Adjoint-based Unsteady Airfoil Design Optimization with Application to Dynamic Stall

Karthik Mani
Associate Research Scientist
kmani@uwyo.edu

Brian A. Lockwood
Graduate Student
blockwoo@uwyo.edu

Dimitri J. Mavriplis
Professor
mavripl@uwyo.edu

University of Wyoming, Laramie, WY

Abstract

This paper presents the development and application of an adjoint-based optimization method to designing airfoils with the objective of alleviating dynamic stall effects in helicopter rotor blades. The unsteady flow problem is simulated using the NSU2D code, which is a two-dimensional unsteady, viscous, turbulent Reynolds averaged Navier-Stokes (RANS) finite-volume solver. The corresponding adjoint code computes sensitivities of the optimization objective functions with respect to a set of design inputs and is developed through discrete linearization of all components building up the flow solver. Objective functions are defined such that their minimization results in alleviation of undesirable dynamic stall characteristics while the design inputs are parameters that control the shape of the airfoil. The method is applied to an example problem where the goal is to reduce the peak pitching moment during the hysteresis cycle of the SC1095 airfoil while maintaining the baseline time-dependent lift profile. To our knowledge, this work represents the first application of the unsteady adjoint method to optimizing airfoil geometry in dynamic stall problems.

Introduction

Dynamic stall is an unsteady aerodynamic phenomenon on airfoils, wings, and rotor blades, which results in adverse lift, drag, and pitching moment characteristics, compared with static stall, due to unsteady vortices and separation patterns. Dynamic stall constitutes an important problem in the unsteady aerodynamics of rotorcraft. The retreating blade of a rotorcraft in forward flight in a low dynamic pressure, high angle of attack environment is particularly susceptible, and can limit the lifting capability and forward flight speed of the helicopter [1]. Dy-

Presented at the American Helicopter Society 68th Annual Forum, Fort Worth, Texas, May 1-3, 2012.

namic stall also adversely impacts the aeroelastics of the rotor blade (flutter), rotor hub loads, and fatigue life. Fundamental research into dynamic stall has been ongoing for decades [2] in order to understand the underlying unsteady flow physics and airfoil shape dependencies. More recent advances have experimentally investigated passive and active dynamic stall alleviation techniques [3, 4, 5, 6, 7, 8] with generally good success. These modifications include leading edge droop, vortex generators, airfoil recontouring, and active flow control with zero-mass jets. However, often the development of these modifications use trial and error, low-order aerodynamic models, and steady aerodynamic design methods. Thus the actual results are sometimes unpredictable when wind tunnel tested. Overall understanding is still lacking as to the effects that airfoil shape modification have on dynamic stall characteristics and the resultant detailed changes to the flow field.

In order to successfully simulate dynamic stall, viscous and turbulent effects must be taken into account in a time-accurate manner, requiring, at a minimum, the use of an unsteady Reynolds averaged Navier-Stokes simulation capability with a suitable turbulence model. Computational fluid dynamics (CFD) analyses have achieved mixed success in predicting dynamic stall, mainly due to deficiencies in turbulence and transition modeling [4, 6, 9, 10, 11, 12, 13]. Other deficiencies in experimental data, such as extreme unsteadiness, insufficient boundary condition information, and unknown test data error sources, also contribute to poor comparisons. Difficult cases, such as smooth surface separation and trailing edge stall progression (NACA0015), have been predicted successfully using advanced DES-like turbulence modeling [10, 11]. However, in order to remain practical, design tools must remain confined at most to simpler unsteady RANS methods. The risk in relying on simple RANS CFD-based design methods is that the calculations, particularly on the initial geometry, will be grossly in error, and that final numerically generated designs may exploit incorrect physics. However, it has been shown that some types of dynamic stall progressions, for example leading edge initiation (e.g. OA209 and SC1095 airfoil) and milder stall, are often predictable using current RANS technology [4, 13]. These types of airfoils now dominate rotor blade development.

Aerodynamic shape optimization can be used to achieve specific performance objectives through the application of suitable geometric shape modifications. Shape optimization has been used extensively for steady-state problems, such as aircraft cruise drag reduction. Two-dimensional airfoil optimization techniques date back decades and have seen widespread use. The development of adjoint techniques has proven to be an enabling technology for steady-state aerodynamic optimization, especially in 3D, since the full sensitivity vector of a single objective with respect to any number of design variables can be computed with a single adjoint solution, at a cost roughly equivalent to a single flow solution [14]. However, because of the flow physics complexity and computational cost, optimization of unsteady and moving body problems has received significantly less attention [15, 16, 17, 18, 19, 20]. However, recent efforts have lead to the development of both two-dimensional [17, 19, 21] and three-dimensional [18, 20] unsteady adjoint capabilities.

Given these developments, it seems that the technology is in place for the development of a useful dynamic stall alleviation design methodology. The success of current two and three-dimensional practical RANS simulations for realistic configurations, coupled with new developments in unsteady adjoint optimization techniques, can be leveraged to produce effective two and three-dimensional helicopter rotor design tools targeting dynamic stall alleviation. Even though current capabilities for the prediction of dynamic stall effects are still under development, advancement in state-of-the-art optimization cannot possibly wait until dynamic stall analysis is a completely solved problem. Rather, advances in dynamic stall analysis and optimization capabilities must be sought concurrently.

This paper presents the development of the adjoint linearization of the unsteady RANS equations in two dimensions for computing the gradient vector for use in an optimization environment. All components building up the flow solver including the turbulence model are linearized and the gradient vector computed is the discretely exact sensitivity of the objective function with respect to the set of design inputs that control the shape of the airfoil. The optimization example problem presented, where the peak pitching moment is reduced without compromising lift, shows great promise for the method within the limits of the solver's ability to capture the correct physics in the flow.

Analysis Problem Formulation

Governing equations and discretization

The conservative form of the Navier-Stokes equations is used in solving the flow problem. In vectorial form the conservative form of the Navier-Stokes equations may be written as:

$$\frac{\partial \mathbf{U}(\mathbf{x},t)}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{U}) + \mathbf{G}(\mathbf{U})) = 0 \tag{1}$$

where $\mathbf{U}$ represents the vector of conserved variables (mass, momentum, and energy), $\mathbf{F}(\mathbf{U})$ represents the convective fluxes and $\mathbf{G}(\mathbf{U})$ represents the viscous fluxes. Applying the divergence theorem and integrating over a moving control volume $A(t)$ bounded by the control surface $B(t)$ yields:

$$\int_{A(t)} \frac{\partial \mathbf{U}}{\partial t} dA + \int_{B(t)} \mathbf{F}(\mathbf{U}) \cdot \mathbf{n} dB + \int_{B(t)} \mathbf{G}(\mathbf{U}) \cdot \mathbf{n} dB = 0 \tag{2}$$

Using the differential identity:

$$\frac{\partial}{\partial t} \int_{A(t)} \mathbf{U} dA = \int_{A(t)} \frac{\partial \mathbf{U}}{\partial t} dA + \int_{B(t)} \mathbf{U}(\dot{\mathbf{x}} \cdot \mathbf{n}) dB \tag{3}$$

equation (2) is rewritten as:

$$\begin{aligned} \frac{\partial}{\partial t} \int_{A(t)} \mathbf{U} dA + \int_{B(t)} [\mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}] \cdot \mathbf{n} dB \\ + \int_{B(t)} \mathbf{G}(\mathbf{U}) \cdot \mathbf{n} dB = 0 \end{aligned} \tag{4}$$

Considering cell-averaged values for the state $\mathbf{U}$ and a moving but non-deforming control volume A, the spatially discrete version is then:

$$A \frac{\partial \mathbf{U}}{\partial t} + \mathbf{S}(\mathbf{U}, \mathbf{x}(\mathbf{t}), \dot{\mathbf{x}}(\mathbf{t}), \mathbf{n}(\mathbf{t})) = 0 \tag{5}$$

where $\mathbf{S}$ represents the spatially integrated discrete convective and viscous fluxes. A is the control volume, $\mathbf{x}(\mathbf{t})$ the time varying mesh coordinates, $\dot{\mathbf{x}}(\mathbf{t})$ the time varying grid speeds, and $\mathbf{n}(\mathbf{t})$ the time varying normals of the control volume boundary faces which are computed from $\mathbf{x}(\mathbf{t})$.

The time derivative term is discretized using either a first-order accurate backward-difference-formula (BDF1), or a second-order accurate BDF2 scheme. These discretizations are shown in equations (6) and (7) respectively. The index n is used to indicate the current time-level as the convention throughout the paper.

$$A \frac{\partial \mathbf{U}}{\partial t} = \frac{A\left(\mathbf{U}^n - \mathbf{U}^{n-1}\right)}{\Delta t} \tag{6}$$

$$A \frac{\partial \mathbf{U}}{\partial t} = \frac{A\left(\frac{3}{2}\mathbf{U}^n - 2\mathbf{U}^{n-1} + \frac{1}{2}\mathbf{U}^{n-2}\right)}{\Delta t} \tag{7}$$

The solver uses a spatially second-order accurate node-centered finite-volume full Navier-Stokes discretization with matrix dissipation for the convective fluxes. The one equation Spalart-Allmaras [24] turbulence model is used to simulate the effect of turbulence in the flow. Further details of the discretization used in the flow solver can be found in references [25, 26].

Solution procedure

For an implicit time-step the nonlinear flow residual is defined as:

$$\mathbf{R}^n = A \frac{\partial \mathbf{U}}{\partial t} + \mathbf{S}(\mathbf{x}^n, \dot{\mathbf{x}}^n, \mathbf{U}^n) = 0 \tag{8}$$

where the discretization of the time derivative is based on the chosen time-integration scheme. The second term is the spatial residual of the flow and is a function of the grid velocities $\dot{\mathbf{x}}$, the current mesh coordinates $\mathbf{x}$ and the flow solution at the current time-level $\mathbf{U}^n$. The mesh orientation at the current time-level is obtained through rigid prescribed rotations as a function of time and the corresponding grid speeds are computed using the analytic derivative of the prescribed motion equations. The non-linear residual is linearized and solved using Newton's method at each time-level as:

$$\begin{gathered} \left[\frac{\partial \mathbf{R}(\mathbf{U}^k)}{\partial \mathbf{U}^k}\right] \delta \mathbf{U}^k = -\mathbf{R}(\mathbf{U}^k) \\ \mathbf{U}^{k+1} = \mathbf{U}^k + \delta \mathbf{U}^k \end{gathered} \tag{9}$$

where the linear system is solved using either agglomeration multigrid [27] or preconditioned GMRES [28] with multigrid as the preconditioner. Jacobi iterations are used as the smoother in the multigrid algorithm.

Geometry parametrization

Modification of the baseline geometry is achieved through displacements of the surface nodes defining the geometry. In order to ensure smooth geometry shapes, any displacement of a surface node is controlled by a bump function which influences neighboring nodes with an effect that diminishes moving away from the displaced node. The bump function used for the work presented in this paper is the Hicks-Henne Sine bump function [29]. The design variables or inputs for the optimization examples presented form a vector of weights controlling the magnitude of bump functions placed at various chordwise locations. Modifications to the surface of the airfoil are propagated to the interior mesh using the tension spring analogy algorithm for mesh deformation. Details of the implementation of the mesh deformation algorithm can be found in references [21, 30].

Sensitivity Formulation for the Unsteady Flow Problem

The time-integrated objective function L^g that is to be minimized can be written with intermediate dependencies as a function of the design variables $\mathbf{D}$ in order to assist with the chain rule differentiation process. For the problem of the unsteady pitching airfoil, the design variables form a vector of weights controlling the magnitudes of bumps placed at various chordwise locations. With intermediate dependencies the objective function can be written as:

$$L^g = \sum_{n=0}^{n_{steps}} L^n \left(\mathbf{U}^n(\mathbf{D}), \mathbf{x}^n(\mathbf{x}^0(\mathbf{D})) \right) \tag{10}$$

where $L^n, L^{n-1} \cdots$ denote local objective function values at each time-step. Differentiating equation (10) results in a general expression for the final sensitivity derivative $\frac{dL^g}{d\mathbf{D}}$ for the unsteady flow problem as:

$$\frac{dL^g}{d\mathbf{D}} = \sum_{n=0}^{n_{steps}} \frac{\partial L^g}{\partial L^n} \left[\frac{\partial L^n}{\partial \mathbf{U}^n} \frac{\partial \mathbf{U}^n}{\partial \mathbf{D}} + \frac{\partial L^n}{\partial \mathbf{x}^n} \frac{\partial \mathbf{x}^n}{\partial \mathbf{D}} \right] \tag{11}$$

where intermediate derivatives are not shown for simplicity. For typical airfoil shape design problems there are multiple design variables and only a single objective function. This requires the transpose of equation (11) in order to obtain the adjoint or reverse linearization of the sensitivity as shown below.

$$\frac{dL^{g\,T}}{d\mathbf{D}} = \sum_{n=0}^{n_{steps}} \left[\frac{\partial \mathbf{U}^{n\,T}}{\partial \mathbf{D}} \frac{\partial L^{n\,T}}{\partial \mathbf{U}^n} + \frac{\partial \mathbf{x}^{n\,T}}{\partial \mathbf{D}} \frac{\partial L^{n\,T}}{\partial \mathbf{x}^n} \right] \frac{\partial L^{g\,T}}{\partial L^n} \tag{12}$$

Expanding the summation backward:

$$\frac{dL^{g\,T}}{d\mathbf{D}} = \underbrace{\left(\frac{\partial \mathbf{U}^{n\,T}}{\partial \mathbf{D}} \frac{\partial L^{n\,T}}{\partial \mathbf{U}^n} + \frac{\partial \mathbf{x}^{n\,T}}{\partial \mathbf{D}} \frac{\partial L^{n\,T}}{\partial \mathbf{x}^n} \right) \frac{\partial L^{g\,T}}{\partial L^n}}_{(X)} + \underbrace{\left(\frac{\partial \mathbf{U}^{n-1\,T}}{\partial \mathbf{D}} \frac{\partial L^{n-1\,T}}{\partial \mathbf{U}^{n-1}} + \frac{\partial \mathbf{x}^{n-1\,T}}{\partial \mathbf{D}} \frac{\partial L^{n-1\,T}}{\partial \mathbf{x}^{n-1}} \right) \frac{\partial L^{g\ \ T}}{\partial L^{n-1}}}_{(Y)} \cdots\cdots \tag{13}$$

Here $\partial L^g / \partial L^n$ is a scalar quantity, while $\partial L^n / \partial \mathbf{U}^n$ and $\partial L^n / \partial \mathbf{x}^n$ are vectors that are easily computable since L^n is a scalar quantity. These are merely linearizations with respect to the flow variables and the mesh coordinates. However, the sensitivity of the flow state to the design variables $\frac{\partial \mathbf{U}}{\partial \mathbf{D}}$ is a matrix and cannot be directly obtained. In order to obtain a suitable expression for this term we revisit equation (8) which forms the constraint to be satisfied at each time-step in the sensitivity problem. For a BDF1 time discretization the flow constraint equation (i.e. the non-linear flow residual at a given time-level n) can be written as:

$$\mathbf{R}^n \left(\mathbf{x}^n(\mathbf{x}^0(\mathbf{D})), \mathbf{U}^n(\mathbf{D}), \mathbf{U}^{n-1}(\mathbf{D}) \right) = 0 \tag{14}$$

Differentiating this with respect to the design variables $\mathbf{D}$ yields:

$$\frac{d\mathbf{R}^n}{d\mathbf{D}} = \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n} \frac{\partial \mathbf{x}^n}{\partial \mathbf{D}} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \frac{\partial \mathbf{U}^n}{\partial \mathbf{D}} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^{n-1}} \frac{\partial \mathbf{U}^{n-1}}{\partial \mathbf{D}} = 0 \tag{15}$$

which, when transposed and rearranged, results in an expression for the flow variable sensitivity as:

$$\frac{\partial \mathbf{U}^{n\,T}}{\partial \mathbf{D}} = -\left(\frac{\partial \mathbf{x}^{n\,T}}{\partial \mathbf{D}} \frac{\partial \mathbf{R}^{n\,T}}{\partial \mathbf{x}^n} + \frac{\partial \mathbf{U}^{n-1\,T}}{\partial \mathbf{D}} \frac{\partial \mathbf{R}^{n\ \ T}}{\partial \mathbf{U}^{n-1}} \right) \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^{-T} \tag{16}$$

Considering only term (X) in equation (13) and substituting the above expression we obtain:

$$
\begin{aligned}
(X) = -\left(\frac{\partial \mathbf{x}^n}{\partial \mathbf{D}}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T + \frac{\partial \mathbf{U}^{n-1}}{\partial \mathbf{D}}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^{n-1}}^T \right) \\
\times \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^{-T} \frac{\partial L^g}{\partial \mathbf{U}^n}^T + \frac{\partial \mathbf{x}^n}{\partial \mathbf{D}}^T \frac{\partial L^g}{\partial \mathbf{x}^n}^T
\end{aligned}
\tag{17}
$$

Introducing an adjoint variable $\Lambda_{\mathbf{U}}^n$ at time-level n defined as:

$$
\Lambda_{\mathbf{U}}^n = -\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^{-T} \left[\frac{\partial L^g}{\partial \mathbf{U}^n} \right]^T \tag{18}
$$

we obtain a linear flow adjoint system to be solved iteratively as shown below.

$$
\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^T \Lambda_{\mathbf{U}}^n = -\left[\frac{\partial L^g}{\partial \mathbf{U}^n} \right]^T \tag{19}
$$

This is nearly identical to the linear systems that arise in the Newton solver for the nonlinear analysis problem with the exception of the transposed flow Jacobian matrix. Convergence of both systems are similar since the eigenvalues of the linear operator in both cases are identical. The same methods employed to solve the linear systems in the nonlinear problem may be used to solve for the adjoint variable vector.

Once the adjoint variable vector has been obtained iteratively, term (X) now becomes:

$$
\begin{aligned}
(X) = \underbrace{\frac{\partial \mathbf{x}^n}{\partial \mathbf{D}}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T \Lambda_{\mathbf{U}}^n}_{(A)^n} + \underbrace{\frac{\partial \mathbf{U}^{n-1}}{\partial \mathbf{D}}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^{n-1}}^T \Lambda_{\mathbf{U}}^n}_{(B)^n} \\
+ \underbrace{\frac{\partial \mathbf{x}^n}{\partial \mathbf{D}}^T \frac{\partial L^g}{\partial \mathbf{x}^n}^T}_{(C)^n}
\end{aligned}
\tag{20}
$$

The sum of terms $(A)^n$ and $(C)^n$ forms the contribution to the total gradient vector $[dL/d\mathbf{D}]^T$ from time-level n as:

$$
(A)^n + (C)^n = \frac{\partial \mathbf{x}^0}{\partial \mathbf{D}}^T \frac{\partial \mathbf{x}^n}{\partial \mathbf{x}^0}^T \left\{ \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T \Lambda_{\mathbf{U}}^n + \frac{\partial L^g}{\partial \mathbf{x}^n}^T \right\} \tag{21}
$$

where the second term is simply the transpose of the grid rotation matrix:

$$
[\beta]^{nT} = \frac{\partial \mathbf{x}^n}{\partial \mathbf{x}^0}^T \tag{22}
$$

The backward recurrence relation is now clear since term $(B)^n$ can be combined with the appropriate flow sensitivity term at time-level $n-1$ that appears in (Y) and the process can and be repeated. Assembling the complete sensitivity vector $[dL/d\mathbf{D}]^T$ involves starting at the final time-level and sweeping backward in time while solving for a vector of flow adjoint variables at each time-level. The general form of the flow adjoint equation to be solved at an arbitrary time-level n for BDF1 and BDF2 based time discretizations are given as:

$$
BDF1: \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^T \Lambda_{\mathbf{U}}^n = -\frac{\partial L^g}{\partial \mathbf{U}^n}^T + \frac{\partial \mathbf{R}^{n+1}}{\partial \mathbf{U}^n}^T \Lambda_{\mathbf{U}}^{n+1} \tag{23}
$$

$$
\begin{aligned}
BDF2: \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}^n} \right]^T \Lambda_{\mathbf{U}}^n = -\frac{\partial L^g}{\partial \mathbf{U}^n}^T + \frac{\partial \mathbf{R}^{n+1}}{\partial \mathbf{U}^n}^T \Lambda_{\mathbf{U}}^{n+1} \\
+ \frac{\partial \mathbf{R}^{n+2}}{\partial \mathbf{U}^n}^T \Lambda_{\mathbf{U}}^{n+2}
\end{aligned}
\tag{24}
$$

where $\Lambda_{\mathbf{U}}^n$ are the variables to be solved for and all terms on the right-hand-side are known from previous (future) steps during the backward sweep in time.

Solver Validation

Dynamic stall prediction

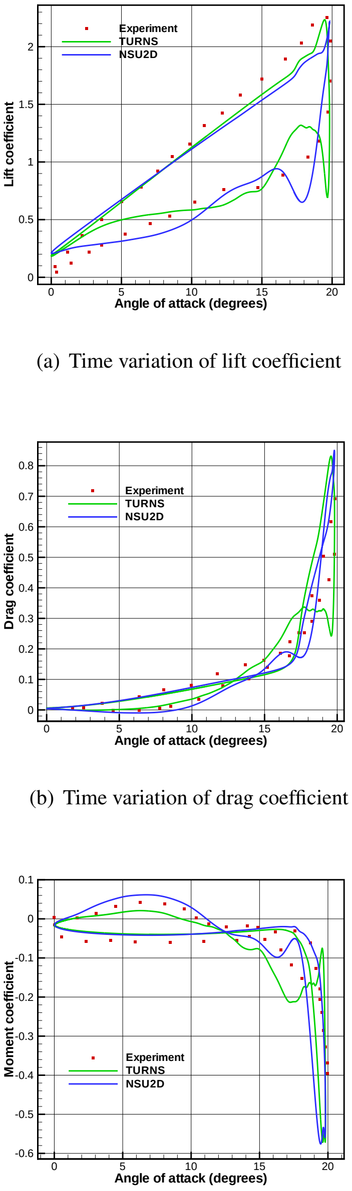
The NSU2D code was used to simulate a dynamic stall event for the SC1095 airfoil at a Mach number of 0.302 and a Reynolds number of 3.86 million. The mean angle-of-attack for this case was 9.92^o and a sinusoidal pitching motion of amplitude 9.9^o with a reduced frequency of 0.148 was prescribed. This case is documented experimentally as Frame 37109 in the NASA Technical Memo 84245 [2]. The computations using NSU2D were performed on a grid of approximately 80,000 points, using a total of 10,000 time-steps of size $\Delta t = 0.02$, with 25 multigrid (5 grid levels) per time-step. The simulation covered just over 3 periods of motion, with the results displaying periodic behavior after the first period. The entire simulation required a total of 7 hours wall-clock time using OpenMP parallelization on 8 cores. The flow was assumed to be fully turbulent for this case and no transition was specified. Figure 1 shows the time-dependent loads computed by NSU2D and compares them to experimental data. Reasonable agreement is seen in the histories of all three load distributions, lift, drag and moment. Figure 2 shows the typical convergence of the flow equations and the adjoint equations at an arbitrary time-step with 25 multigrid cycles. A parameter study was performed where the number of time-steps and number of multigrid cycles at each time-step were varied to determine optimal values and the results are shown in Figure 3.

Gradient verification

Verification of the gradient vector obtained from the adjoint solver was done using a three-step process. The forward or tangent linearization gradients were first verified against gradient components obtained from a complex version of the analysis solver, after which the principle of duality [17] between transpose operations was used to verify the adjoint gradients against the forward linearization values. The complex version of the analysis solver was developed be redefining all floating point numbers as complex and overloading certain operators such MIN,MAX,ABS etc. to handle complex arguments. The complex solver takes advantage of the fact that functions evaluated using complex arguments can be used to determine the derivative of the function with respect to the arguments by introducing a complex perturbation in the argument. The derivative $f'(x)$ of any real function $f(x)$ can be computed as:

$$f'(x) = Im[f(x+ih)]/h \qquad (25)$$

where h is the complex perturbation introduced into the argument x and $f(x+ih)$ is the complex equivalent of the real function $f(x)$. The derivative is then simply the imaginary part of the complex function divided by the perturbation. Complex perturbations are introduced into the design variables one at a time and the solver is run in complex mode to determine the sensitivity of the objective function to that design variable. While the complex step method much like the finite-difference method requires N complex solutions for N design variables, it is not subject to subtraction errors that affect traditional finite differencing. This allows the perturbation to be significantly small with values generally being in the 10^{-100} range for the current work. In fact it is a requirement that the complex perturbation be small enough such that operations do not result in the contamination of the real part, i.e. less than



(a) Time variation of lift coefficient

(b) Time variation of drag coefficient

(c) Time variation of moment coefficient

Figure 1: Predicted time-dependent loads for the SC1095 airfoil as computed by NSU2D compared with TURNS [31] and experimental data.

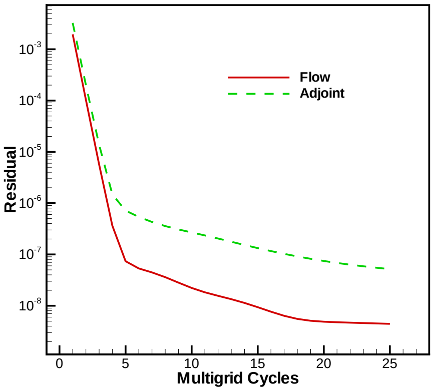


Figure 2: Typical convergence of the flow and adjoint equations at an arbitrary time-step for the SC1095 test case with 25 multigrid cycles.

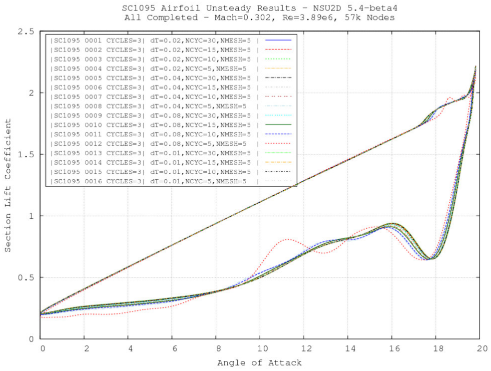


Figure 3: Time-step and multigrid cycles sensitivity study for the SC1095 test case.

Functional	C_l	C_m
Adjoint	-0.0257896419950**4**	-2.42401026265272
Forward	-0.0257896419950**4**	-2.42401026265272
Complex	-0.0257896419950**5**	-2.42401026265272

Table 1: Comparison of adjoint sensitivity against forward sensitivity and those obtained by the complex-step method for a single design variable, i.e. single bump function located on the upper surface at midchord. (Non-matching digits in bold font)

machine precision for the real part. Table 1 compares the gradients obtained from different methods and indicates near perfect agreement.

Optimization Example

The goal of the optimization example presented here was to reduce the peak pitching moment during the dynamic stall cycle without compromising the time variation of the lift coefficient. The dynamic stall case is identical to the one presented in the validation section, i.e. SC1095 airfoil undergoing periodic pitching at a reduced frequency of $k_c = 0.148$ in freestream conditions of Mach number 0.302 and Reynolds number 3.86×10^6. The mean angle-of-attack is $\alpha_0 = 9.92^o$ and the pitch amplitude is $\alpha_m = 9.9^o$. The time variation of the angle-of-attack of the airfoil is given by:

$$\alpha = \alpha_0 + \alpha_m \sin(\omega t - \pi/2) \qquad (26)$$

where a phase shift of 90^o is included to assure that the airfoil begins $(t = 0)$ the pitch cycle with the lowest angle-of-attack possible. This is necessary to preserve code stability during startup since free-stream flow parameters are used as the initial condition at all nodes in the mesh. The time-integration used 750 time-steps of size $\Delta t = 0.08$ which translates into a approximately 1 period of pitch. 25 multigrid cycles were used at each time-step for both the flow and adjoint solutions.

The objective function for this case was defined as:

$$L^g = \left[\sum_{n=1}^{n_{steps}} (C_m^n)^{100} \right]^{\frac{1}{100}} + W \sum_{n=1}^{n_{steps}} (C_l^n - C_{l_{target}}^n)^2 \qquad (27)$$

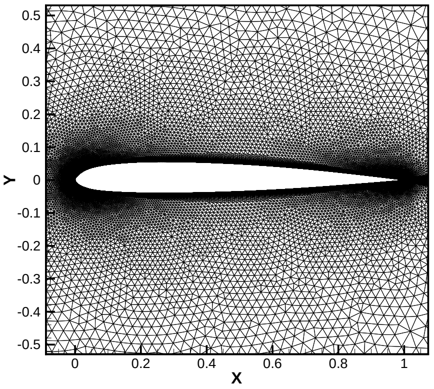


Figure 4: Computational mesh of the SC1095 airfoil consisting of 56268 nodes used in the optimization example.

where the first term corresponds to a high order norm that targets the peak moment during the pitch cycle and the second term acts as a constraint on the time variation of the lift. W is a weight on the constraint which may be modified during the course of the optimization if necessary. The target lift coefficients at each time-step were those of the baseline SC1095 airfoil.

The geometry parameterization consisted of 10 design variables acting as weights on bump functions, half of which were placed on the upper surface and the other half placed on the lower surface of airfoil. The bump functions were equally spaced in the chordwise direction between a non-dimensional chord of $c = 0$ and $c = 0.8$. This was done to prevent the optimization from making drastic modifications to the trailing edge turning angle. The optimization algorithm used in this case was the L-BFGS reduced hessian gradient-based method [23]. The gradient vector is computed by running the flow solver followed by the adjoint code and then passed into the L-BFGS routine to obtain a modified design. The case was run on an single cluster node with 16 cores and utilized OpenMP parallelization. A single function/gradient call consisting of a forward analysis run and a backward adjoint solve took approximately 20 minutes of wall clock time. The computational mesh consisted of 56298 nodes and is shown in Figure 4.

The optimization was started with the baseline SC1095 airfoil using a loose lift constraint weight of $W = 0.001$ and run until a limit of 50 function/gradient requests by the optimizer were reached. L-BFGS being a local gradient-based optimization algorithm has a tendency to not move from the initial airfoil configuration when a strong weight for the constraint is used initially since that is where the constraint is satisfied exactly. Loosening the weight during startup allows the optimizer to focus on minimizing the peak moment during the pitch cycle and move out of the starting area in the design space as opposed to focusing on satisfying the constraint. Once a significant reduction in the peak moment was achieved, the lift constraint weight was tightened to $W = 0.01$. The optimization was then run again until 50 requests for function/gradient values were made by the optimizer. At this point the constraint weight was further tightened to its final value of $W = 0.1$. This final stage of the optimization primarily focused on satisfying the constraint and terminated when 122 requests for function/gradient values were made by the optimizer. The L-BFGS optimization algorithm uses multiple function/gradient values within a single design iteration as it involves line searches during the process. As a consequence the number of design iterations is typically less than the number of calls to the function/gradient code. However, it is important to track and provide termination criteria based on the number of function/gradient requests rather than the number of design iterations since the cost of the optimization is directly associated with the evaluation of the function and gradient. 17 design iterations were performed in the third and final stage of the optimization by the L-BFGS algorithm using information from 122 function and gradient values. Overall approximately 200 function/gradient runs were made during the design process resulting in a wall clock time of about 70 hours.

The convergence of the last stage of the optimization is shown in Figures 5 and 6. The reduction in the objective function at each design iteration and the number of function/gradient requests made at each iteration is shown in Figure 5. Figure 6 shows the convergence of the objective function and the constraint separated out of the objective during the course of the optimization. Figure 7 shows the final optimized design for the airfoil and compares it to the baseline SC1095 airfoil. Figure 8 shows the time-variation of the loads for the final optimized design

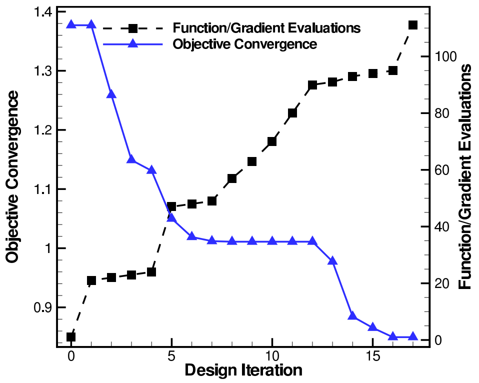


Figure 5: Convergence of objective for the optimization example with L-BFGS design iterations and number of function/gradient evaluations.

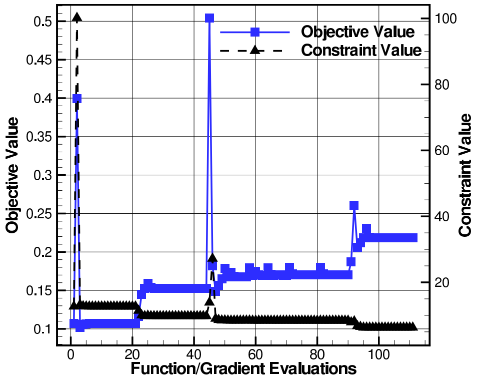
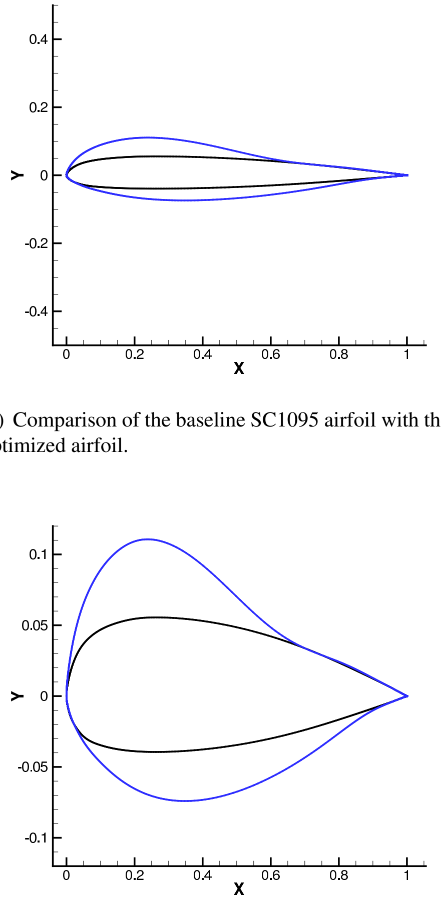


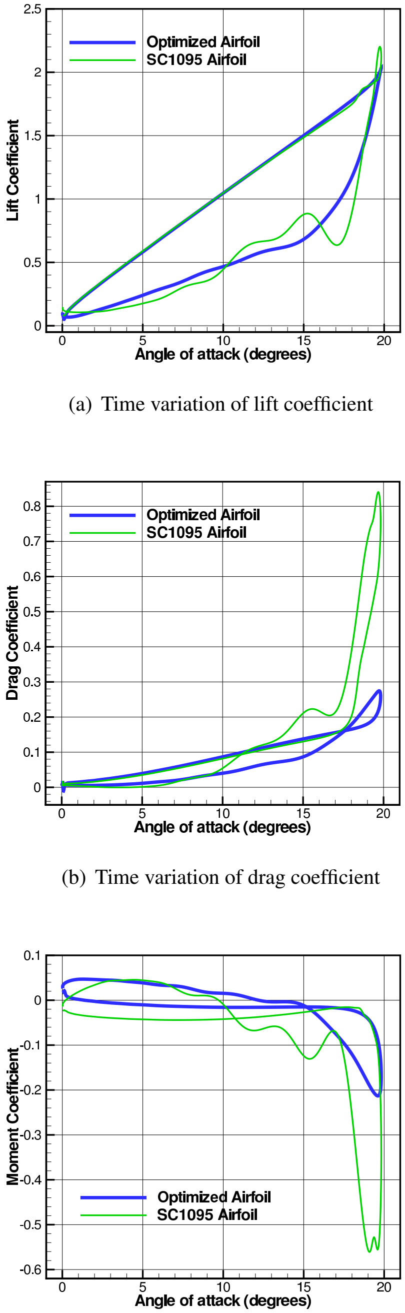
Figure 6: Convergence of objective function and constraint separated from objective function for the optimization example against number of function/gradient evaluations.



(a) Comparison of the baseline SC1095 airfoil with the final optimized airfoil.

(b) Comparison using an exaggerated scale.

Figure 7: Comparison of optimized and baseline airfoils.



(a) Time variation of lift coefficient

(b) Time variation of drag coefficient

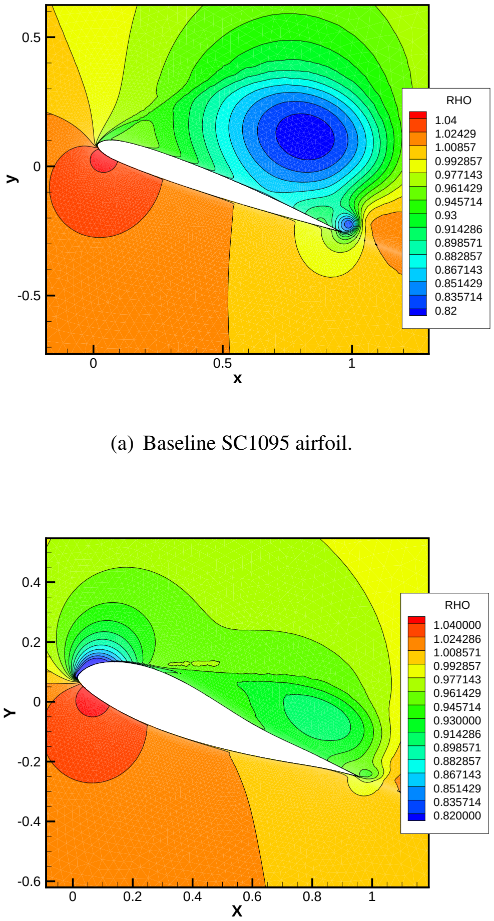
(c) Time variation of moment coefficient

Figure 8: Comparison of initial and final time-dependent loads for the optimization example.

of the airfoil. A large reduction of nearly 60% is seen in the peak value for the moment coefficient during the cycle, while the time variation of the lift is not significantly changed particularly during the upstroke. Although the drag coefficient was not included in the objective formulation, a significant reduction in the peak drag value is also observed in the final design. It was noticed from the flowfield animations of the final design that leading edge initiation of separation had been eliminated and separation with a weaker vortex closer to midchord had been introduced. This explains a mitigated peak moment as a weaker vortex rolls past the trailing edge and a shallower stall in lift distribution as less circulation is lost in the process. Figure 9 compares the density contours between the baseline SC1095 airfoil and the final optimized design of the airfoil at the highest angle-of-attack during the pitch cycle. The plot indicates a significantly weaker vortex being shed from the optimized airfoil.

Conclusions

This work has demonstrated the possibility of performing airfoil shape optimization for dynamic stall alleviation. The use of a time dependent adjoint approach enables the calculation of the sensitivities with respect to any number of design variables with a single adjoint backwards time-integration sweep which is equivalent to the cost of a single forward time-dependent flow simulation. Thus, for two-dimensional simulations, optimization cases can be run on a multicore workstation within several days. Although the example shown herein was successful in reducing pitching moment magnitudes for the prescribed test case, the performance of the resulting thicker airfoil shape will likely degrade at other operating conditions. Thus, future work will focus on extending the process to include multi-point and multi-objective optimization. Additionally, the bump functions used to define the airfoil shapes are ill-suited for producing leading-edge droop configurations, which are well known to be beneficial for stall alleviation. Thus, the incorporation of more sophisticated and flexible shape design parameters will also be investigated.



(a) Baseline SC1095 airfoil.

(b) Final optimized design.

Figure 9: Comparison of the density contours between the baseline SC1095 airfoil and the final optimized design airfoil close to the highest angle of attack in the pitch cycle of the optimization example presented.

Acknowledgements

This work was supported by ARMY SBIR Phase 1 contract W911W6-11-C-0037.

References

[1] P. B. Martin, K. W. McAlister, M. S. Chandrasekhara, and W. Geissler. Dynamic stall measurements and computations for a VR-12 airfoil with a variable droop leading edge. 59th Annual Forum of the American Helicopter Society, Phoenix, AZ, 2003.

[2] K. W. McAlister, S. L. Pucci, W. J. McCroskey, and L. W. Carr. An experimental study of dynamic stall on advanced airfoil sections. Volume 2: Pressure and force data. USAAVRADCOM TR-82-A-8 (or NASA TM 84245), September 1982.

[3] P. Martin, J. Wilson, J. Berry, T-C Wong, M. Moulton, and M. McVeigh. Passive control of compressible dynamic stall. AIAA Paper 2008-7506, 26th AIAA Applied Aerodynamics Conference, Honolulu, HI, August 2008.

[4] W. Geissler, G. Dietz, and H. Mai. Dynamic stall and its passive control investigations on the OA209 airfoil section. 31st European Rotorcraft Forum, Florence, Italy, September 2005.

[5] M. Kerho. Adaptive airfoil dynamic stall control. Journal of Aircraft, 44(4), 2007.

[6] H. Mai, G. Dietz, W. Geissler, K. Richter, J. Bosbach, H. Richard, and K. de Groot. Dynamic stall control by leading edge vortex generators. Journal of the American Helicopter Society, 53(1):2636, 2006.

[7] M. Chandrasekhara, P. Martin, and C. Tung. Compressible dynamic stall performance of a variable droop leading edge airfoil with a gurney flap. AIAA Paper 2004-0041, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2004.

[8] H. Nagib, J. Kiedasch, D. Greenblatt, I. Wygnanski, and A. Hassan. Effective flow control for rotorcraft applications at flight Mach numbers. AIAA Paper 2001-2974, 31st AIAA Fluid Dynamics Conference and Exhibit, Anaheim, CA, June 2001.

[9] S. Choi, J. Alonso, E. van der Weide, and J. Sitaraman. Validation study of aerodynamic analysis tools for design optimization of helicopter rotors. AIAA Paper 2007-3929, 18th AIAA Computational Fluid Dynamics Conference, Miami FL., June 2007.

[10] E. Duque, L. Sankar, S. Menon, O. Bauchau, S. Ruffin, M. Smith, K. Ahuja, K. Brentner, L. Long, P. Morris, and F. Gandhi. Revolutionary physics-based design tools for quiet helicopters. AIAA Paper 2006-1068, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2006.

[11] A. Shelton, K. Braman, M. Smith, and S. Menon. Improved turbulence modeling for rotorcraft. American Helicopter Society 62nd Annual Forum, Phoenix, Arizona, May 2006.

[12] J. Szydlowski and M. Costes. Simulation of flow around a NACA0015 airfoil for static and dynamic stall configurations using RANS and DES. AHS International 4th Decennial Specialists Conference on Aeromechanics, San Francisco, CA, January 2004.

[13] K. Duraisamy, W. McCroskey, and J. Baeder. Wind tunnel wall interference effects on steady subsonic airfoil flows. AIAA Paper 2006-2993, 24th Applied Aerodynamics Conference, San Francisco, CA, June 2006.

[14] A. Jameson. Re-engineering the design process through computation. Journal of Aircraft, 36:3650, 1999.

[15] K. Ghayour and O. Baysal. Limit-cycle shape optimization using time-dependent transonic equation. AIAA Paper 99-3375, Proceedings of the 14th Computational Fluid Dynamics Conference, Norfolk VA, June 1999.

[16] S. K. Nadarajah and A. Jameson. Optimal control of unsteady flows using a time-accurate method. AIAA Paper 2002-5436, September 2002.

[17] K. Mani and D. J. Mavriplis. An unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. AIAA Journal, 46(6):13511364, 2008.

[18] D. J. Mavriplis. Solution of the unsteady discrete adjoint for three- dimensional problems on dynamically deforming unstructured meshes. AIAA Paper 2008-0727, January 2008.

[19] M. P. Rumpfkeil and D. W. Zingg. The optimal control of unsteady flows with a discrete adjoint method. Optimization and Engineering, 2008. doi:10.1007/s11081-008-9035-5.

[20] E. Nielsen and B. Diskin. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. AIAA Journal, 48(6):11951206, 2010.

[21] K. Mani and D. J. Mavriplis. Adjoint based sensitivity formulation for fully coupled unsteady aeroelasticity problems. AIAA Journal, 47(8):19021915, 2009.

[22] Hicks, R. and Henne, P., Wing Design by Numerical Optimization, Journal of Aircraft, Volume 15-7, 1978, pp 407-412.

[23] C. Zhu and R. H. Byrd and J. Nocedal, L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization, ACM Transactions on Mathematical Software, Volume 23-4,1997,pp 550-560.

[24] Spalart, P. R. and Allmaras, S. R., ”A One-Equation Turbulence Model for Aerodynamic Flows” AIAA Paper 92-0439

[25] Valarezo, W. O. and Mavriplis, D. J., ”Navier-Stokes applications to high-lift airfoil analysis,” AIAA J. of Aircraft , Vol. 32, No. 3, 1995, pp. 618-624.

[26] Mavriplis, D.J., ”Viscous Flow Analysis Using a Parallel Unstructured Multigrid Solver” AIAA Journal, 2000, 0001-1452 vol.38 no.11 (2067-2076) doi: 10.2514/2.894

[27] Mavriplis, D., ”Multigrid Techniques for Unstructured Meshes,” Notes prepared for 26th Computational Fluid Dynamics Lecture Series Program of the von Kar-man Institute of Fluid Dynamics, Rhode St Genese, Belgium, 1995.

[28] Y. Saad and M.H. Schultz, ”GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”, SIAM J. Sci. Stat. Comput., 7:856-869, 1986. doi:10.1137/0907058.

[29] Hicks, R. and Henne, P., ”Wing Design by Numerical Optimization,” Journal of Aircraft , Vol. 15-7, 1978, pp. 407-412.

[30] Batina, J. T., ”Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes,” AIAA Journal, Vol. 28-8, August 1990, pp. 1381-1388.

[31] Sitaraman, J. and Baeder, J.D.,”Evaluation of the Wake Prediction Methodologies used in CFD based Rotor Airload Computations”, AIAA-2006-3472, AIAA 24th Conference on Applied Aerodynamics, Washington, DC, January 2006.