gamma \, e \, (\vec{v} \cdot \vec{n}_\infty)}{(\vec{v} - \vec{u}_x) \cdot \vec{n}_\infty} + \frac{\vec{v}^{\,2}}{2} \right], \tag{28}$$

$$\psi_{\rho u} = -\psi_{\rho E} \left[\frac{\gamma \, e \, n_x}{(\vec{v} - \vec{u}_x) \cdot \vec{n}_\infty} + u \right], \tag{29}$$

$$\psi_{\rho v} = -\psi_{\rho E} \left[\frac{\gamma \, e \, n_y}{(\vec{v} - \vec{u}_x) \cdot \vec{n}_\infty} + v \right], \tag{30}$$

$$\psi_{\rho w} = -\psi_{\rho E} \left[\frac{\gamma \, e \, n_z}{(\vec{v} - \vec{u}_x) \cdot \vec{n}_\infty} + w \right], \tag{31}$$

where $e = E - \frac{\vec{v}^{\,2}}{2}$ is the internal energy per unit mass. These conditions are equivalent to those of Hayashi et al. at an outlet if they are reduced to two dimensions in the absence of mesh motion ($\vec{u}_x = 0$). While they also pursue a characteristic form of the adjoint equations to provide numerical boundary conditions, we will choose the values for the remaining free variables ($\vec{\phi}$ and $\psi_{\rho E}$) found at the boundaries that satisfy the discretized adjoint Euler equations within the domain. With this semi-discrete approach, we will be maintaining compatibility between the governing PDEs and the boundary conditions.

The above steps can be repeated as a systematic procedure for finding boundary conditions when other types of physical conditions are imposed in the direct problem (such as prescribing stagnation conditions at an inlet). One must simply introduce the perturbations of the chosen quantities along with the appropriate transformation matrix from the conservative variables into Eqn. 23 and solve for the admissible conditions in the same manner. This procedure could be particularly useful for internal flows that are sensitive to inlet/outlet boundary conditions (often solved on truncated domains) or for flow control applications, such as jets, where sensitivities with respect to inlet/outlet quantities might be desired.

Supersonic inlet and outlet boundaries are straightforward in comparison. At a supersonic inlet, all of the flow variables can be prescribed as physical boundary conditions in the direct problem (all characteristics are incoming) which means that none of the adjoint variables will have prescribed values in the adjoint problem. This approach assumes that $\delta U = 0$ due to the direct problem boundary conditions, and therefore, Eqn. 23 is automatically satisfied. In the case of a supersonic outlet, no flow variables can be specified in the direct problem (all characteristics are outgoing), so all of the adjoint variables can be prescribed with $\Psi = \mathbf{0}$ being the choice that exactly satisfies Eqn. 23.

IV. Numerical Implementation

The following sections contain numerical implementation strategies for each of the major components needed for unsteady aerodynamic shape optimization. The optimal shape design loop requires PDE analysis with dynamic meshes for computing functional and sensitivity information, the definition of suitable design variables for parameterizing the geometry, a mesh deformation algorithm for perturbing the numerical grid after shape changes, and a gradient-based optimizer to drive the design variables toward an optimum for the chosen optimization problem.

All components were implemented within the SU^2 software suite (**S**tanford **U**niversity **U**nstructured).[20] This collection of C++ codes is built specifically for PDE analysis and PDE-constrained optimization on unstructured meshes, and it is particularly well-suited for aerodynamic shape design. Modules for performing flow and adjoint solutions, acquiring gradient information by projecting surface sensitivities into the design space, and mesh deformation techniques are included in the suite, amongst others. Scripts written in the Python programming language are also used to automate execution of the SU^2 suite components, especially for performing shape optimization.

A. Numerical Methods for PDE Analysis

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

The unsteady Euler equations are spatially discretized using a central scheme with JST-type artificial dissipation,[21] and the adjoint equations use a slightly modified, non-conservative JST scheme. Time integration is handled by a second-order accurate dual-time stepping approach for both the analysis and adjoint problems.[22] Again, note that 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.

B. Treatment of Dynamic Meshes with Sliding Interfaces

In the results that follow, only rigid mesh motion will be considered during the solution of unsteady flow and adjoint problems, although mesh deformation techniques will be used during the design process to perturb volume grids after applying surface shape changes. With rigid mesh transformations, there is no relative motion between individual grid nodes, aside from the case of sliding mesh interfaces where two or more rigid sub-grids, or zones, are in motion relative to each other.

1. General Description of Rigid Mesh Motion

Rigid rotational and translational motion for a mesh node with each physical time step can be generally described by

$$\vec{x}^{\,n+1} = \mathfrak{R}\,\vec{x}^{\,n} + \Delta\vec{x}, \tag{32}$$

where $\vec{x}^{\,n} = \{x, y, z\}^{\mathsf{T}}$ is the current node position in Cartesian coordinates, $\vec{x}^{\,n+1}$ is the updated node location at the next physical time instance, $\Delta\vec{x}$ is a vector describing the translation of the nodal coordinates between time steps, and in three dimensions, the rotation matrix, $\mathfrak{R}$, is given by

$$\mathfrak{R} = \begin{bmatrix} \cos\theta_y\cos\theta_z & \sin\theta_x\sin\theta_y\cos\theta_z - \cos\theta_x\sin\theta_z & \cos\theta_x\sin\theta_y\cos\theta_z + \sin\theta_x\sin\theta_z \\ \cos\theta_y\sin\theta_z & \sin\theta_x\sin\theta_y\sin\theta_z + \cos\theta_x\cos\theta_z & \cos\theta_x\sin\theta_y\sin\theta_z - \sin\theta_x\cos\theta_z \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y \end{bmatrix} \tag{33}$$

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. 32 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 $\Delta\vec{x}$ are computed and Eqn. 32 is applied at each node of the mesh.

For all types of rigid mesh motion, the local grid velocity at a node, $\vec{u}_x$, which is needed for solving the ALE form of the governing equations, can be computed by storing the node locations at prior time instances and using a finite differencing approximation that is consistent with the chosen dual time-stepping scheme. For second-order accuracy in time, the mesh velocities are given by

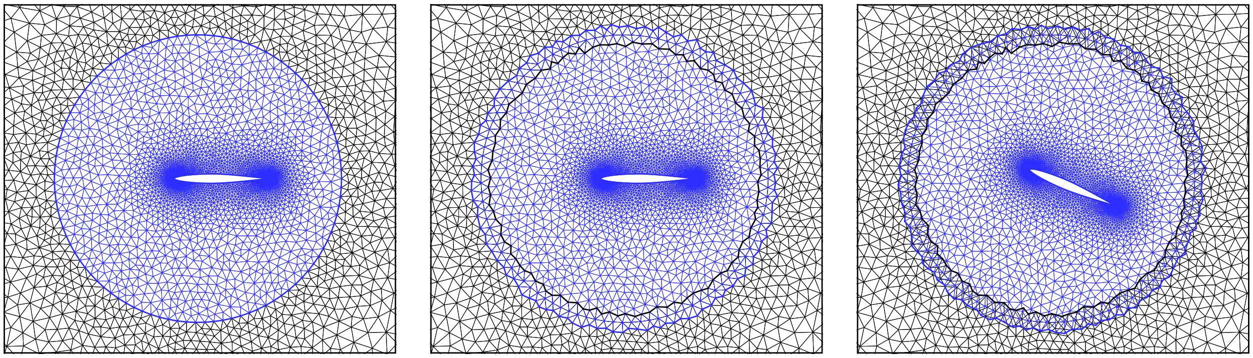
$$\vec{u}_x \approx \frac{3\vec{x}^{\,n+1} - 4\vec{x}^{\,n} + \vec{x}^{\,n-1}}{2\Delta t}, \tag{34}$$

where Δt is the physical time step. 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.

For the two-dimensional pitching airfoil results presented below, the angle of attack as a function of time is given by

$$\alpha(t) = \alpha_o + \alpha_m \sin(\omega t), \tag{35}$$

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, ω_r, is a non-dimensional parameter often specified for consistency between flows. From Eqn. 35, $\Delta\vec{\theta}$ between successive time steps can be found.


(a) Original 1-to1 grid interface.

(b) Overlap region after the creation of halo layers with a preprocessing code.

(c) Sliding motion of the interior zone within the static outer zone. Interpolation is now required between the grids.

Figure 3. Schematics detailing the creation and operation of a sliding mesh interface.

2. *Sliding Mesh Interface Implementation*

A sliding mesh strategy on unstructured grids has been developed and implemented within SU^2 for the simulation of surfaces in relative motion, and it is based on the creation of overlapping halo layers. An arbitrary number of sub-grids, or zones, and their corresponding sliding interfaces can be specified, and a pre-processing code orchestrates the construction of the overlapping regions. The halo layers are constructed by performing nearest neighbor searches for the nodes and elements adjacent to the sliding interface within the donor zone and by adding copies of these nodes and elements to the original zone (requires reconstruction of the grid connectivity information). Fig. 3 depicts this process for a pitching airfoil case where two initial 1-to-1 zones (coincident, repeated nodes) are augmented with the halo layers.

Upon completion of the pre-processing, a new mesh with updated connectivity is produced (corresponding to Fig. 3b). The new mesh file also contains a basic communication structure based on the current MPI implementation. Without any relative motion of the zones during a simulation, the communication defaults to a simple send-receive condition (similar to MPI), as the overlapping nodes remain coincident. Note that only the halo nodes along the outer edges of the overlapping regions participate in the communication.

Once the overlapping regions exist, independent mesh motion can be applied to the zones. However, if relative motion is specified, the overlapping halo layers are no longer coincident, and search and interpolation routines are required to complete the communication of solution information across the interface. Fig. 3c demonstrates the relative motion of a pitching zone sliding within a fixed outer zone. This situation entails the identification of the element in the donor zone that "owns" each halo node along the sliding zone boundary and a suitable interpolation of the solution within the element to the halo node with each physical time step. In order to maintain scalability for massively-parallel simulations, particular attention should be given to advanced search and interpolation techniques.[23]

C. Design Variable Definition, Mesh Deformation, and Optimization

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

In the two-dimensional airfoil calculations that follow, Hicks-Henne bump functions were employed[24] 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

$$f_n(x) = \sin^3(\pi x^{e_n}),\ e_n = \frac{\log(0.5)}{\log(x_n)},\ x \in [0,1], \tag{36}$$

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. After applying the bump functions to recover a new surface shape with each design iteration, a spring analogy method is used to deform the volume mesh around the airfoil.[25]

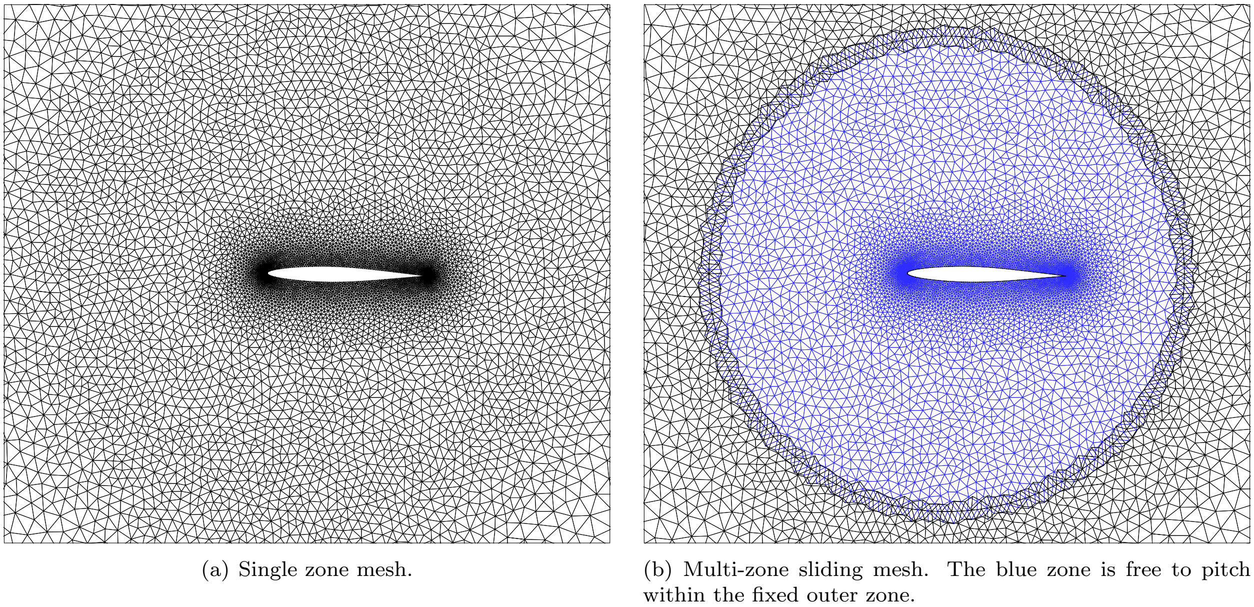
The optimization results presented in this work make use of the *SciPy* library (`http://www.scipy.org`), a well-established, open-source software package for mathematics, science, and engineering. The *SciPy* library provides many user-friendly and efficient numerical routines for the solution of non-linear constrained optimization problems, such as conjugate gradient, Quasi-Newton, or sequential least-squares programming algorithms. At each design iteration, the *SciPy* routines require as input only the values and gradients of the objective functions, computed by means of our continuous adjoint approach, as well as the set of any chosen constraints.

V. Numerical Results

For the numerical results that follow, the NACA 64A010 airfoil was chosen as the initial geometry. This airfoil has been widely studied for unsteady problems, and thus allows for meaningful comparison with existing experimental and numerical results. A baseline unstructured mesh consisting of 20,676 triangular elements, 10,488 nodes, 200 edges along the airfoil, and 100 edges along the far-field boundary was constructed around the airfoil. In order to analyze the performance of the new sliding mesh capability, the same baseline mesh was then divided into two separate zones: a small, circular zone around the airfoil (radius of 1.5c), and a background zone extending to the far-field. The creation of the overlapping halo layers added a total of 339 new nodes to the mesh. As we will be considering pitching motions for this airfoil, the two meshes are shown in Fig. 4 in the maximum pitch angle state.

A. Verification and Validation of the Unsteady Flow and Adjoint Equations

For validating our implementation of the unsteady Euler equations in ALE form, a comparison was made against the well-known CT6 data set of Davis.[26] 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, w_r, of 0.202, $M_\infty = 0.796$, a mean angle of attack of 0 degrees, and a maximum pitch angle of 1.01 degrees. All numerical simulations were performed with 25 times steps per period for a total of 10 periods. Fig. 5 contains a comparison of the pressure contours



(a) Single zone mesh.

(b) Multi-zone sliding mesh. The blue zone is free to pitch within the fixed outer zone.

Figure 4. Views of the baseline and sliding meshes for the NACA 64A010 at $\alpha = 1.01$ degrees.

for $\alpha = 1.01$ degrees both with and without the sliding mesh interface. Fig. 5 also shows a comparison of the lift coefficient versus angle of attack between SU^2 and experiment 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 baseline and sliding mesh cases produce essentially identical results, suggesting that for this problem, the sliding mesh approach does not introduce considerable error. The numerical results agree well with experimentally measured values and also compare favorably with other inviscid results.

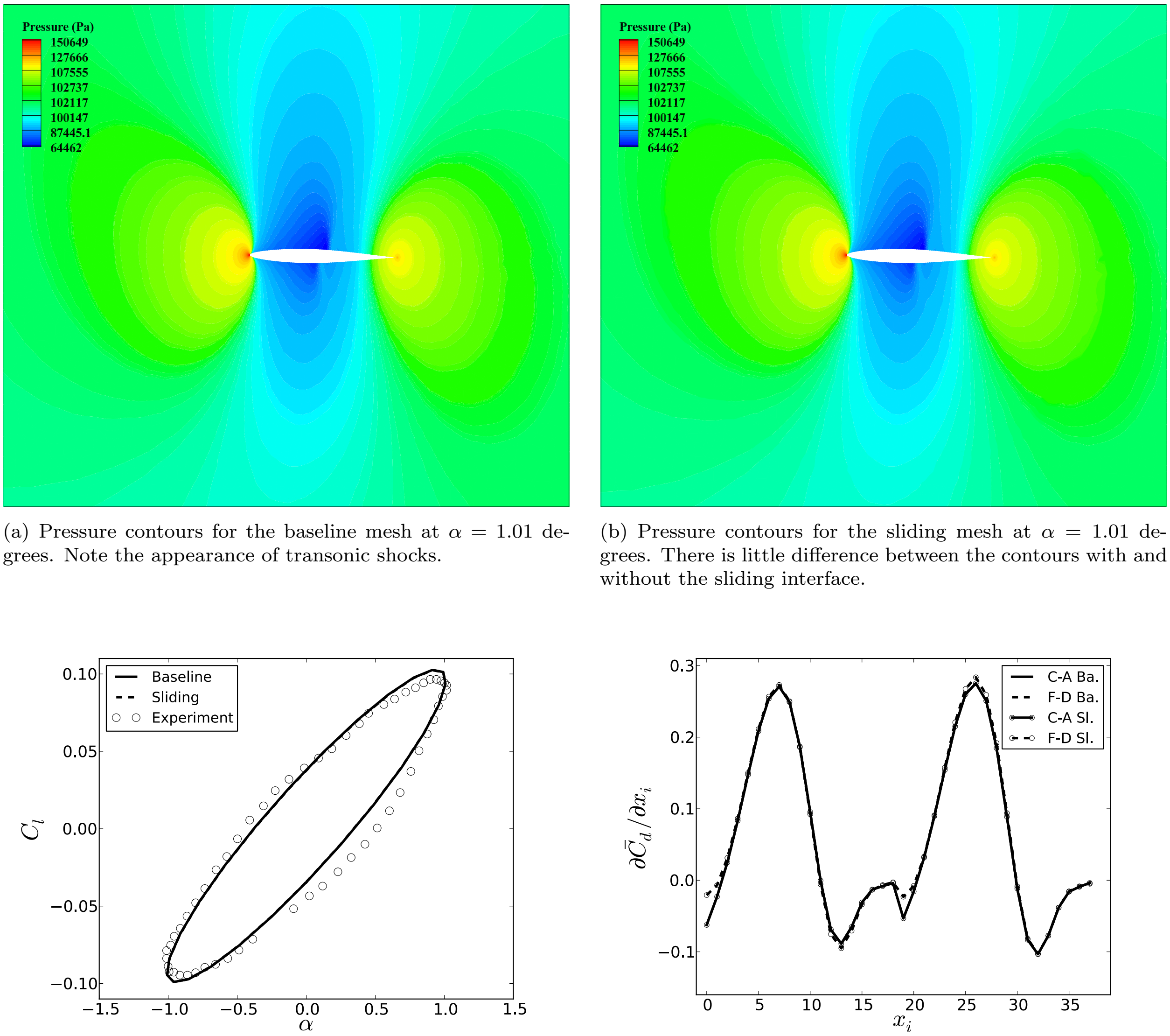
In order to verify the accuracy of the gradient information obtained via the time-accurate adjoint, 38 Hicks-Henne bump functions were chosen as design variables. The bumps were equally spaced ($0.05c$) along the upper and lower surfaces of the NACA 64A010. After solving the adjoint equations using the stored solution data from the numerical experiment performed above, a comparison was made between the time-averaged drag gradients (with respect to the Hicks-Henne design variables) calculated using both the continuous adjoint and a finite differencing approach with small step sizes for the bump deformations. This procedure was performed for both the baseline and sliding mesh cases. As seen in Fig. 5, all gradients compare very favorably, verifying both the time-accurate continuous adjoint and the sliding mesh algorithm.

Table 2. Initial and final time-averaged drag coefficient values for the baseline and sliding mesh design cases.

	Initial $\bar{C}_d$	Final $\bar{C}_d$	$\bar{C}_d$ Reduction (%)
Baseline	0.002526	0.001022	59.5
Sliding	0.002532	0.001000	60.5

B. Redesign of the NACA 64A010 Airfoil

Lastly, a redesign of the pitching NACA 64A010 airfoil was performed using the gradient information obtained from the time-accurate adjoint formulation on both the baseline and sliding meshes. The specific shape optimization problem was for unconstrained, time-averaged drag minimization using the Hicks-Henne design variables. Upon completion, the time-averaged drag, $\bar{C}_d$, was successfully reduced by 59.5 % for the baseline case and 60.5 % for the sliding case. Table 2 contains drag coefficient details during the optimization process. Profile shapes of the initial and final designs are compared in Fig. 6. Again, note that the baseline and sliding cases result in very similar results. While the optimization results given here are for an unconstrained



(a) Pressure contours for the baseline mesh at $\alpha = 1.01$ degrees. Note the appearance of transonic shocks.

(b) Pressure contours for the sliding mesh at $\alpha = 1.01$ degrees. There is little difference between the contours with and without the sliding interface.

(c) Coefficient of lift versus angle of attack (degrees) for the baseline and sliding meshes compared against experimental data. Note that nonlinear effects cause lift hysteresis.

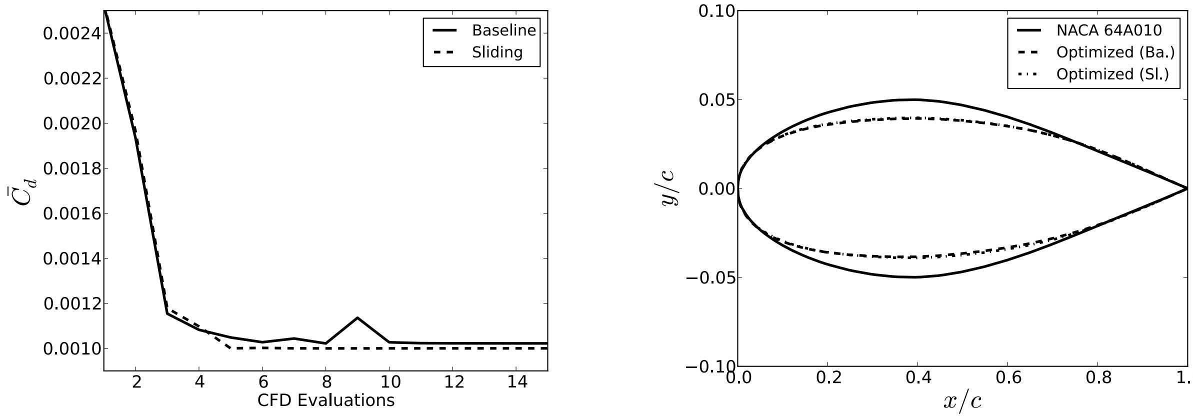
(d) Direct comparison of the time-averaged drag gradients obtained by the continuous adjoint (C-A) and finite differencing (F-D) on the baseline (Ba.) and sliding (Sl.) meshes.

Figure 5. Verification and validation results for a pitching NACA 64A010.

problem, constraints on the lift or moment, for instance, can be easily added in the optimization framework by solving an additional adjoint problem for each.

VI. Conclusions

In this article, a time-accurate, continuous adjoint formulation has been presented, verified, and applied. This formulation allows for the efficient design of aerodynamic surfaces in unsteady flows using a gradient-based optimization framework. More specifically, the treatment given in this article is a systematic methodology for the unsteady Euler equations centered around finding *surface sensitivities* with the use of differential geometry formulas which has no dependence on volume mesh sensitivities when computing the first variation of a functional (only a surface integral remains). A systematic method for deriving suitable far-field boundary conditions for the adjoint problem was also demonstrated. Finally, a new sliding mesh algorithm for unstructured meshes was developed and tested. In conjunction with the unsteady adjoint



(a) Optimization history of a time-averaged drag minimization for a pitching airfoil. The two cases achieved a final time-averaged drag within 1 % of each other.

(b) Profile shape comparison for the initial NACA 64A010 and the minimum time-averaged drag airfoils resulting from the baseline and sliding mesh cases. The two optimized profiles are nearly identical.

Figure 6. Optimal shape design results for a pitching airfoil.

formulation, the sliding mesh capability allows for the design of surfaces in relative motion.

The gradient information provided by the time-accurate adjoint has been verified for design variables in two dimensions with and without the use of sliding meshes. In both situations, the gradients compared very favorably with those obtained via finite differencing. Optimal shape design results for a pitching airfoil demonstrated the effectiveness of the unsteady adjoint approach. Furthermore, shape design with sliding meshes recovered nearly identical results as the baseline case, suggesting that their use in conjunction with the unsteady adjoint can provide a stable and accurate design approach. The combination of sliding mesh interfaces and the time-accurate continuous adjoint produces a powerful design tool for unsteady problems with surfaces in relative motion.

In the future, these capabilities will be further developed within the current framework in order to tackle larger, more challenging applications where unsteady effects are important and sliding meshes are required, such as the design of counter-rotating open rotors, advanced rotorcraft design, or turbomachinery component design.

VII. Acknowledgements

T. Economon would like to acknowledge Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

References

[1] van der Weide, E., Kalitzin, G., Schluter, J., Alonso, J. J., "Unsteady Turbomachinery Computations Using Massively Parallel Platforms," *AIAA-2006-421*, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 9-12, 2006.

[2] Blades, E. L., Marcum, D. L., "A sliding interface method for unsteady unstructured flow simulations," *International Journal for Numerical Methods in Fluids*, Vol. 53, 2007, pp. 507-529.

[3] Steijl, R. and Barakos, G., "Sliding mesh algorithm for CFD analysis of helicopter rotorfuselage aerodynamics," *International Journal for Numerical Methods in Fluids*, Vol. 58, 2008, pp. 527-549.

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

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

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

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

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

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

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

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

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

[13] Economon, T. D., Palacios, F., Alonso, J. J., “A Coupled-Adjoint Method for Aerodynamic and Aeroacoustic Optimization,” *AIAA-2012-5598*, 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM Multidisciplinary Analysis and Optimization Conference, Indianapolis, Indiana, September, 2012.

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

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

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

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

[18] Giles, M. B., Pierce, N. A., “Adjoint Equations in CFD: duality, boundary conditions, and solution behavior,” *AIAA Paper 1997-1850*, 1997.

[19] Hayashi, M., Ceze, M., Volpe, E., “Characteristics-based boundary conditions for the Euler adjoint problem,” *International Journal for Numerical Methods in Fluids*, 2012.

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

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

[22] Jameson, A., “Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings,” *AIAA 91-1596*, 10th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24-26, 1991.

[23] Alonso, J. J., Hahn, S., Ham, F., Herrmann, M., Iaccarino, G. and van der Weide, E., “CHIMPS: A High- Performance Scalable Module for Multi-Physics Simulations,” *AIAA Paper 2006-5274*, 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Sacramento, CA, July 2006.

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

[25] Degand, C. and Farhat, C., “A three-dimensional torsional spring analogy method for unstructured dynamic meshes, *Computers & Structures*, Vol. 80, pp. 305-316, 2002.

[26] Davis, S. S., “NACA 64A010 (NASA Ames model) Oscillatory Pitching, Compendium of Unsteady Aerodynamic Measurements, AGARD, Rept. R-702, Neuilly sur-Seine, France, Aug. 1982.