AIAA Aviation
22-26 June 2015, Dallas, TX
16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

The 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 22 – 26 June 2015, Dallas, Texas

A Discrete Adjoint Framework for Unsteady Aerodynamic and Aeroacoustic Optimization

Beckett Y. Zhou,* Tim Albring,† and Nicolas R. Gauger‡
Chair for Scientific Computing, TU Kaiserslautern
Bldg 34, Paul-Ehrlich-Strasse, 67663 Kaiserslautern, Germany

Thomas D. Economon,§ Francisco Palacios,¶ and Juan J. Alonso ‖
Department of Aeronautics and Astronautics, Stanford University
Stanford, CA 94305, U.S.A.

In this paper, we present an unsteady aerodynamic and aeroacoustic optimization framework in which algorithmic differentiation (AD) is applied to the open-source multi-physics solver SU2 to obtain design sensitivities. An AD-based consistent discrete adjoint solver is developed which directly inherits the convergence properties of the primal flow solver due to the differentiation of the entire nonlinear fixed-point iterator. In addition, a coupled CFD-CAA far-field noise prediction framework using a permeable surface Ffowcs Williams-Hawkings approach is also developed. The resultant AD-based discrete adjoint solver is applied to both drag and noise minimization problems. The results suggest that the unsteady adjoint information provided by this AD-based discrete adjoint framework is accurate and robust, due to the algorithmic differentiation of the entire design chain including the dynamic mesh movement routine and various turbulence model, as well as the hybrid CFD-CAA model.

I. Introduction

The past several decades have seen significant progresses in the numerical methods for the design and optimization of aerospace systems. Since the advent of adjoint-based methods,[1,2] for which the computational cost is independent of the number of design variables, researchers have been able to tackle many large-scale and practical problems, such as aerodynamic and aerostructural optimizations of complete aircraft configurations.[3,4] In most applications, the underlying physical problem is considered to be in a steady state, as is evident from the rich body of literatures on the topic of aerodynamic shape optimization with steady Euler and Reynolds-averaged Navier-Stokes equations (RANS).

However, many aerospace problems are unsteady in nature, such as active flow control, turbomachinery, aeroelastic flutter, biologically-inspired flight and aeroacoustics. In comparison to the large strides made in its steady counterpart, unsteady adjoint-based optimization has not received as much attention and methods available to address such problem are consequently less mature. This has been primarily due to the need to store prohibitively large amount of solution data required to solve the unsteady adjoint equation. Furthermore, many unsteady problems involve moving aerodynamic surfaces. The need to accurately account for the requisite mesh movement in the governing equations and the accompanying adjoint equations introduces additional difficulties.

Nevertheless, with the growth of computing power and the improvement of time-accurate numerical methods, has led to more research effort in this area over the past decade. The renewed interest in reduction of aircraft noise – an inherently unsteady phenomenon, due to the ever-stringent aviation noise regulations, also served as a necessary catalyst. Rumpfkeil and Zingg developed a discrete adjoint formulation for unsteady aerodynamic shape optimization and noise minimization based on 2D unsteady RANS (URANS),[5] Mani and Mavriplis[6] presented a discrete adjoint

*Ph.D. Candidate, Student Member AIAA, yuxiang.zhou@scicomp.uni-kl.de
†Ph.D. Candidate, tim.albring@scicomp.uni-kl.de
‡Professor, Senior Member AIAA, nicolas.gauger@scicomp.uni-kl.de
§Ph.D. Candidate, Student Member AIAA, economon@stanford.edu
¶Engineering Research Associate, Senior Member AIAA, fpalacios@stanford.edu
‖Associate Professor, Associate Fellow AIAA, jjalonso@stanford.edu

framework with deforming meshes for unsteady aerodynamic optimization of pitching airfoils. Nielsen and Diskin[7] applied their discrete adjoint methodology with dynamic overset meshes to a large-scale complex helicopter configuration. Most recently, Economon et al.[8,9] developed a continuous adjoint framework for unsteady aerodynamic and aeroacoustic optimization.

Under the discrete framework, the effort of the adjoint implementation can be eased by the use of automatic (or algorithmic) differentiation (AD), replacing the laborious and error-prone hand-differentiation of the discretized equations. This however, is not the only advantage of AD-based adjoint. By construction, AD-based adjoints are accurate to machine precision as they do not incur any roundoff or truncation error. In addition, the *frozen turbulence* assumption typically used in many URANS-based adjoint formulations is eliminated, since the turbulence models while not analytically differentiable are still *algorithmically* differentiable. The application of AD also leads to robust and consistent adjoints in that the adjoint solver inherit the same convergence properties as the primal solver. Lastly, if the operator overloading AD method is used with expression template technique in C++,[10] the resultant discrete adjoint framework offers extra flexibility – the adjoint solver can be automatically updated with primal code modification and one can easily define any objective function from any state variable. This is an extremely attractive characteristic for unsteady optimization problems in the multidisciplinary setting using a suite of multi-physics solvers where the objective function may be different depending on the type of problems being addressed. AD-based discrete adjoint has been applied to steady aerodynamic shape design problems by Bischof et al.[11] and Gauger et al.[12] More recently, it was successfully applied by Nemeli et al.[13] in the unsteady setting to the active flow control of an industry-relevant high-lift configuration HIREX from Airbus.

In this paper, we present the development of an AD-based discrete adjoint framework on an open source multi-physics suite SU2[14] for applications to unsteady aerodynamic and aeroacoustic optimization problems. The remainder of the paper is organized as follows. In Section 2, the unsteady SU2 solver including a coupled CFD-CAA model using a permeable surface Ffowcs Williams-Hawkings approach as well as the optimization framework based on discrete adjoint and AD are presented. Optimization results are presented in Section 3 while conclusions and directions for future activities are outlined in Section 4.

II. Unsteady Optimization Framework

II.A. Unsteady Multi-Physics Solver SU2

The Stanford University Unstructured (SU2) open source software suite was specifically developed for solving problems governed by partial differential equations (PDEs) and PDE-constraned optimization problems. It was developed with the aerodynamic shape optimization problems in mind. Therefore the suite is centered around a RANS solver capable of simulating compressible, turbulent flows commonly found in problems in aerospace engineering. The governing equations are spatially discretized using the finite volume method, on unstructured meshes. A number of convective fluxes discretization schemes have been implemented, such as the Jameson-Schimdt-Turkel (JST) scheme and the upwind Roe scheme. The turbulence can be either modeled by the Spalart-Allmaras(S-A) model or the Menter Shear Stress Transport (SST) Model. For unsteady flows, a dual time-stepping method can be used to obtain time-accurate solutions. SU2 suite has recently seen extensions in the multi-disciplinary setting such as the inclusion of a wave equation solver and a structural solver, making it well-suited for the unsteady multi-physics problems considered in this work.

For the sake of conciseness, details regarding the formulations and implementations of the SU2 solver suite will not be presented in this paper. Further details such as mesh deformation, dynamic mesh movement, multigrid implementations, validation and verification cases, as well as the continuous adjoint framework developed in tandem with SU2, the readers are referred to the published work by Palacios et al.[9,14] of the SU2 team. In the next section we direct the attention of the readers to a new coupled CFD-CAA far-field noise prediction framework developed for the current work.

II.B. Coupled CFD-CAA Noise Prediction using a Permeable Surface Ffowcs Williams-Hawkings Approach

It is known for turbulent flows at low Mach numbers, direct computations of far-field noise is computationally prohibitive.[15] A common way to perform far-field noise prediction is then to adopt a hybrid CFD-CAA approach in which the near-field noise source region is computed using a high-fidelity CFD model and then propagated to the far-field using a computationally cheaper wave equation like CAA model. To that end, integral methods based on the Kirchhoff or Ffowcs Williams-Hawkings (FW-H) formulations offer a more efficient approach for calculating acoustic pressure at arbitrary observer locations by performing boundary integrals once the appropriate field data is known.

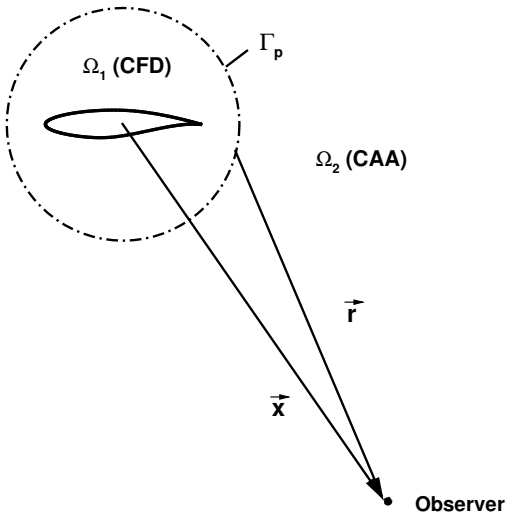


Figure 1. Permeable control surface Γ_p separating the CFD and CAA domains

In this manner, the radiated noise from a complex system can be calculated given the near-field flow data supplied by a CFD solution. In this work, we develop a coupled CFD-CAA far-field noise prediction framework using a permeable surface Ffowcs Williams-Hawkings approach.

The permeable FW-H formulation distinguishes itself from its original formulation in that it allows fluid to flow through the discontinuity surface. Consequently, one can define any arbitrary smooth surrounding surface Γ_p around the aerodynamic body S where details of the flow field are extracted and the noise source can be propagated to the far-field. A schematic of such permeable surface is shown on Figure 1. The fluid domain is therefore divided into two regions – the near-field CFD region Ω_1 and far-field CAA region Ω_2. Further, we define the shape of Γ_p by a function, $f = 0$, such that $f < 0$ inside the control surface and $f > 0$ outside the control surface.

The key difference between this work and an earlier work by Economon et al.[8] in that in that work the permeable FW-H formulation in the wave equation form is solved using a finite element method (FEM). In this work we follow the boundary integral formulation presented by Di Francescantonio,[16] which rewrites the original FW-H into the permeable surface form as follows:

$$\Box^2[c^2(\rho-\rho_o)] = \frac{\partial}{\partial t}[\rho_o U_n \delta(f)] - \frac{\partial}{\partial x_i}[L'_{ij} n_j \delta(f)] + \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j}. \tag{1}$$

where

$$U_i = u_i + [(\rho/\rho_o) - 1](u_i - v_i) \tag{2}$$

$$L_{ij} = P_{ij} - p_o\delta_{ij} + \rho u_i(u_j - v_j), \tag{3}$$

$$T_{ij} = P'_{ij} + \rho u_i u_j - c^2(\rho-\rho_o)\delta_{ij}. \tag{4}$$

Note that ρ is the density, $u_{i,j,k}$ is the flow velocity, $v_{i,j,k}$ is the velocity of the control surface Γ_p. The subscript o denotes ambient conditions while the superscript $'$ denotes perturbation values. P_{ij} is the compressive stress tensor.

By convoluting with a free space Green's function, a boundary integral form can be obtained. Following the derivation of Farassat's formulation 1A,[17] time derivatives with respect to observer time can be moved inside of the integrals and adjusted to be with respect to source time. This manipulation can offer some benefits during numerical implementation. Lastly, by assuming a non-deforming surface which is far enough away from the source to contain all non-linearities, the quadrupole volume terms can be removed, and the following boundary integral expression is recovered for the pressure at an arbitrary observer location,

$$\begin{aligned} 4\pi p'_{obs} = &\int_{\Gamma_p}\left[\frac{\rho_o(\dot{U}_i n_i + U_i \dot{n}_i)}{r|1-M_r|^2}\right]_{ret} d\Gamma_p + \int_{\Gamma_p}\left[\frac{\rho_o U_i n_i K}{r^2|1-M_r|^3}\right]_{ret} d\Gamma_p + \frac{1}{c}\int_{\Gamma_p}\left[\frac{\dot{F}_i \hat{r}_i}{r|1-M_r|^2}\right]_{ret} d\Gamma_p \\ &+ \int_{\Gamma_p}\left[\frac{F_i\hat{r}_i - F_i M_i}{r^2|1-M_r|^2}\right]_{ret} d\Gamma_p + \frac{1}{c}\int_{\Gamma_p}\left[\frac{F_i \hat{r}_i K}{r^2|1-M_r|^3}\right]_{ret} d\Gamma_p. \end{aligned} \tag{5}$$

where $K = \dot{M}_i\hat{r}_i r + M_r c - M^2 c$ and $F_i = L_{ij}n_j$. For a stationary permeable surface, all terms involving M or M_r become zero and Equation 5 simplifies to:

$$4\pi p'_{obs} = \int_{\Gamma_p}\left[\frac{\rho_o \dot{U}_n}{r}\right]_{ret} d\Gamma_p + \frac{1}{c}\int_{\Gamma_p}\left[\frac{\dot{F}_r}{r}\right]_{ret} d\Gamma_p + \int_{\Gamma_p}\left[\frac{F_r}{r^2}\right]_{ret} d\Gamma_p. \tag{6}$$

Note that the AD-based discrete adjoint framework is particularly well suited for such multi-module computational design chain. Whereas the continuous adjoint framework requires cumbersome derivations of new adjoint boundary conditions at the interface between the CFD and CAA domains, AD-based discrete adjoint simply differentiates through the entire coupled CFD-CAA program algorithmically, giving adjoint information relating the far-field noise objective function and the shape variables directly.

II.C. AD-based Unsteady Discrete Adjoint Framework

The implementation of the discrete adjoint formulation in this work is eased by the use of automatic differentiation (AD) [a], eliminating the error-prone hand-differentiation of the discretized equations. AD was developed based on the observation that any simulation code, regardless of its complexity is merely a sequence of elementary operations whose differentiation rules are well known. Therefore, by successive applications of the chain-rule through the computer program, it is possible to compute both the simulation output and its derivative with respect to prescribed design variables simultaneously. A remarkable feature of AD, owing to its construction, is that it does not incur any truncation errors compared to the traditional finite difference method. In particular, the derivatives are accurate to machine accuracy. This is a very attractive characteristic of AD, since accurate evaluation of the gradient requires exact differentiation of the fixed point iterator G^n as evidenced by Equations 20 and 21 in the following discussion.

The AD can be performed in the forward and reverse (adjoint) mode. The forward mode, albeit exact, requires one evaluation for each component of the gradient vector. In contrast, the reverse mode is capable of computing the entire gradient vector in one stroke of the forward and reverse simulation in time. For this reason, the reverse mode is also referred to as the adjoint mode. The one-stroke gradient computation of the reverse mode AD is achieved at the expense of high memory requirement due to the need to save all intermediate variables in an unsteady computation. For this reason, wherever the reverse mode is used in this study, it is always implemented in conjunction with the memory-saving checkpointing technique, which stores the flow solution at certain points in time during the forward sweep as *checkpoints*. The flow solutions are then recomputed from these checkpoints in the reverse sweep for adjoint variables. The maximum number of time steps allowable between the two consecutive checkpoints is dictated by the memory available at each core. For long time computations, the many checkpoints required results in high hard disk storage requirement and an increase in run time due to the need to recompute the flow solutions as well as reading and writing of large volumes of data at each checkpoint. A good discussion of the two AD modes along with checkpointing techniques can be found in.[18,19]

Now we present our AD-based unsteady discrete adjoint framework using a simple system of PDEs as an example. For the sake of brevity, influences of the mesh are omitted. Consider a system of semi-discretized PDEs as follows:

$$\frac{dU}{dt} + R(U) = 0 \tag{7}$$

where U is the spatially discretized state vector and $R(U)$ is the discrete spatial residual vector. For the sake of illustration, we assume the second-order backward difference is used for time discretization, which leads to the following system of equations:

$$R^*(U^n) = \frac{3}{2\Delta t}U^n + R(U^n) - \frac{2}{\Delta t}U^{n-1} + \frac{1}{2\Delta t}U^{n-2} = 0, \qquad n = 1,\ldots,N \tag{8}$$

The application of dual-time stepping method then solves the following problem through a fictitious time τ to converge to a steady state solution in (8):

$$\frac{dU^n}{d\tau} + R^*(U^n) = 0 \tag{9}$$

Further assume the implicit Euler method is used to time march the above equation to steady state.

$$U^n_{p+1} - U^n_p + \Delta\tau R^*(U^n_{p+1}) = 0, \qquad p = 1,\ldots,M \tag{10}$$

The resultant nonlinear system can be linearized around U^n_p to solve for the state U^n_{p+1}

$$U^n_{p+1} - U^n_p + \Delta\tau\left[R^*(U^n_p) + \left.\frac{\partial R^*}{\partial U}\right|^n_p (U^n_{p+1} - U^n_p)\right] = 0, \qquad p = 1,\ldots,M \tag{11}$$

This can be written in the form of a fixed-point iteration:

$$U^n_{p+1} = G^n(U^n_p, U^{n-1}, U^{n-2}), \qquad p = 1,\ldots,M, \quad n = 1,\ldots,N \tag{12}$$

where G^n represents an iteration of the pseudo time stepping. U^{n-1} and U^{n-2} are the converged state vectors at time iterations $n-1$ and $n-2$ respectively. The fixed point iteration converges to the numerical solution U^n:

$$U^n = G^n(U^n, U^{n-2}, U^{n-2}), \qquad n = 1,\ldots,N \tag{13}$$

[a] performed using AD tool ADEPT[10]

The discrete optimization problem can then be posed as:

$$\min_{\alpha} \quad J = \frac{1}{N}\sum_{n=1}^{N}\hat{J}(U^n,\alpha) \tag{14}$$

$$\text{subject to} \quad U^n = G^n(U^n,U^{n-1},U^{n-2},\alpha), \qquad n=1,\ldots,N \tag{15}$$

where α is the vector of design variables. One can express the Lagrangian associated with the above constrained optimization problem as follows:

$$L = \frac{1}{N}\sum_{n=1}^{N}\hat{J}(U^n,\alpha) - \sum_{n=1}^{N}[(\bar{U}^n)^T(U^n - G^n(U^n,U^{n-1},U^{n-2},\alpha))] \tag{16}$$

where $\bar{U}^n$is the adjoint state vector at time level n. The first order optimality conditions are given by:

$$\frac{\partial L}{\partial \bar{U}^n} = 0, \quad n=1,\ldots,N \qquad \text{(State equations)} \tag{17}$$

$$\frac{\partial L}{\partial U^n} = 0, \quad n=1,\ldots,N \qquad \text{(Adjoint equations)} \tag{18}$$

$$\frac{\partial L}{\partial \alpha} = 0, \qquad \text{(Control equation)} \tag{19}$$

From (18), the unsteady discrete adjoint equations can be derived in the fixed point form as:

$$\bar{U}^n_{i+1} = \left(\frac{\partial G^n}{\partial U^n}\right)^T \bar{U}^n_i + \left(\frac{\partial G^{n+1}}{\partial U^n}\right)^T \bar{U}^{n+1} + \left(\frac{\partial G^{n+2}}{\partial U^n}\right)^T \bar{U}^{n+2} + \frac{1}{N}\left(\frac{\partial \hat{J}^n}{\partial U^n}\right)^T, \; n=N,\ldots,1 \tag{20}$$

where $\bar{U}^{n+1}$ and $\bar{U}^{n+2}$ are converged adjoint state vectors at time levels $n+1$ and $n+2$. The unsteady adjoint equations above are solved backward in time. At each time level n we iterate through inner iteration i until we have converged to $\bar{U}^n$. The highlighted terms here are evaluated in reverse mode AD at each iteration. To do so, reverse accumulation[18] is performed at the beginning of each time level n to store the computational graph by evaluating G using converged state solution U_n. Then each inner iteration i proceeds by re-evaulating the tape using the updated adjoint vector $\bar{U}^n_i$, giving the highlighted terms. This continues within each time level n until the adjoint vector as converged to $\bar{U}^n$. The sensitivity gradient can then be computed from the adjoint solutions:

$$\frac{dL}{d\alpha} = \sum_{n=1}^{N}\left(\frac{1}{N}\frac{\partial \hat{J}^n}{\partial \alpha} + (\bar{U}^n)^T\frac{\partial G^n}{\partial \alpha}\right) \tag{21}$$

In this work, we apply the operator overloading AD method with expression template technique[10] to the SU^2 suite. This leads to additional flexitbilities in the resultant discrete adjoint framework – the adjoint solver can be automatically updated with primal code modification and one can easily define any objective function from any state variable. This is an extremely attractive characteristic for unsteady optimization problems in the multidisciplinary setting using a suite of multi-physics solvers where the objective function may be different depending on the type of problems being addressed.

III. Results

The SU2 suite has been differentiated using AD in both forward and reverse (adjoint) modes. The table below demonstrates that the two unsteady gradients match well. Furthermore, checkpointing and reverse accumulation have also been implemented for the reverse mode at each time iteration. In the next two subsections, we present optimization results from drag and noise minimization problems involving dynamic mesh movement routine and various turbulence model, as well as the hybrid CFD-CAA design chain – all of which can be cumbersome and error-tasks using the continuous adjoint and hand-discrete approaches.

III.A. Lift-Constrained Drag Minimization of a Pitching Airfoil in Transonic Turbulent Flow

In this test case, we consider the lift-constrained drag minimization problem of an RAE2822 airfoil in transonic ($M_\infty = 0.8$) and turbulent ($Re = 6.5\times10^6$) flow regime, pitching about its quarter-chord point with a mean angle of attack of 2.0 degrees, an amplitude of 3.0 degrees and a frequency of 251.3 rad/s.

i	Forward AD	Reverse AD
1	1.04680400655973	1.046804006560 03
2	0.469600562524447	0.469600562524395
3	0.246014397617865	0.246014397617779
4	0.128741852545148	0.128741852545171
5	1.0501672436756	1.0501672436755

Table 1. Comparison between the gradients forward-mode and reverse modes of AD

The mesh used is an un-structured, O-grid that wraps around the RAE 2822 airfoil. It has 22,842 elements in total with 192 edges making up the airfoil boundary and 40 edges along the far field boundary. It is a hybrid-element mesh with quadrilaterals in the region adjacent to the airfoil surface and triangles in the remaining portion of the computational domain. The first grid point of the airfoil surface is at a distance of 10^{-5} chord, and the far field boundary is located approximately one hundred chord lengths away from the airfoil. Characteristic-based far-field boundary conditions are enforced on the far-field boundary, and a no-slip, adiabatic boundary condition is enforced on the airfoil. The convective fluxes are computed using the Jameson-Schmidt-Turkel (JST) scheme while the turbulent viscosity is calculated using the Menter Shear Stress Transport (SST) Model. A time step of 0.001 is used for the dual time-stepping, resulting in 25 time steps per period of oscillation for a total of 8 periods.

A total of 38 Hicks-Henne bump functions are chosen as design variables and they are equally spaced along the upper and lower surfaces of the airfoil (19 on each of the upper and lower surfaces). The objective function is defined as the time-averaged drag within the periodic steady state:

$$J_D = \bar{C}_d = \frac{1}{N - N^*} \sum_{n=N^*+1}^{N} C_d^n \tag{22}$$

where C_d^n is the instantaneous drag coefficient at time level n. N is the total number of time steps and N^* is the number of time steps before entering the optimization window in which $\bar{C}_d$ is to be minimized as shown on Figure 5. In this case $N^* = 50$ and the optimization window contains 6 periods of oscillation. The time-average lift $\bar{C}_l$ is constrained in the optimization so that $\bar{C}_l^{Optimized} = \bar{C}_l^{RAE2822} = 0.4$. A volume constraint is also imposed to ensure that the volume of the optimized airfoil remains at least as large as the baseline RAE2822 airfoil.

The time-averaged drag is reduced by 38% after 24 CFD evaluations, while the mean lift is maintained at $\bar{C}_l = 0.4$, as shown on Figure 2 (a). A comparison between the baseline RAE2822 and the new airfoil profiles is shown on Figure 2 (b). Note that the curvature in the first 50% chord has been drastically reduced in the optimized airfoil, since the airfoil is pitching at a positive mean angle of attack. The effect can be seen by comparing the Mach number contour of the airfoils at their mean (Figure 3 (b) and Figure 4 (b)) and maximum (Figure 3 (c) and Figure 4 (c)) angles of attack – the shock strength on the upper surface has been significantly reduced, although the presence of the mean lift and volume constraints prevents the shock from being fully removed. Note that at all three angles shown on Figure 3 and Figure 4, the shock position has moved further downstream in the 'improved airfoil', reducing the extent of shock-induced separation on the airfoil surface. Consequently, the new airfoil achieves lower drag within the optimization window as shown on Figure 5. The time-averaged drag is reduced by 38% from the baseline value.

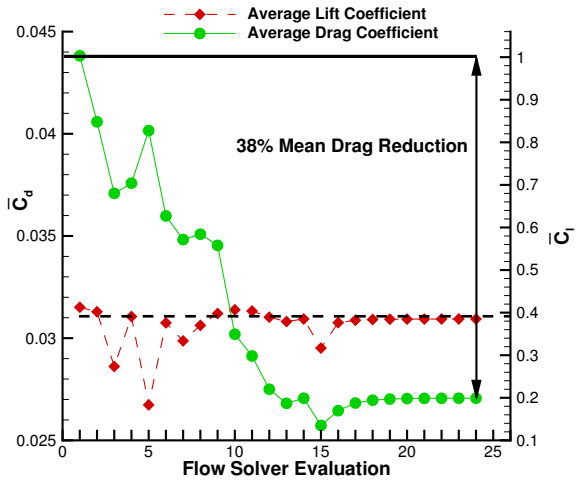
III.B. Lift-Constrained Drag and Noise Minimization of a Pitching Airfoil in Transonic Inviscid Flow

In this test case, we apply the AD-based discrete adjoint framework to the lift-constrained drag and noise minimization problem of a pitching airfoil in transonic ($M_\infty = 0.796$) inviscid flow. The baseline geometry considered is the NACA64A010 airfoil pitching about its quarter-chord point with a mean angle of attack of 0 degrees, an amplitude of 1.01 degrees and a frequency of 251.3 rad/s.

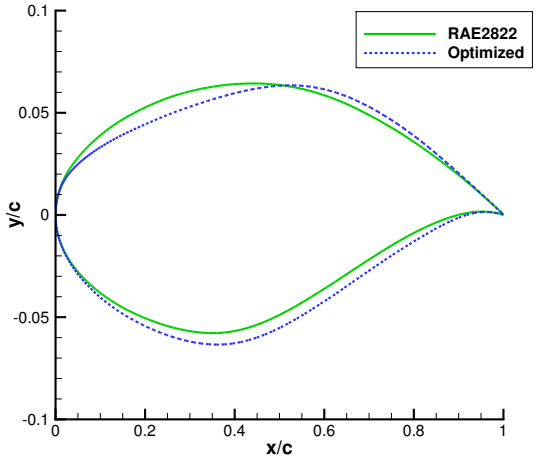
An unstructured mesh is constructed around the baseline airfoil. It consists of 16,937 triangular elements, 8,606 nodes, 200 edges along the airfoil, and 75 edges along the far-field boundary. In both drag and noise minimization cases presented here, the unsteady simulations of the pitching airfoil are performed using a second-order accurate dual time-stepping scheme with 25 time steps per period for a total of 10 periods of forced pitching.

A total of 50 Hicks-Henne bump functions are chosen as design variables and they are equally spaced along the upper and lower surfaces of the airfoil (25 on each of the upper and lower surfaces).

For the drag minimization problem, the objective function is defined by Equation 22. For the noise-minimization problem, the objective function is defined as the time-averaged pressure fluctuation within the periodic steady state at

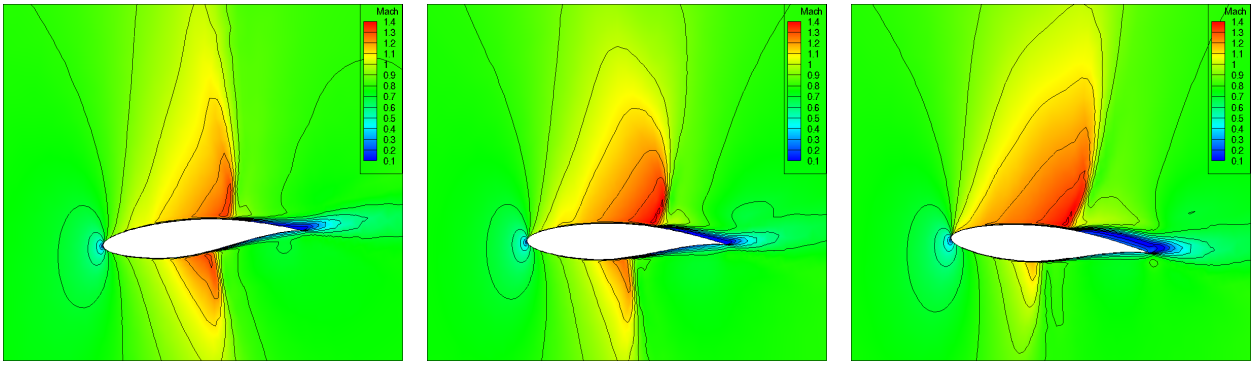


(a) Optimization convergence history of time-averaged drage and lift coefficients.



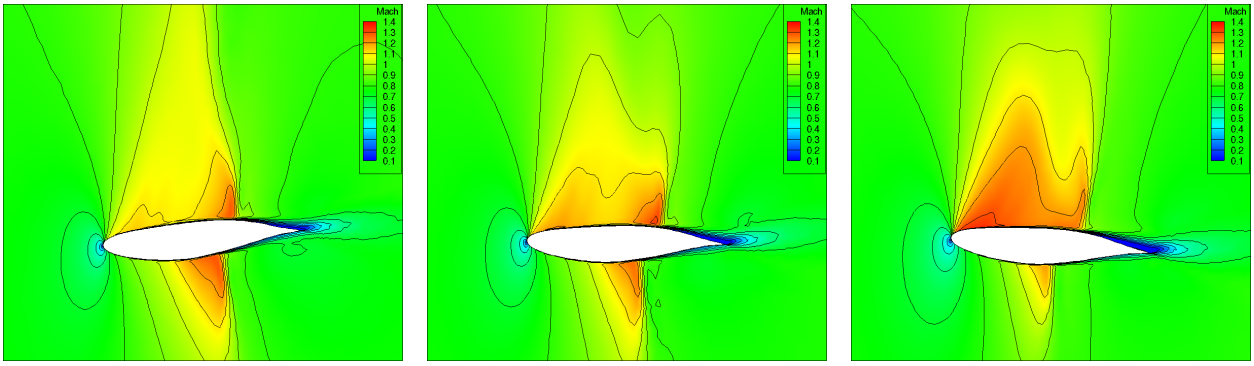
(b) Comparison between the baseline RAE2822 airfoil and the optimized airfoil

Figure 2. Optimization convergence history and the comparison between the baseline RAE2822 airfoil and the optimized airfoil



(a) $\alpha = -1^\circ$ (b) $\alpha = 2^\circ$ (c) $\alpha = 5^\circ$

Figure 3. Mach number contour of the baseline RAE2822 airfoil at various time instances over one period



(a) $\alpha = -1^\circ$ (b) $\alpha = 2^\circ$ (c) $\alpha = 5^\circ$

Figure 4. Mach number contour of the optimized airfoil at various time instances over one period

an observer location:

$$J_N = \frac{1}{N-N^*} \sum_{n=N^*+1}^{N} (p^n_{obs} - p_\infty)^2 \tag{23}$$

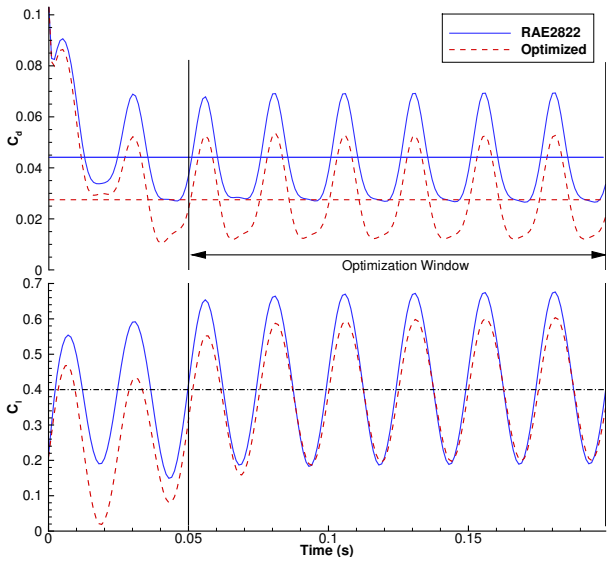


Figure 5. Time history of lift and drag coefficients of the RAE2822 and the optimized airfoil. The horizontal lines denote time-averaged values

where p^n_{obs} is the instantaneous pressure at time level n computed at an observer location and p_∞ is the free-stream pressure. The observation point is placed $10c$ below the trailing edge of the airfoil. N is the total number of time steps and N^* is the number of time steps before entering the optimization window in which the magnitude of the pressure fluctuation is to be minimized. In this case $N^* = 50$ and the optimization window contains 8 periods of pressure fluctuation measured at the observation point.

For both drag and noise minimization cases, the time-average lift $\bar{C}_l$ is constrained in the optimization so that $\bar{C}_l{}^{Optimized} = \bar{C}_l{}^{NACA64A010} = 0$. A volume constraint is also imposed to ensure that the volume of the optimized airfoil remains at least as large as the baseline NACA64A010 airfoil.

A lift-constrained drag minimization is first performed on the baseline airfoil. The time-averaged drag is reduced by 59% after 16 CFD evaluations, while the mean lift is maintained at $\bar{C}_l = 0$, as shown on Figure 6 (a). A comparison of the time histories of C_d between the baseline and the optimzed airfoil is shown on Figure 6 (b). The drag is greatly reduced over the entire optimization window.

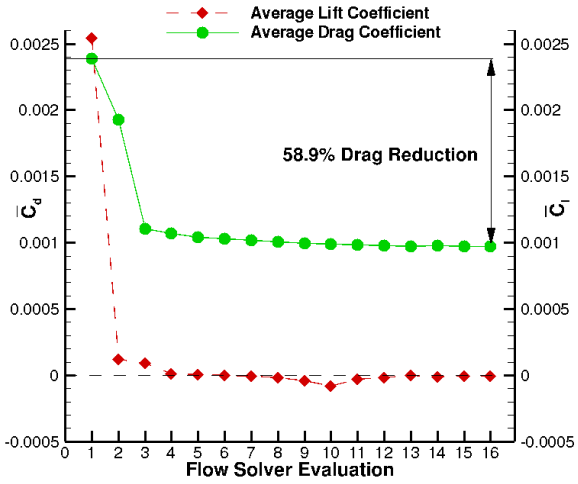
Next, a lift-constrained noise minimization is performed on the baseline airfoil. The time-averaged pressure fluctuation is reduced by 45.6% after 16 CFD evaluations, while the mean lift is maintained at $\bar{C}_l = 0$, as shown on Figure 7 (a). A comparison of the time histories of $p' = p_{obs} - p_\infty$ between the baseline and the optimzed airfoil is shown on Figure 7 (b). It can be seen that the magnitude of pressure fluctuation is significantly reduced over the optimization window for the optimized design.

Figure 8 compares the two optimal airfoils with the baseline. The drag-minimized airfoil is thinner than the baseline airfoil around the mid-chord. This helps removing the shock on both surfaces as the airfoil pitches therefore reducing the wave drag. Its profile also remains symmetrical as the baseline NACA64A010 since the airfoil is pitching about the horizontal axis and the zero mean lift constraint is imposed. The noise-minimized airfoil is much thicker than the baseline airfoil, featurings a significant bulge on the pressure side. This may have been the result of optimizing the pressure fluctuation at a single observation point – the optimizer deforms the shape so as to deflect the pressure waves away from the prescribed observation point at the cost of increased drag.

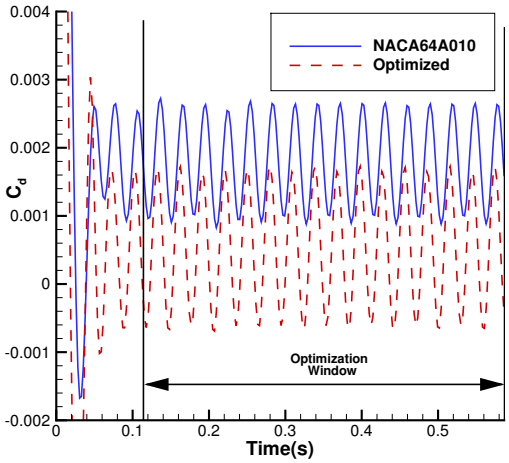
Comparing the performance of the two optimal designs in terms of both drag and noise objectives on Table 8, it can be seen that the time-averaged noise objective J_N is slightly reduced from its baseline value in the drag-minimized design. However, the time-averaged drag is more than 2 orders of magnitudes higher than its baseline value in the noise-minimized design, due to the enlarged airfoil profile and high curvatures which lead to strong shocks at such transonic flow condition. In a similar study conducted by Rumpfkeil and Zingg,[5] the reverse trend is noted – the drag-minimized airfoil results in a noise objective value that is twice as high as the noise-minimized one. While their study was performed for a blunt trailing edge in the subsonic flow regime, it is apparent from their work and this test case that the drag and noise can be two competing objectives.

It is clear from the study of this simple test case that drag and noise minimizations do not lead to the same optimal shape. Moreover, if conducted in parallel, they may not even result in the similar optimal performance in this two-

objective problem in that a noise-minimized design may lead to unsatisfactory rise in drag. This serves to illustrate an important point that aeroacoustic optimization cannot be an 'after-thought' – it must be included in the initial design process in order to minimize its impact on other aerodynamic performances. While design and optimization using steady aerodynamics for lift and drag is still the industry standard today, to reduce aerodynamic noise – an inherently unsteady phenomenon, efficient unsteady simulation and design tools are of paramount importance. To that end, the AD-based discrete adjoint framework developed on the basis of the SU2 multi-physics software suite offers an accurate and robust way to tackle challenges arising from the unsteady aeronautical design problems.

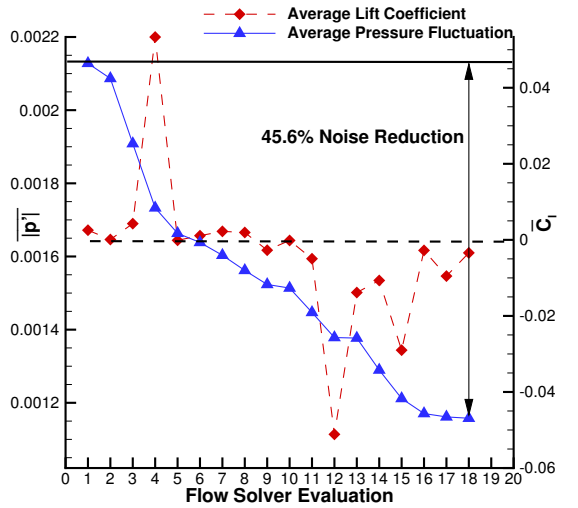


(a) Optimization convergence history of time-averaged drage and lift coefficients.

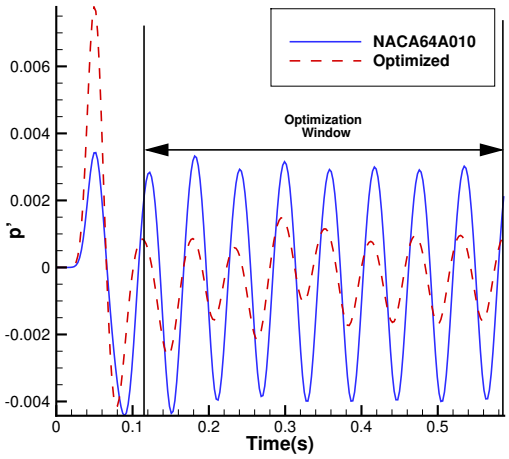


(b) Time history of C_d for the baseline airfoil and the optimized airfoil

Figure 6. Lift-constrained drag minimization result of a Pitching Airfoil in Transonic Inviscid Flow



(a) Optimization convergence history of time-averaged pressure fluctuation and lift coefficient.



(b) Time history of p' for the baseline airfoil and the optimized airfoil

Figure 7. Lift-constrained noise minimization result of a Pitching Airfoil in Transonic Inviscid Flow

IV. Conclusions and Future Work

In this paper, we present an unsteady aerodynamic and aeroacoustic optimization framework in which algorithmic differentiation (AD) is applied to the open-source multi-physics solver SU2 to obtain design sensitivities. An AD-

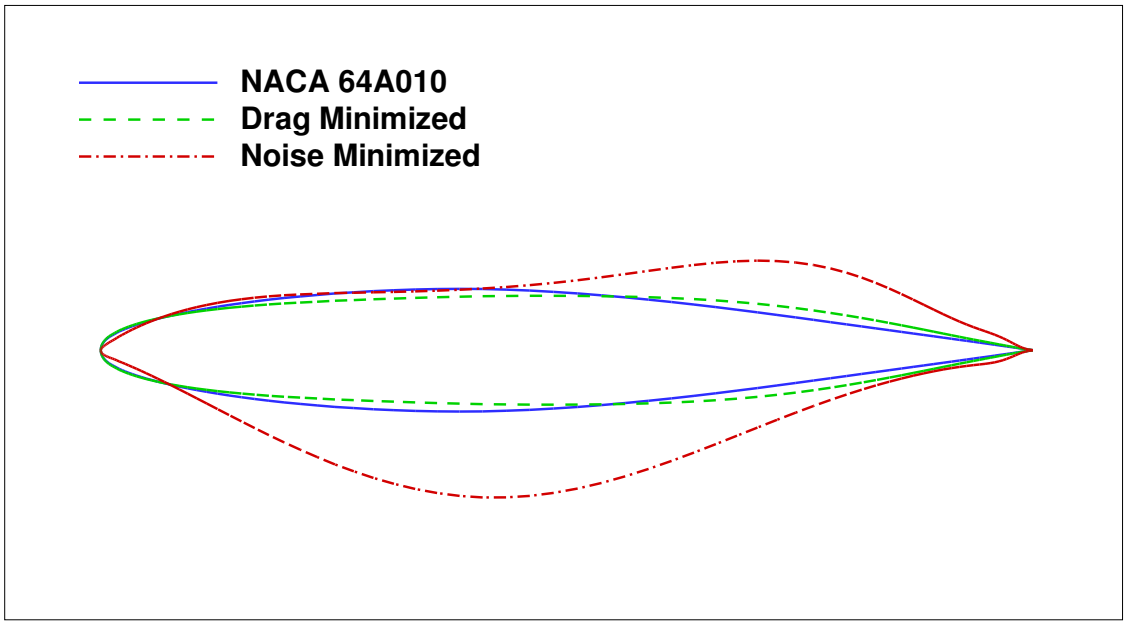


Figure 8. Comparison between NACA64A010, drag-minimized and noise-minimized airfoils

	NACA64A010	Drag Minimized	Noise Minimized
J_D	2.39×10^{-3}	9.75×10^{-4}	1.57×10^{-1}
J_N	2.13×10^{-3}	2.03×10^{-3}	1.16×10^{-3}

Table 2. Comparison between drag minimization and noise minimization results

based consistent discrete adjoint solver is developed which directly inherits the convergence properties of the primal flow solver due to the differentiation of the entire nonlinear fixed-point iterator. This includes the differentiation of the dynamic mesh movement routine and various turbulence model, as well as the hybrid CFD-CAA design chain – all of which cumbersome and error-tasks using the continuous adjoint and hand-discrete approaches.

The resultant AD-based discrete adjoint solver is applied to two test cases – lift-constrained drag minimization of a pitching airfoil in transonic turbulent flow and the lift-constrained drag and noise minimization problem of a pitching airfoil in transonic flow.

The results reveal that the unsteady adjoint information provided by this AD-based discrete adjoint framework is accurate and robust. The drag and noise objective values are significantly reduced in both cases. The second test case shows that drag and noise minimizations do not yield the same optimal shape and the two objectives may even be competing. Depending on the definition of the far-field noise observation point or plane, it is possible for a noise-minimized airfoil to result in unacceptable rise in drag. This highlights the need for efficient simulation and design tools tailored for unsteady problems in order to take both aerodynamic and acoustic performances into account early in the design process.

In the near future, we plan to apply this methology to problems with more complex geometries such as a lift-constrained noise minimization for a multi-element airfoil in landing configurations. Since the URANS solver has been differentiated with AD, for low-speed turbulent problems one may simply switch to URANS for the flow computation of the hybrid CFD-CAA solver in the noise source region. The directivity of noise will also be controlled by measuring the pressure fluctuation at a far-field plane or at multiple radial locations instead of at a single location.

Furthermore, with parallel development efforts from various SU2 contributors in other disciplines, we intend to extend this framework to address other inherently unsteady multi-physics design problems, such as aeroelastic flutter and reacting flows.

V. Acknowledgement

The first author would like to acknowledge the partial funding by the Natural Science and Engineering Research Council of Canada (NSERC). The authors would also like to gratefully acknowledge Dr. Markus Hillenbrand of the RHRK high performance computing center for his support in procuring essential computational resources via the 'Elwetritsch' high performance cluster at the TU Kasierslautern.

References

[1]Pironneau, O. On optimum design in fluid mechanics. *Journal ofFluid Mechanics* **64**, 97–110 (1974).

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

[3]Kenway, G. K. W. and Martins, J. R. R. A. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft* **51** (1), 144–160 (2014).

[4]Lyu, Z. and Martins, J. R. R. A. Aerodynamic design optimization studies of a blended-wing-body aicraft. *Journal of Aircraft* **51** (5), 1604–1617 (2014).

[5]Rumpfkeil, M. P. and Zingg, D. W. A hybrid algorithm for far-field noise minimization. *Computers and Fluids* **39**(9), 1516–1528 (2010).

[6]Mani, K. and Mavriplis, D. J. Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal* **46**(6), 1351–1364 (2008).

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

[8]Economon, T. D., Palacios, F., and Alonso, J. J., A coupled-adjoint method for aerodynamic and aeroacoustic optimization. *AIAA Paper 2012-5598*, (2012).

[9]Economon, T. D., Palacios, F., and Alonso, J. J., An unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Paper 2014-2300*, (2014).

[10]Hogan, R. Fast reverse-mode automatic differentiation using expression templates in C++. *Transactions on Mathematical Software* **40** (26), 1–16 (2014).

[11]Hovland, P., Mohammadi, B., and Bischof, C., Automatic differentiation and Navier-Stokes computations. *Computation Methods for Optimal Design and Control*, 265–284 (1998).

[12]Gauger, N. R., Walther, A., Moldenhauer, C., and Widhalm, M., Automatic differentiation of an entire design chain for aerodynamic shape optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* **96**, 454–461 (2007).

[13]Nemili, A., Özkaya, E., Gauger, N., Kramer, F., Höll, T., and Thiele, F., Optimal design of active flow control for a complex high-lift Cconfiguration. *AIAA Paper 2014-2515*, (2014)

[14]Palacios, F., Economon, T. D., Aranake, A. C., Copeland, S. R., Lonkar, A. K., Lukaczyk, T. W., Manosalvas, D. E., Naik, K. R., Padron, A. S., Tracey, B., Variyar, A., and Alonso, J. J., Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows. *AIAA Paper 2014-0243*, (2014)

[15]Colonius, T., and Lele, S. K., Computational aeroacoustics: progress on nonlinear problems of sound generation. *Progress in Aerospace Sciences* **40**, 345–416 (2004).

[16]Di Francescantonio, P., A New Boundary Integral Formulation for the Prediction of Sound Radiation. *Journal of Sound and Vibration* **202**(4), 491–509 (1997).

[17]Lyrintzis, A. S., Surface integral methods in computational aeroacoustics – From the (CFD) near-field to the (Acoustic) far-field. *International Journal of Aeroacoustics* **2**(2), 95–128 (2003).

[18]Griewank, A., and Walther, A., Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. *Other Titles in Applied Mathematics*, SIAM, 9780898716597., (2008).

[19]Nemili, A., Özkaya, E., Gauger, N. Carnarius, A., Thiele, F., Optimal Control of Unsteady Flows Using Discrete Adjoints. *AIAA Paper 2011-3720*, (2011)