AIAA JOURNAL
Vol. 51, No. 6, June 2013

Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids

Eric J. Nielsen*
NASA Langley Research Center, Hampton, Virginia 23681
and
Boris Diskin†
National Institute of Aerospace, Hampton, Virginia 23681

DOI: 10.2514/1.J051859

A discrete adjoint-based design methodology for unsteady turbulent flows on three-dimensional dynamic overset unstructured grids is formulated, implemented, and verified. The methodology supports both compressible and incompressible flows and is amenable to massively parallel computing environments. The approach provides a general framework for performing highly efficient and discretely consistent sensitivity analysis for problems involving arbitrary combinations of overset unstructured grids that may be static, undergoing rigid or deforming motions, or any combination thereof. General parent–child motions are also accommodated, and the accuracy of the implementation is established using an independent verification based on a complex-variable approach. The methodology is used to demonstrate aerodynamic optimizations of a wind-turbine geometry, a biologically inspired flapping wing, and a complex helicopter configuration subject to trimming constraints. The objective function for each problem is successfully reduced, and all specified constraints are satisfied.

Nomenclature

$\mathbf{A}$	=	interpolation matrix
A, B	=	amplitudes of rotation in degrees
a, b, c, d	=	temporal coefficients
C	=	aerodynamic coefficient
$\mathbf{C}$	=	$m_q \times 1$ vector of zeros and ones, indicator of time derivatives
c	=	wing chord
C_L	=	lift coefficient
C_{M_x}, C_{M_y}	=	lateral and longitudinal moment coefficients
C_Q	=	torque coefficient
$\mathbf{C}_s$	=	$m_s \times 1$ vector of zeros and ones, indicator of time derivatives at solve points
C_T	=	thrust coefficient
$\mathbf{D}$	=	vector of design variables
Diag	=	diagonal matrix operator
E	=	total energy per unit volume
$\mathbf{F}$	=	$m_q \times 3$ flux matrix
f, s	=	general functions
$\mathbf{F}_{\text{inv}}$, $\mathbf{F}_{\text{visc}}$	=	inviscid and viscous fluxes
f_{obj}	=	objective function
$\mathbf{G}$	=	grid operator
g_1, g_2	=	explicit constraint functions
$\mathbf{I}$	=	projector operator
i	=	$\sqrt{-1}$
J	=	number of cost function components
$\mathbf{K}$	=	$m_x \times m_x$ linear elasticity coefficient matrix
L	=	Lagrangian function
m_d	=	size of vector $\mathbf{D}$
m_f	=	size of solution vector at fringe points
m_h	=	size of solution vector at hole points
m_q	=	size of solution vector $\mathbf{Q}$
m_s	=	size of solution vector at solve points
m_x	=	size of vector $\mathbf{X}$
N	=	number of time levels
n	=	time level
$\hat{\mathbf{n}}$	=	3×1 outward-pointing normal vector
$\mathbf{P}$	=	$m_h \times m_q$ pseudo-Laplacian matrix
p	=	pressure; also cost function exponent
$\mathbf{Q}$	=	$m_q \times 1$ vector of volume-averaged conserved variables
$\mathbf{q}$	=	$m_q \times 1$ vector of conserved variables
R	=	3×3 rotation matrix
$\mathbf{R}$	=	$m_s \times 1$ vector of spatial undivided residuals
$\mathcal{R}$	=	$m_x \times m_x$ block-diagonal rotation matrix
R_{GCL}	=	residual of static geometric conservation law
$\mathbf{R}_{\text{GCL}}$	=	$m_s \times 1$ vector of R_{GCL}
S	=	control volume surface area
T	=	4×4 transform matrix
t	=	time
u, v, w	=	Cartesian components of velocity
V	=	control volume
$\mathbf{V}$	=	$m_q \times m_q$ diagonal matrix of cell volumes
$\mathbf{W}$	=	3×1 face velocity vector
$\mathbf{X}$	=	$m_x \times 1$ vector of grid coordinates
$\mathbf{x}$	=	3×1 position vector
x	=	independent variable
x, y, z	=	Cartesian coordinate directions
α	=	interpolation coefficient
β	=	scaling parameter for incompressible continuity equation
ε	=	perturbation
θ	=	angle of rotation; also blade pitch
θ_c	=	collective input
θ_{1c}	=	lateral cyclic input
θ_{1s}	=	longitudinal cyclic input
$\mathbf{\Lambda}$	=	$m_q \times 1$ flowfield adjoint variable
$\mathbf{\Lambda}_g$	=	$m_x \times 1$ grid adjoint variable
ρ	=	density
$\boldsymbol{\tau}^n$	=	$m_x \times 1$ translation vector
$\boldsymbol{\tau}$	=	3×1 translation vector
ψ	=	blade azimuth
ω	=	cost function component weight

Presented as Paper 2012-0554 at the 2012 Aerospace Sciences Meeting, Nashville, TN, 9–12 January 2012; received 26 January 2012; revision received 11 June 2012; accepted for publication 13 June 2012; published online 5 April 2013.

*Research Engineer, Computational AeroSciences Branch, MS 128. Senior Member AIAA.

†Fellow, 100 Exploration Way; currently Visiting Associate Professor, Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904. Associate Fellow AIAA.

ω_1, ω_2	=	frequencies of rotation, rad/s
$\circ$	=	Hadamard vector multiplication operator
$\odot$	=	extension of $\circ$ to a vector-matrix product

Subscripts/Superscripts

c	=	child motion
f	=	fringe point
h	=	hole point
i, j, k, m, n	=	indices
in	=	quantity at initial conditions
nb	=	quantity at simply connected neighbor
p	=	parent motion
s	=	solve point
x, y, z	=	axis of rotation
∞	=	quantity at freestream conditions
$*$	=	target quantity
$-$	=	volume-averaged or time-average quantity; also complement of a vector

I. Introduction

As ACCESS to powerful high-performance computing resources has become more widespread in recent years, the use of high-fidelity physics-based simulation tools for analysis of complex aerodynamic flows has become increasingly routine. The availability and affordability of high-fidelity analysis tools has in turn stimulated an enormous body of research aimed at applying such tools to formal design optimization of complex aerospace configurations. A survey of the relevant literature shows that optimization methods based on the Euler and Reynolds-averaged Navier–Stokes equations have indeed gained a strong foothold in the design cycle for problems governed by steady flows [1,2]. Conversely, formal optimization methods for problems involving unsteady flow are also under development [3–9], but in general are not as mature at the present time. This lag can be attributed at least in part to the increased computational cost typically associated with unsteady simulations.

For gradient-based optimization of problems involving many design variables, the ability to generate sensitivity information at a relatively low cost is critical. Unlike forward differentiation techniques such as finite differencing [10], direct differentiation [11], and complex-variable methods [12], the adjoint approach performs sensitivity analysis at a cost comparable to that of a flow solution and independent of the number of design variables [13]. Efficient evaluation of sensitivities of an output with respect to all input parameters has led to numerous applications of adjoint-based methods in various areas of research and engineering. Some recent adjoint-based developments include a mathematically rigorous approach to error estimation and mesh adaptation [14], simultaneous design of shape and active flow control parameters for a high-lift configuration [3], efficient methods for uncertainty quantification [15], sonic boom optimization [16], laminar flow control [17], and many others.

Adjoint methods can be further classified into continuous and discrete variants, depending on the order in which the differentiation and discretization of the governing equations is performed. A discrete adjoint approach to sensitivity analysis is taken here. The methodology has been widely used for a broad class of optimization problems involving both steady and unsteady flows [3,5,18–24]. One of the advantages of the discrete adjoint approach is that the sensitivities computed by this method can be verified to machine precision by comparison with complex-variable sensitivities [12]. The approach requires a complete linearization of the discrete governing equations with respect to both the flowfield variables and mesh coordinates. Strictly speaking, the adjoint approach for unsteady flows requires the evaluation of these linearizations at each physical time step. Therefore, the predominant challenge in extending a steady-state implementation to the unsteady regime is the development of an efficient infrastructure to store and access the entire forward solution as needed.

The analysis of vehicles experiencing large relative motion of vehicle components is often accomplished using overset discretizations. Design optimization for unsteady flows using such discretizations serves as the primary motivation for the current work. An implementation of the discrete adjoint approach for optimization of general three-dimensional unsteady turbulent flows on single-block unstructured grids is described in [3,5]. Others have previously demonstrated adjoint-based capabilities for overset mesh discretizations; however, such works have been restricted to steady flows [25–29]. The methodology described here is intended for aerodynamic optimization of configurations characterized by large dynamic grid motions.

The primary contributions of this paper are the development, implementation, verification, and demonstration of an adjoint-based methodology for optimization and design of the most general unsteady aerodynamic flows. In the case of rotary-wing flows, an optimization reported here involves a full helicopter configuration subject to trimming constraints and completes the series of studies addressing models of progressively higher fidelity. The previously considered models include actuator disk approaches [30], noninertial formulations [20], and dynamic grid formulations involving isolated rotors [5]. The generality of dynamic overset unstructured grid methods makes this methodology applicable to the most general flows occurring in a variety of practical computational fluid dynamics applications, e.g., store/stage separation, turbomachinery, wind-turbine systems, rotary-wing systems, biologically inspired devices, and many others. Several diverse large-scale design applications are demonstrated in this paper.

The material is presented in the following order. The governing equations and some fundamental concepts of overset mesh systems are presented first. The approach taken to solve the flow equations is reviewed, followed by a derivation of the accompanying discrete adjoint equations. Details of the solution strategy are covered, and the accuracy of the implementation for a very general dynamic motion case is verified using an independent approach based on complex variables. Finally, successful demonstrations of the design methodology are shown for a wind-turbine geometry, a biologically inspired flapping wing, and a realistic helicopter configuration. The Appendix contains derivations for high-order temporal schemes.

II. Governing Equations

In this paper, the unsteady turbulent flow equations are used in both compressible and incompressible formulations. The primary distinction between these formulations is that the incompressible continuity equation does not have a time derivative term; all other (compressible and incompressible) equations do have time derivatives. For a simultaneous description of the unsteady compressible and incompressible Navier–Stokes equations, it is convenient to introduce an indicator of time derivative $\mathbf{C}$ and a Hadamard vector multiplication operator [31]. The vector $\mathbf{C}$ is a logical vector composed of zeros and ones and has the same dimension as the residual vector. Ones correspond to equations with time derivatives, while zeros correspond to equations with no time derivatives. The logical complement to $\mathbf{C}$, $\bar{\mathbf{C}}$, is a vector of the same dimension in which zeros are replaced by ones and vice versa. The Hadamard operator is denoted as $\circ$ and acts on two vectors of the same dimension, which are multiplied in an element-by-element fashion. The result of the Hadamard multiplication is a vector of the same dimension. The simultaneous description of the flow equations involves the Hadamard multiplication of the vector $\mathbf{C}$ with the vector of time derivatives. The resulting equations can be written in the following form for both moving and stationary control volumes:

$$\mathbf{C} \circ \frac{\partial}{\partial t} \int_V \mathbf{q}\, \mathrm{d}V + \oint_{\partial V} (\mathbf{F}_{\mathrm{inv}} - \mathbf{F}_{\mathrm{visc}}) \hat{\mathbf{n}}\, \mathrm{d}S = 0 \tag{1}$$

where V is the control volume bounded by the surface ∂V, and $\hat{\mathbf{n}}$ is an outward-pointing unit normal. The vector $\mathbf{q}$ represents the conserved variables for mass, momentum, and energy. The matrices $\mathbf{F}_{\mathrm{inv}}$ and $\mathbf{F}_{\mathrm{visc}}$ denote the inviscid and viscous fluxes, respectively; the flux

matrices are composed of three columns, each representing the flux components along coordinate directions. The product of a flux matrix and $\hat{\mathbf{n}}$ is a vector of the same dimension as $\mathbf{q}$.

For a moving control volume, the viscous flux is unchanged, while the inviscid flux accounts for the difference in the fluxes due to the movement of control volume faces. Given inviscid fluxes $\mathbf{F}$ on a static grid, the corresponding fluxes $\mathbf{F}_{\text{inv}}$ on a moving grid can be defined as $\mathbf{F}_{\text{inv}} = \mathbf{F} - (\mathbf{C} \circ \mathbf{q} + \bar{\mathbf{C}})\mathbf{W}^T$, where $\mathbf{W} = [W_x, W_y, W_z]^T$ is a local face velocity. In other words, for an equation with a time derivative corresponding to the conservation of a scalar quantity q, $\mathbf{f}_{\text{inv}} = \mathbf{f} - q\mathbf{W}^T$, where $\mathbf{f}$ and $\mathbf{f}_{\text{inv}}$ are corresponding rows of the matrices $\mathbf{F}$ and $\mathbf{F}_{\text{inv}}$, respectively. For an equation without a time derivative, $\mathbf{f}_{\text{inv}} = \mathbf{f} - \mathbf{W}^T$.

By defining a volume-averaged quantity $\bar{\mathbf{q}}$ within each control volume,

$$\bar{\mathbf{q}} = \frac{1}{V}\int_V \mathbf{q}\,\mathrm{d}V \tag{2}$$

the conservation equations given by Eq. (1) take the form

$$\mathbf{C} \circ \frac{\partial(\bar{\mathbf{q}}V)}{\partial t} + \oint_{\partial V} (\mathbf{F}_{\text{inv}} - \mathbf{F}_{\text{visc}})\hat{\mathbf{n}}\,\mathrm{d}S = 0 \tag{3}$$

Here, the conserved variables and the inviscid flux matrix for compressible flows are defined as $\bar{\mathbf{q}} = [\rho, \rho u, \rho v, \rho w, E]^T$ and

$$\mathbf{F}_{\text{inv}} = \begin{bmatrix} \rho(u-W_x) & \rho(v-W_y) & \rho(w-W_z) \\ \rho u(u-W_x)+p & \rho u(v-W_y) & \rho u(w-W_z) \\ \rho v(u-W_x) & \rho v(v-W_y)+p & \rho v(w-W_z) \\ \rho w(u-W_x) & \rho w(v-W_y) & \rho w(w-W_z)+p \\ E(u-W_x)+pu & E(v-W_y)+pv & E(w-W_z)+pw \end{bmatrix} \tag{4}$$

and the perfect gas equation of state is assumed. For incompressible flows, $\bar{\mathbf{q}} = [p, u, v, w]^T$ and

$$\mathbf{F}_{\text{inv}} = \begin{bmatrix} \beta(u-W_x) & \beta(v-W_y) & \beta(w-W_z) \\ u(u-W_x)+p & u(v-W_y) & u(w-W_z) \\ v(u-W_x) & v(v-W_y)+p & v(w-W_z) \\ w(u-W_x) & w(v-W_y) & w(w-W_z)+p \end{bmatrix} \tag{5}$$

where β is a scaling parameter analogous to the artificial compressibility parameter [32]. Recall, however, that the incompressible continuity equation does not have a time derivative. The viscous flux $\mathbf{F}_{\text{visc}}$ is not explicitly shown here. For turbulent flows, the equations are closed with an appropriate turbulence model for the eddy viscosity.

The high-order (up to third-order) backward difference (BDF) discretizations for the time derivative of a function s are defined as

$$\frac{\partial s}{\partial t} = \frac{1}{\Delta t}(as^n + bs^{n-1} + cs^{n-2} + ds^{n-3}) \tag{6}$$

where n is a time level, and the coefficients are given in Table 1. The number after the BDF abbreviation indicates the order of the scheme. The coefficients listed for the BDF2opt scheme are a linear combination of the BDF2 and BDF3 coefficients taken from [33,34]. The resulting scheme is second-order accurate but has a leading truncation error term less than that of the BDF2 scheme.

Table 1 Coefficients for BDF schemes

Scheme	a	b	c	d
BDF1	1	−1	0	0
BDF2	3/2	−2	1/2	0
BDF3	11/6	−3	3/2	−1/3
BDF2opt	1.66	−2.48	0.98	−0.16

Using a BDF1 scheme, the discrete form of the flow equations at time level n is given as

$$\mathbf{C} \circ \frac{\bar{\mathbf{q}}^n V^n - \bar{\mathbf{q}}^{n-1} V^{n-1}}{\Delta t} + \mathbf{R}^n = 0 \tag{7}$$

where V^n and $\bar{\mathbf{q}}^n$ are a control volume and the corresponding solution vector at time level n, and $\mathbf{R}^n$ is a vector of spatial undivided residuals approximating the flux term in Eq. (3). The first-order temporal scheme is chosen for the sake of simplicity; higher-order BDF schemes are used in practical computations and the following demonstrations. The Arbitrary Lagrangian–Eulerian [35] node-centered finite volume discretization of Eq. (3) used in the current work and described in [36] employs the following discrete formulation:

$$\mathbf{C} \circ \frac{\bar{\mathbf{q}}^n - \bar{\mathbf{q}}^{n-1}}{\Delta t} V^n + \mathbf{R}^n + R^n_{\text{GCL}}(\mathbf{C} \circ \bar{\mathbf{q}}^{n-1} + \beta\bar{\mathbf{C}}) = 0 \tag{8}$$

Here,

$$R^n_{\text{GCL}} = \oint_{\partial V} [\mathbf{W}^n]^T \hat{\mathbf{n}}\,\mathrm{d}S \tag{9}$$

where $\mathbf{W}^n$ is a vector of local face velocities at time level n. Note that substituting a spatially and temporally constant state vector, $\bar{\mathbf{q}}$, into Eq. (7) results in a geometric conservation law (GCL) [37]

$$\frac{V^n - V^{n-1}}{\Delta t} - R^n_{\text{GCL}} = 0 \tag{10}$$

for an equation with a time derivative and

$$-\beta R^n_{\text{GCL}} = 0 \tag{11}$$

for the incompressible continuity equation. Equation (8) is obtained by subtracting the GCL residual, multiplied by $\bar{\mathbf{q}}^{n-1}$ for equations with time derivatives, from Eq. (7).

III. Overset Grids

An overset grid formulation is characterized by the presence of two or more overlapping component grids. Each grid point and its corresponding control volume may be classified as one of four types based on the nature of the equation to be solved for that control volume. "Solve" points are points at which the discretized flow equations given by Eq. (8) are defined. "Fringe" points are points in overlap regions where interpolated data are specified in lieu of boundary conditions. The equations defined at fringe points are interpolation equations, such that the solution at a fringe point, $\bar{\mathbf{q}}_f$, is defined as a linear combination of solution values at solve points, $\bar{\mathbf{q}}_s$:

$$\bar{\mathbf{q}}_f - \sum_k \alpha^k \bar{\mathbf{q}}_s^k = 0, \qquad \sum_k \alpha^k = 1 \tag{12}$$

Typically, the fringe point and the solve points appearing in Eq. (12) belong to different overlapping component grids. "Hole" points refer to points outside the computational domain, e.g., within the boundaries of a wing. In the current approach, the solution at hole points, $\bar{\mathbf{q}}_h$, is set to the average of the solution values at its simply connected neighbors, $\bar{\mathbf{q}}^j_{\text{nb}}$. This averaging procedure is equivalent to a discrete pseudo Laplacian, which is an elliptic operator:

$$\sum_j (\bar{\mathbf{q}}_h - \bar{\mathbf{q}}^j_{\text{nb}}) = 0 \tag{13}$$

where the hole-point neighbors are identified by j. Finally, "orphan" points refer to grid points located within the computational domain for which neither the flow equations are imposed nor can suitable points be found from which to interpolate solution information. In the current effort, the same pseudo-Laplacian procedure is defined for

hole and orphan points, so that orphan points are treated as hole points and are not considered as a separate entity in the formulation to follow.

For dynamic grid motions, the character of each grid point may change as a function of time. It is preferable to have grid topologies such that the residuals of the governing equations at solve and fringe points do not depend on solution values at hole points. At a minimum, hole-point solutions should not contribute to residuals at solve and fringe points within the same time level. In practice, it can be difficult to prevent solutions at hole points from contributing to residuals at solve points through the time derivative; however, maximizing the extent of fringe regions and reducing the time step can help to alleviate this difficulty.

The domain-connectivity information required by the overset implementation is established using the software libraries described in [38]. This methodology has been used extensively with the flow solver for performing analysis of multibody problems undergoing large relative motions [30,36,39–45]. Given the topology of each component grid, each grid point in the composite grid is determined to be a solve, fringe, hole, or orphan point. This procedure is performed dynamically during the solution process as required by the grid motion. The mesh elements containing fringe points are established, and the weighting coefficients required to interpolate data at such points are evaluated. For cases in which the grid motion is periodic, the user may choose to store the domain-connectivity information during the first cycle of motion for use in subsequent cycles. Once the interpolation coefficients are known, the complementary library described in [46] is used to determine the current solution at fringe points. The solution at hole and orphan points is determined based on user-supplied subroutines specifying the desired treatment at such locations. In the current approach, the pseudo Laplacian given by Eq. (13) is used.

IV. Flow Solver

References [23,36,47,48] describe the flow solver[‡] used in the current work. The code can be used to perform aerodynamic simulations across the speed range, and an extensive list of options and solution algorithms is available for spatial and temporal discretizations on general static or dynamic mixed-element unstructured meshes, which may or may not contain overset grid topologies.

In the current study, the spatially second-order-accurate discretization uses a finite volume approach in which the dependent variables are stored at the vertices of tetrahedral meshes. Inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe [49], and viscous fluxes are formed using an approach equivalent to a finite element Galerkin procedure. The incompressible implementation is based on [48,50]. For dynamic mesh cases, the mesh velocity terms are evaluated using backward differences consistent with the discrete time derivative; this makes the spatial and GCL residuals dependent on grids at previous time levels. The eddy viscosity is modeled using the one-equation approach of Spalart and Allmaras [51]. The turbulence model is integrated all the way to the wall without the use of wall functions and is weakly coupled, i.e., solved separately from the mean flow equations at each time step. Scalability to thousands of processors is achieved through parallel domain decomposition, preprocessing, and solver mechanics [3,52]. Postprocessing operations such as the generation of isosurface and computational schlieren animations are also performed in parallel, avoiding the need for a single image of the mesh or solution at any time and ultimately yielding a highly efficient end-to-end parallel simulation paradigm. To date, this approach has been used to carry out computations on meshes containing as many as 2 billion points and 12 billion tetrahedral elements.[§]

To collectively describe equations and solutions defined at solve, fringe, and hole points, it is convenient to introduce corresponding projectors $\mathbf{I}_s^n$, $\mathbf{I}_f^n$, and $\mathbf{I}_h^n$ at time level n. These operators are rectangular matrices of respective dimensions $m_s \times m_q$, $m_f \times m_q$, and $m_h \times m_q$, and for which the rows contain a single unity entry complemented by zeros. The values m_s, m_f, and m_h are the solution dimensions at all solve, fringe, and hole points, respectively, and $m_q = m_s + m_f + m_h$ is the solution dimension at all grid points. The projectors are used to extract solutions at grid points of a specific type: $\mathbf{Q}_s^n = \mathbf{I}_s^n\mathbf{Q}^n$, $\mathbf{Q}_f^n = \mathbf{I}_f^n\mathbf{Q}^n$, and $\mathbf{Q}_h^n = \mathbf{I}_h^n\mathbf{Q}^n$, where $\mathbf{Q}^n$ is the vector of solution values at all grid points and $\mathbf{Q}_s^n$, $\mathbf{Q}_f^n$, and $\mathbf{Q}_h^n$ are the vectors of solution values at solve, fringe, and hole points, respectively. Finally, note that the projector operators can vary in time.

The discrete form of the flow equations with a BDF1 scheme for the time derivative at time level n can be written as

$$\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + [(\mathbf{I}_s^n\mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n] \circ \mathbf{R}_{\mathrm{GCL}}^n = 0 \tag{14}$$

In Eq. (14) and all discussions to follow, $\mathbf{R}^n$ and $\mathbf{R}_{\mathrm{GCL}}^n$ are $m_s \times 1$ vectors that include residuals at solve points, $\mathbf{V}^n$ is an $m_q \times 1$ vector of control volumes for all equations at time level n, $\mathbf{V}_s^n = \mathbf{I}_s^n\mathbf{V}^n$ is a subset of $\mathbf{V}^n$ corresponding to solve points, $\mathbf{C}_s^n$ is an $m_s \times 1$ vector indicator of a time derivative restricted to solve points at time level n, and $\bar{\mathbf{C}}_s^n$ is the complement of $\mathbf{C}_s^n$. Note that a solve point at time level n may or may not be a solve point at time level $n-1$.

The equations at fringe points are defined as

$$\mathbf{A}^n\mathbf{Q}^n = 0 \tag{15}$$

where $\mathbf{A}^n$ is the $m_f \times m_q$ matrix defining the interpolation of solution data from overset grid solutions at time level n as introduced in Eq. (12). The equations at hole points are defined as

$$\mathbf{P}^n\mathbf{Q}^n = 0 \tag{16}$$

where $\mathbf{P}^n$ is the $m_h \times m_q$ matrix of the pseudo Laplacian given by Eq. (13).

The Jacobian of the implicit equations (14–16) at time level n is a 3×3 block matrix of the form

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n) + \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n} & \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n} & \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n} \\ \mathbf{A}_s^n & \mathbf{A}_f^n & \mathbf{A}_h^n \\ \mathbf{P}_s^n & \mathbf{P}_f^n & \mathbf{P}_h^n \end{bmatrix} \tag{17}$$

where $\mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n)$ is a diagonal $m_s \times m_s$ matrix with the vector $\mathbf{C}_s^n \circ \mathbf{V}_s^n$ on the main diagonal; $\mathbf{A}_f^n$ is an $m_f \times m_f$ diagonal matrix describing interpolation at fringe points; $\mathbf{A}_s^n$ and $\mathbf{A}_h^n$ are matrices with respective dimensions $m_f \times m_s$ and $m_f \times m_h$ describing interpolation from solve and hole points; and $\mathbf{P}_s^n$, $\mathbf{P}_f^n$, and $\mathbf{P}_h^n$ are matrices with respective dimensions $m_h \times m_s$, $m_h \times m_f$, and $m_h \times m_h$ describing contributions of solve, fringe, and hole points to the pseudo Laplacian defined at hole points. Note that, if the solution at hole points does not contribute to residuals at solve and fringe points within the same time level, then $\partial\mathbf{R}^n/\partial\mathbf{Q}_h^n = 0$, $\mathbf{A}_h^n = 0$, and the equations at hole points decouple from the equations at solve and fringe points.

V. Grid Equations

The general grid equations can be defined in the form

$$\mathbf{G}^n(\mathbf{X}, \mathbf{D}) = 0 \tag{18}$$

where the $m_x \times 1$ vector $\mathbf{X}$ represents the coordinates of the composite overset mesh (meshes at several time levels may be involved), $\mathbf{D}$ is the vector of design variables, and n denotes the time level and indicates that the grid operator may vary in time. Moreover, different grid operators $\mathbf{G}^n$ may be specified for different component grids. The specific formulations for different grid motions are introduced next.

[‡]Data available online at http://fun3d.larc.nasa.gov [retrieved 20 December 2011].

[§]Private communication with D. Hammond, NASA Langley Research Center, 2011.

A. Grids Undergoing Rigid Motion

For problems in which rigid mesh motion is required, the motion is generated by a 4×4 transform matrix, T, as outlined in [36]. This transform matrix enables general translations and rotations of a grid according to the relation

$$\mathbf{x} = \mathbf{T}\mathbf{x}^0 \tag{19}$$

which moves a point from an initial position $\mathbf{x}^0 = (x^0, y^0, z^0)^T$ to its new position $\mathbf{x} = (x, y, z)^T$:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \tau_x \\ R_{21} & R_{22} & R_{23} & \tau_y \\ R_{31} & R_{32} & R_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^0 \\ y^0 \\ z^0 \\ 1 \end{bmatrix} \tag{20}$$

In an expanded form, $\mathbf{x} = R\mathbf{x}^0 + \boldsymbol{\tau}$. Here, the 3×3 matrix R defines a general rotation and the vector $\boldsymbol{\tau}$ specifies a translation. The matrix $\mathbf{T}$ is generally time dependent. One useful feature of this approach is that multiple transformations telescope via matrix multiplication. This formulation is particularly attractive for composite parent–child body motion, in which the motion of one body is often specified relative to another. The reader is referred to the discussion in [36] for more details. For a rigid-motion formulation, the grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \mathcal{R}^n\mathbf{X}^0 + \boldsymbol{\tau}^n - \mathbf{X}^n \tag{21}$$

or in abbreviated notation,

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \mathbf{T}^n\mathbf{X}^0 - \mathbf{X}^n \tag{22}$$

Here, $\mathbf{X}^0$ and $\mathbf{X}^n$ are the grid vectors at the initial time level and time level n, respectively; $\mathcal{R}^n$ is an $m_x \times m_x$ block-diagonal matrix with 3×3 blocks representing rotation, and m_x being the size of vector $\mathbf{X}^n$; and $\boldsymbol{\tau}^n$ is an m_x-size translation vector. The matrix $\mathcal{R}^n$ and vector $\boldsymbol{\tau}^n$ may explicitly depend on $\mathbf{D}$.

B. Deforming Grids

The simplest example of a deforming grid simulation is a static grid undergoing deformations as a result of a shape-optimization process. In this case, the grid is not time dependent and is modeled as an elastic medium that obeys the elasticity relations of solid mechanics. An auxiliary system of linear partial differential equations (PDEs) is solved to determine the mesh coordinates after each shape update. Discretization of these PDEs yields a system of equations

$$\mathbf{K}(\mathbf{X} - \bar{\mathbf{X}}) = \mathbf{X}_{\text{bound}} - \bar{\mathbf{X}}_{\text{bound}} \tag{23}$$

where $\mathbf{K}$ represents the elasticity coefficient matrix; $\mathbf{X}$ is the vector of grid coordinates being solved for; $\bar{\mathbf{X}}$ is the vector of coordinates of a reference grid; and $\mathbf{X}_{\text{bound}}$ and $\bar{\mathbf{X}}_{\text{bound}}$ are the vectors of corresponding boundary coordinates, complemented by zeros for all interior coordinates. The coefficients of the matrix $\mathbf{K}$ depend on $\bar{\mathbf{X}}$. The material properties of the system given by Eq. (23) are chosen based on either the local cell geometry or proximity to the surface and are invariant with respect to coordinate transformations. The system is solved using a preconditioned generalized minimal residual algorithm. For further details on the approach, see [19,36,53].

For static grid deformation, the only grid operator used at all times is

$$\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv -\mathbf{K}(\mathbf{X} - \bar{\mathbf{X}}) + \mathbf{X}_{\text{bound}} - \bar{\mathbf{X}}_{\text{bound}} \tag{24}$$

where $\mathbf{X}_{\text{bound}}$ may explicitly depend on $\mathbf{D}$, $\bar{\mathbf{X}}$ is an independent grid obtained either from a grid generator or from the previous optimization iteration, and $\bar{\mathbf{X}}_{\text{bound}}$ is the vector of corresponding boundary coordinates.

When time-dependent deforming grids are required, the rigid motion as described in the previous section is not valid. For small relative grid deformations, the linear elasticity equations given by Eq. (23) are solved at each time level with the matrix $\mathbf{K} = \mathbf{K}^0$ computed at the initial time level and fixed throughout the time evolution; $\mathbf{X}^n_{\text{bound}}$ includes the description of the current body positions. The grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv -\mathbf{K}^0(\mathbf{X}^n - \bar{\mathbf{X}}) + \mathbf{X}^n_{\text{bound}} - \bar{\mathbf{X}}_{\text{bound}} \tag{25}$$

C. Parent–Child Motions

Large relative motions are described through parent–child relations, in which the collective motion of a child body is described as the product $\mathbf{T}_p\mathbf{T}_c$, where $\mathbf{T}_p$ is the collective parent transform matrix (which itself can be a chain of parent–child products), and $\mathbf{T}_c$ is the transform matrix describing the motion of the child with respect to the parent. In the current implementation, there is a one-to-one correspondence between moving bodies and component grids. Additional static grids may be associated with the noninertial frame. Thus, a transform matrix describes not only the body motion but may also describe the transformation of the corresponding grid. In general, a parent–child chain of motions can include an arbitrary combination of rigidly moving and deforming overset grids. If a component grid, $\mathbf{X}^n$, is designated as rigid, then all nodes of this grid undergo the same motion described as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{X}^n + \mathbf{T}_p\mathbf{T}_c\mathbf{X}^0 \tag{26}$$

If a component grid is designated as deforming, then the initial grid, $\mathbf{X}^0$, is either given,

$$\mathbf{G}^0(\mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{X}^0 + \bar{\mathbf{X}} \tag{27}$$

or computed from the elasticity equations; Eq. (25). The corresponding body surface undergoes the $\mathbf{T}_p\mathbf{T}_c$ motion, the external boundary and the initial (reference) grid undergo the $\mathbf{T}_p$ motion, and the grid at time level n, $\mathbf{X}^n$, satisfies the elasticity relations

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv -\mathbf{K}^n(\mathbf{X}^n - \mathbf{T}_p\mathbf{X}^0) + \mathbf{X}^n_{bound} - \mathbf{T}_p\mathbf{X}^0_{bound} \tag{28}$$

Here, the matrix $\mathbf{K}^n$ is computed using the moved initial grid $\mathbf{T}_p\mathbf{X}^0$. Note that, because of invariance of the material properties of the elasticity system, the following identity holds:

$$\mathbf{K}^n\mathbf{T}_p = \mathbf{T}_p\mathbf{K}^0 \tag{29}$$

In the current implementation, if any component grid is designated as deforming, then the entire composite grid is designated as deforming, and all component grids are treated as deforming, including those component grids that are in fact rigid. In this scenario, the external boundaries and the reference grid of a rigid component grid are moved with the collective motion of the corresponding body, $\mathbf{T}_p\mathbf{T}_c$, the boundary variations in Eq. (28) become zero, and the obtained grid, $\mathbf{X}^n$, is equivalent to the rigidly moving one; Eq. (26). If all component grids are labeled as either rigid or static, then the composite grid is designated rigid, and all grid points are moved according to Eq. (26).

VI. Cost Functions and Design Variables

The steady-state adjoint implementation described in [18–24] permits multiple objective functions and explicit constraints of the following form, each containing a summation of individual components:

$$f_i = \sum_{j=1}^{J_i} \omega_j (C_j - C_j^*)^{p_j} \tag{30}$$

Here, ω_j represents a user-defined weighting factor, C_j is an aerodynamic coefficient such as the total drag or the pressure or

viscous contributions to such quantities, the superscript $*$ indicates a user-defined target value of C_j, and p_j is a user-defined exponent. Targets are chosen to encourage beneficial changes in the design parameters and are typically far enough from the baseline values to avoid limiting potential improvements. The exponent values are chosen so that f_i is a convex functional, which is important for convergence of gradient-based optimization. The user may specify computational boundaries to which each component function applies. The index i indicates a possibility of introducing several different cost functions or constraints, which may be useful if the user desires separate sensitivities, for example, for lift, drag, pitching moment, etc. The implementation also supports multipoint optimization [20].

For the unsteady formulation, similar general cost functions f_i^n are defined at each time level n. The accumulated cost function f_i can be defined as a discrete sum over a certain time interval $[t_i^1, t_i^2]$:

$$f_i = \sum_{n=N_i^1}^{N_i^2} f_i^n \tag{31}$$

where time levels N_i^1 and N_i^2 correspond to t_i^1 and t_i^2, respectively. The corresponding time integral is approximated as $f_i \Delta t$. The current study also introduces an additional cost function of the following form, which is based on the time-average value of an output:

$$f_i = \left[\left(\frac{1}{(N_i^2 - N_i^1 + 1)} \sum_{n=N_i^1}^{N_i^2} C_i^n\right) - C_i^*\right]^{p_i} \tag{32}$$

The user supplies time intervals over which the cost functions are to be used.

There are three classes of design variables available in the current implementation. The first is composed of global parameters unrelated to the computational grid. These variables include parameters such as the freestream Mach number and angle of attack. Such variables are particularly useful in verifying the implementation of the flowfield adjoint equations.

The second class of design variables provides general shape control of the configuration. The implementation allows the user to employ a geometric parameterization scheme of choice, provided the associated surface grid linearizations are available. For the examples in the current study, the grid parameterization approach described in [54] is used. This approach can be used to define general shape parameterizations of existing grids using a set of aircraft-centric design variables such as camber, thickness, shear, twist, and planform parameters at various locations on the geometry. The user also has the freedom to associate design variables to define more general parameters. In the event that multiple bodies of the same shape are to be designed, such as a set of rotor blades, the implementation allows for a single set of design variables to be used to simultaneously define such bodies. In this fashion, the shape of each body is constrained to be identical throughout the course of the design.

Finally, the third class of design variables governs any kinematics that may be present. The user may invoke simple translation and rotation functions native to the solver; in this case, basic parameters such as frequencies, amplitudes, directional vectors, and centers of rotation are available as design variables. Alternatively, more complicated kinematics and associated design variables may be supplied through a user-defined subroutine satisfying a standard interface. This interface is wrapped with a complex-variable perturbation scheme [12] to numerically evaluate the Jacobian of the specified kinematic motion which is required by the adjoint formulation to follow.

VII. Adjoint Equations

The goal of the design optimization problem for unsteady flows is to choose the design parameters $\mathbf{D}$ to minimize an objective function, $f_{\text{obj}} = f\Delta t$, where f is posed by Eq. (31) or (32) and the subscript i is omitted. For the sake of clarity, the formulation to be presented here is based on a BDF1 scheme for the time derivative, as introduced in Eq. (14). The derivation for higher-order BDF schemes is similar and is presented in the Appendix. Following the methodology described in [5,55], a Lagrangian function is defined as

$$\begin{aligned} L(\mathbf{D},\mathbf{Q},\mathbf{X},\mathbf{\Lambda},\mathbf{\Lambda}_g) &= f\Delta t + ([\mathbf{\Lambda}_g^0]^T\mathbf{G}^0 + [\mathbf{\Lambda}^0]^T\mathbf{R}^{\text{in}})\Delta t \\ &+ \sum_{n=1}^{N}\Bigg\{[\mathbf{\Lambda}_g^n]^T\mathbf{G}^n + [\mathbf{\Lambda}_f^n]^T[\mathbf{A}^n\mathbf{Q}^n] + [\mathbf{\Lambda}_h^n]^T[\mathbf{P}^n\mathbf{Q}^n] \\ &+ [\mathbf{\Lambda}_s^n]^T\left[\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n \right. \\ &\left. + ((\mathbf{I}_s^n\mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n) \circ \mathbf{R}_{\text{GCL}}^n\right]\Bigg\}\Delta t \end{aligned} \tag{33}$$

Here, $\mathbf{\Lambda}_s^n$, $\mathbf{\Lambda}_f^n$, $\mathbf{\Lambda}_h^n$, and $\mathbf{\Lambda}_g^n$ are $m_s \times 1$, $m_f \times 1$, $m_h \times 1$, and $m_x \times 1$ vectors of Lagrange multipliers associated with the solve, fringe, hole, and grid equations, respectively; $[\mathbf{\Lambda}^n]^T = [[\mathbf{\Lambda}_s^n]^T, [\mathbf{\Lambda}_f^n]^T, [\mathbf{\Lambda}_h^n]^T]$; $\mathbf{\Lambda}_s^n = \mathbf{I}_s^n\mathbf{\Lambda}^n$, $\mathbf{\Lambda}_f^n = \mathbf{I}_f^n\mathbf{\Lambda}^n$, and $\mathbf{\Lambda}_h^n = \mathbf{I}_h^n\mathbf{\Lambda}^n$; and $\mathbf{R}^{\text{in}} = 0$ represents the initial conditions. A typical form of the initial conditions is $\mathbf{R}^{\text{in}} \equiv \mathbf{V}^0 \circ (\mathbf{Q}_\infty - \mathbf{Q}^0)$, where $\mathbf{Q}_\infty$ is the freestream solution; other forms, such as a steady-state initial solution, are also possible.

The Lagrangian given by Eq. (33) is differentiated with respect to $\mathbf{D}$, assuming that $\mathbf{V}^n$ depends on $\mathbf{X}^n$; $\mathbf{G}^n$ depends on $\mathbf{X}^n$, $\mathbf{X}^0$, and $\mathbf{D}$; $\mathbf{R}^n$ depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; $\mathbf{R}_{\text{GCL}}^n$ depends on $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; $\mathbf{A}^n$ depends on $\mathbf{X}^n$; $\mathbf{G}^0$ depends on $\mathbf{X}^0$ and $\mathbf{D}$; $\mathbf{R}^{in}$ depends on $\mathbf{Q}^0$, $\mathbf{X}^0$, and $\mathbf{D}$; and $\mathbf{P}^n$, $\mathbf{C}_s^n$, $\bar{\mathbf{C}}_s^n$, $\mathbf{I}_s^n$, $\mathbf{I}_f^n$, and $\mathbf{I}_h^n$ are independent of grid coordinates, solutions, and design parameters.

Regrouping terms to collect the coefficients of $\partial\mathbf{Q}^n/\partial\mathbf{D}$ and equating those coefficients to zero yields the adjoint equations:

S:

$$\begin{aligned} &\frac{1}{\Delta t}\mathbf{C}_s^n \circ \mathbf{V}_s^n \circ \mathbf{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T\mathbf{\Lambda}_h^n \\ &= -\left[\frac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n[\mathbf{I}_s^{n+1}]^T\left[\mathbf{C}_s^{n+1} \circ \left(-\frac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{\text{GCL}}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \end{aligned}$$

F:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_f^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_f^n]^T\mathbf{\Lambda}_h^n \\ &= -\left[\frac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n[\mathbf{I}_s^{n+1}]^T\left[\mathbf{C}_s^{n+1} \circ \left(-\frac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{\text{GCL}}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \end{aligned}$$

H:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T\mathbf{\Lambda}_h^n \\ &= -\left[\frac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n[\mathbf{I}_s^{n+1}]^T\left[\mathbf{C}_s^{n+1} \circ \left(-\frac{1}{\Delta t}\mathbf{V}_s^{n+1} + \mathbf{R}_{\text{GCL}}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] \\ &\text{for } 1 \le n \le N \end{aligned}$$

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^{\text{in}}}{\partial \mathbf{Q}^0}\right]^T\mathbf{\Lambda}^0 = -\left[\frac{\partial f}{\partial \mathbf{Q}^0}\right]^T - [\mathbf{I}_s^1]^T\left[\mathbf{C}_s^1 \circ \left(-\frac{1}{\Delta t}\mathbf{V}_s^1 + \mathbf{R}_{\text{GCL}}^1\right) \circ \mathbf{\Lambda}_s^1\right] \\ &\text{for } n = 0 \end{aligned} \tag{34}$$

where $\mathbf{\Lambda}_s^{N+1} = 0$. The preceding letters indicate the type of points at which the equations are defined; S, F, and H correspond to solve, fringe, and hole points, respectively. Collecting the coefficients of $\partial\mathbf{X}^n/\partial\mathbf{D}$ and equating those coefficients to zero in a similar fashion yields the grid adjoint equations:

$$-\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\left(\mathbf{C}_s^n \circ \frac{\mathbf{Q}_s^n - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t}\right) \odot \frac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^n + \left[\frac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_f^n + \sum_{k=0}^{1} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + ((\mathbf{I}_s^{n+k} \mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^{n+k}}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_s^{n+k} + \left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T$$

for $1 \le n \le N$

$$-\left[\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^0 = \sum_{n=1}^{N} \left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^n + \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}^0 + \sum_{k=1}^{1} \left[\frac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + ((\mathbf{I}_s^k \mathbf{Q}^{k-1}) \circ \mathbf{C}_s^k + \beta \bar{\mathbf{C}}_s^k) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^k}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_s^k + \left[\frac{\partial f}{\partial \mathbf{X}^0}\right]^T$$

for $n = 0$ (35)

Here, $\partial f / \partial \mathbf{X}^n$ is a $1 \times m_x$ row vector, $\partial \mathbf{G}^n / \partial \mathbf{X}^n$ is an $m_x \times m_x$ matrix; $\partial \mathbf{V}_s^n / \partial \mathbf{X}^n$, $\partial \mathbf{R}^n / \partial \mathbf{X}^m$, and $\partial \mathbf{R}_{\mathrm{GCL}}^n / \partial \mathbf{X}^m$ are $m_s \times m_x$ matrices; $\partial (\mathbf{A}^n \mathbf{Q}^n) / \partial \mathbf{X}^n$ is an $m_f \times m_x$ matrix; and $\partial \mathbf{R}^{\mathrm{in}} / \partial \mathbf{X}^0$ is an $m_q \times m_x$ matrix. The operation $\odot$ is an extension of the Hadamard multiplication to a product between an $m_s \times 1$ vector and an $m_s \times m$ matrix, where the second matrix dimension, m, is arbitrary. The operation indicates that the vector multiplies the columns of the matrix in an element-by-element fashion, resulting in a new $m_s \times m$ matrix.

When considering the linearization of $\mathbf{A}^n$, the domain-connectivity information is assumed to be fixed. That is, the weighting coefficients represented by this matrix are considered functions of the mesh coordinates; however, the interpolating elements are considered constant so that the hole-cutting and domain-connectivity algorithms need not be linearized.

With Lagrangian multipliers satisfying Eqs. (34) and (35), the sensitivity derivatives are calculated as follows:

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} \Delta t + \sum_{n=1}^{N} [\mathbf{\Lambda}_g^n]^T \frac{\partial \mathbf{G}^n}{\partial \mathbf{D}} \Delta t + \sum_{n=1}^{N} [\mathbf{\Lambda}_s^n]^T \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + ((\mathbf{I}_s^n \mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^n}{\partial \mathbf{D}}\right] \Delta t + \left([\mathbf{\Lambda}_g^0]^T \frac{\partial \mathbf{G}^0}{\partial \mathbf{D}} + [\mathbf{\Lambda}^0]^T \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{D}}\right]\right) \Delta t \quad (36)$$

where $\partial L / \partial \mathbf{D}$ and $\partial f / \partial \mathbf{D}$ are $1 \times m_d$ row vectors, $\partial \mathbf{G}^n / \partial \mathbf{D}$ is an $m_x \times m_d$ matrix, $\partial \mathbf{R}^n / \partial \mathbf{D}$ and $\partial \mathbf{R}_{\mathrm{GCL}}^n / \partial \mathbf{D}$ are $m_s \times m_d$ matrices, and $\partial \mathbf{R}^{\mathrm{in}} / \partial \mathbf{D}$ is an $m_q \times m_d$ matrix.

To facilitate the solution of Eqs. (34) and (35), the values of $\mathbf{X}^n$, $\partial \mathbf{X}^n / \partial t$, and $\mathbf{Q}^n$ are stored to disk at the conclusion of each physical time step of the flow solution using a strategy designed to minimize file-system overhead. The approach is based on a massively parallel paradigm in which each processor writes to its own unformatted direct-access file at each time step. The data writes are buffered using an asynchronous paradigm so that execution of floating point operations for the subsequent time step may proceed simultaneously. This approach is described and evaluated in [3] and has been found to scale well to several thousand processors using a parallel file system. Rather than recompute the domain-connectivity information during the adjoint solution procedure, a similar I/O paradigm has been implemented to efficiently store this information to disk, although the size of these data is typically an order of magnitude less than the flowfield data. During the solution of Eqs. (34) and (35), data are loaded from disk using a similar paradigm but in reverse, such that data required for the solution at time level $n-1$ are preloaded during the computations for time level n. The sensitivity derivatives [Eq. (36)] are collected during the backward-in-time solution of the adjoint Eqs. (34) and (35), so no disk space is required to store the adjoint solutions.

VIII. Iterative Solution of Equations at Each Time Level

When solving the flow equations, the value of $\mathbf{Q}^{n-1}$ is taken to be an initial approximation for $\mathbf{Q}^n$. The solution of Eqs. (14)–(16) at time level n is obtained through the following iterations, which exploit the form of the Jacobian matrix given by Eq. (17):

F:

$$\mathbf{A}_f^n \Delta \mathbf{Q}_f^{n,m} = -[\mathbf{A}_s^n \mathbf{Q}_s^{n,m} + \mathbf{A}_f^n \mathbf{Q}_f^{n,m} + \mathbf{A}_h^n \mathbf{Q}_h^{n,m}]$$
$$\mathbf{Q}_f^{n,m+1} = \mathbf{Q}_f^{n,m} + \Delta \mathbf{Q}_f^{n,m} \quad (37)$$

S:

$$\left[\frac{1}{\Delta \tau} \mathbf{Diag}(\mathbf{V}_s^n) + \frac{1}{\Delta t} \mathbf{Diag}(\mathbf{C}_s^n \circ \mathbf{V}_s^n) + \frac{\partial \hat{\mathbf{R}}^{n,m}}{\partial \mathbf{Q}_s^n}\right] \Delta \mathbf{Q}_s^{n,m} = -\left[\mathbf{C}_s^n \circ \frac{\mathbf{Q}_s^{n,m} - \mathbf{I}_s^n \mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + \mathbf{R}^{n,m} + ((\mathbf{I}_s^n Q^{n-1}) \circ \mathbf{C}_s^n + \beta \bar{\mathbf{C}}_s^n) \circ \mathbf{R}_{\mathrm{GCL}}^n\right]$$
$$\mathbf{Q}_s^{n,m+1} = \mathbf{Q}_s^{n,m} + \Delta \mathbf{Q}_s^{n,m} \quad (38)$$

H:

$$\mathbf{P}_h^n \Delta \mathbf{Q}_h^{n,m} = -[\mathbf{P}_s^n \mathbf{Q}_s^{n,m+1} + \mathbf{P}_f^n \mathbf{Q}_f^{n,m+1} + \mathbf{P}_h^n \mathbf{Q}_h^{n,m}]$$
$$\mathbf{Q}_h^{n,m+1} = \mathbf{Q}_h^{n,m} + \Delta \mathbf{Q}_h^{n,m} \quad (39)$$

Here, the second superscript m is the iteration count, $\mathbf{R}^{n,m}$ is the spatial nonlinear residual computed for the most recent solution that involves $\mathbf{Q}_f^{n,m+1}$ and $\mathbf{Q}_s^{n,m}$, $\Delta \tau$ is a pseudo-time step, and $\partial \hat{\mathbf{R}}^{n,m} / \partial \mathbf{Q}_s^n$ is the Jacobian of a first-order spatial discretization.

At each iteration, Eq. (37) is solved exactly because $\mathbf{A}_f^n$ is a diagonal matrix, and the fringe solutions are updated first. An approximate solution of the linear system of equations [Eq. (38)] is obtained through several iterations of a multicolor Gauss–Seidel point-iterative scheme, followed by a solution update for $\mathbf{Q}_s^{n,m+1}$. Finally, Eq. (39) is relaxed and solutions at hole points are updated. The convergence rate of the solution at hole points is typically the slowest; relaxation of the pseudo-Laplacian operator is known for poor convergence behavior. If the solution at hole points is decoupled, then its value may be updated only once after the solution at flow and fringe points has been converged.

The adjoint equations are solved backward in time. The solution procedure outlined here is based on the single-grid implementation, which has been previously verified for turbulent flows on three-dimensional unstructured grids undergoing general dynamic motions [5]. The iterative solution of the adjoint equations given by Eq. (34) at time level n is performed in precisely the reverse order as the iterations given by Eqs. (37)–(39):

H:

$$[\mathbf{P}_h^n]^T \Delta \mathbf{\Lambda}_h^{n,m} = -\left[\frac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n [\mathbf{I}_s^{n+1}]^T \left[\mathbf{C}_s^{n+1} \circ \left(-\frac{1}{\Delta t} \mathbf{V}_s^{n+1} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{\Lambda}_s^{n+1}\right] - [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^{n,m} - \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^{n,m} - [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^{n,m}$$
$$\mathbf{\Lambda}_h^{n,m+1} = \mathbf{\Lambda}_h^{n,m} + \Delta \mathbf{\Lambda}_h^{n,m} \quad (40)$$

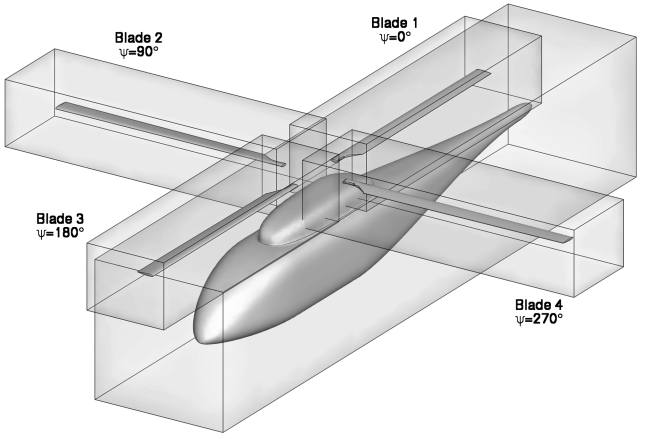


Fig. 1 Near-field view of geometry and composite grid system used for linearization accuracy study.

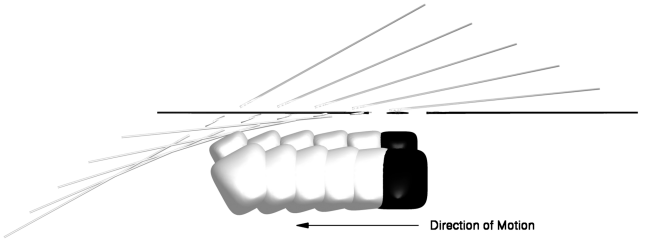


Fig. 2 Imposed motion for linearization accuracy study. Geometry shown every 720 deg of rotor azimuth.

S:

$$\left[\frac{1}{\Delta\tau}\mathbf{Diag}(\mathbf{V}_s^n)+\frac{1}{\Delta t}\mathbf{Diag}(\mathbf{C}_s^n\circ\mathbf{V}_s^n)+\frac{\partial\hat{\mathbf{R}}^{n,m}}{\partial\mathbf{Q}_s^n}\right]\Delta\mathbf{\Lambda}_s^{n,m}$$
$$=-\left[\frac{\partial f}{\partial\mathbf{Q}_s^n}\right]^T-\mathbf{I}_s^n[\mathbf{I}_s^{n+1}]^T\left[\mathbf{C}_s^{n+1}\circ\left(-\frac{1}{\Delta t}\mathbf{V}_s^{n+1}+\mathbf{R}_{\mathrm{GCL}}^{n+1}\right)\circ\mathbf{\Lambda}_s^{n+1}\right]$$
$$-\frac{1}{\Delta t}\mathbf{C}_s^n\circ\mathbf{V}_s^n\circ\mathbf{\Lambda}_s^{n,m}-\left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{Q}_s^n}\right]^T\mathbf{\Lambda}_s^{n,m}-[\mathbf{A}_s^n]^T\mathbf{\Lambda}_f^{n,m}-[\mathbf{P}_s^n]^T\mathbf{\Lambda}_h^{n,m+1}$$
$$\mathbf{\Lambda}_s^{n,m+1}=\mathbf{\Lambda}_s^{n,m}+\Delta\mathbf{\Lambda}_s^{n,m} \tag{41}$$

F:

$$[\mathbf{A}_f^n]^T\Delta\mathbf{\Lambda}_f^{n,m}$$
$$=-\left[\frac{\partial f}{\partial\mathbf{Q}_f^n}\right]^T-\mathbf{I}_f^n[\mathbf{I}_s^{n+1}]^T\left[\mathbf{C}_s^{n+1}\circ\left(-\frac{1}{\Delta t}\mathbf{V}_s^{n+1}+\mathbf{R}_{\mathrm{GCL}}^{n+1}\right)\circ\mathbf{\Lambda}_s^{n+1}\right]$$
$$-\left[\frac{\partial\mathbf{R}^n}{\partial\mathbf{Q}_f^n}\right]^T\mathbf{\Lambda}_s^{n,m+1}-[\mathbf{A}_f^n]^T\mathbf{\Lambda}_f^{n,m}-[\mathbf{P}_f^n]^T\mathbf{\Lambda}_h^{n,m+1}$$
$$\mathbf{\Lambda}_f^{n,m+1}=\mathbf{\Lambda}_f^{n,m}+\Delta\mathbf{\Lambda}_f^{n,m} \tag{42}$$

Solutions for the grid adjoint equations are obtained through relaxation of Eq. (35).

IX. Verification of Adjoint Implementation

To verify the accuracy of the implementation, comparisons are made with results generated through an independent approach based on the use of complex variables. This approach was originally suggested in [12,56], and it was first applied to a Navier–Stokes solver in [57]. Using this formulation, an expression for the derivative of a real-valued function $f(x)$ may be found by expanding the function in a complex-valued Taylor series, using an imaginary perturbation $i\varepsilon$:

$$\frac{\partial f}{\partial x}=\frac{\mathrm{Im}[f(x+i\varepsilon)]}{\varepsilon}+O(\varepsilon^2) \tag{43}$$

The primary advantage of this method is that true second-order accuracy may be obtained by selecting step sizes without concern for subtractive cancellation errors typically present in real-valued Frechet derivatives. Through the use of an automated scripting procedure outlined in [58], this capability can be immediately recovered at any time for the baseline flow solver. For computations using this method, the imaginary step size has been chosen to be 10^{-50}, which highlights the robustness of the complex-variable approach. For each verification test, all equation sets are converged to machine precision for both the complex-variable and adjoint approaches. Since the package described in [46] cannot directly accommodate complex-valued grids and solutions, the integer-valued donor and receptor information is instead transferred to the solver, which performs the requisite complex-valued donor weight computations and solution interpolations. This procedure has been verified to produce identical real components as compared to the routines internal to the package of [46].

The test case used to verify the accuracy of the implementation is based on the rotorcraft configuration shown in Fig. 1. The conventional rotorcraft definition for the azimuth angle ψ is also shown in the figure. The fuselage is described by a component mesh consisting of 88,001 nodes and 505,437 tetrahedral elements. Each of the four rotor blades is modeled using a component grid containing 103,296 nodes and 601,459 tetrahedral elements. The entire configuration is combined with a background grid consisting of 50,156 nodes and 285,587 tetrahedral elements to yield a composite mesh system with 551,341 nodes and 3,196,860 tetrahedral elements.

A very general combination of forced motions is applied to the configuration as follows. The fuselage mesh is subjected to a rigid fixed-rate rotational and translational motion in the starboard direction. The motion of each rotor blade is treated as a child of the fuselage motion, and it consists of an additional rigid fixed-rate rotation in the azimuthal direction. Each blade is also subjected to a final child motion consisting of a forced vertical flapping that is modeled as a 1 deg oscillatory rotation about the rotor hub with a two-per-revolution frequency, and it is accommodated with the deforming mesh mechanics. The background mesh is held fixed in inertial space. The overall motion of the configuration is shown in Fig. 2, while the vertical extent of the blade tip motion due to flapping is shown in Fig. 3. In summary, the composite motion is a family of four

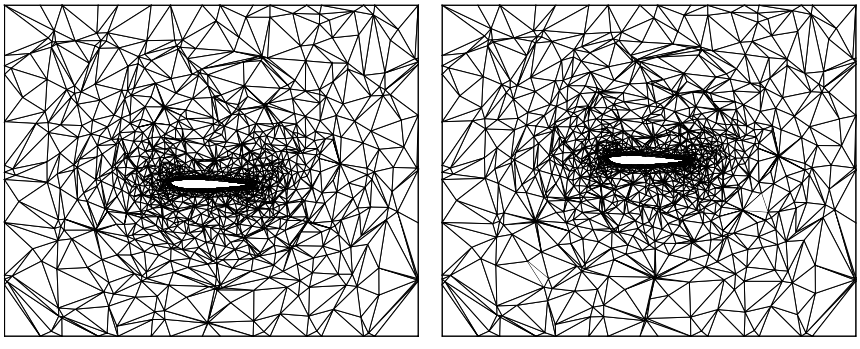


Fig. 3 Cross sections of deforming blade mesh showing maximum vertical displacements at blade tip during linearization accuracy study.

Table 2 Values of sensitivity derivative $\partial C_{L5}/\partial$D for different design variables and temporal discretizations for compressible flow[a,b]

Variable	BDF1	BDF2	BDF2opt	BDF3
Angle of attack	A: 0.116458961683733	A: 0.102099965021956	A: 0.102915752531413	A: 0.103785048456802
	C: 0.116458961683734	C: 0.102099965021956	C: 0.102915752531413	C: 0.103785048456802
Rotation rate: blade 1	A: 0.619149219921508	A: 0.609270815829788	A: 0.592456231940897	A: 0.575091540944799
	C: 0.619149219933539	C: 0.609270815842755	C: 0.592456231953869	C: 0.575091540957581
Shape: blade 2	A: 0.056440771725301	A: 0.064382783171893	A: 0.062734653842921	A: 0.060943525618014
	C: 0.056440771725196	C: 0.064382783171802	C: 0.062734653842842	C: 0.060943525617920
Flap frequency: blade 3	A: -0.414712919056299	A: -0.337250987004676	A: -0.344555513267488	A: -0.352419586848976
	C: -0.414712919056270	C: -0.337250987004642	C: -0.344555513267474	C: -0.352419586848961
Rotation rate: fuselage	A: 6.866802178888885	A: 7.427981437389 84	A: 7.316883059836 01	A: 7.208122185872 93
	C: 6.866802178882 39	C: 7.427981437382 54	C: 7.316883059829 53	C: 7.208122185866 23
Translation rate: fuselage	A: 0.420300051382122	A: 0.400837175635065	A: 0.390973864106570	A: 0.379952931745697
	C: 0.420300051369376	C: 0.400837175622066	C: 0.390973864093789	C: 0.379952931733500
Shape: fuselage	A: -0.007809447236753	A: -0.009590444345683	A: -0.009613538492229	A: -0.009705401931920
	C: -0.007809447236691	C: -0.009590444345727	C: -0.009613538492351	C: -0.009705401931704

[a]Symbols A and C denote adjoint and complex-variable results, respectively.
[b]Discrepancies are shown in bold and underlined.

Table 3 Values of the sensitivity derivative $\partial C_{L5}/\partial$D for different design variables and temporal discretizations for incompressible flow[a,b]

Variable	BDF1	BDF2	BDF2opt	BDF3
Angle of attack	A: 0.000195945789030	A: 0.000234143173131	A: 0.000218182269639	A: 0.000191641169710
	C: 0.000195945789030	C: 0.000234143173131	C: 0.000218182269639	C: 0.000191641169711
Rotation rate: blade 1	A: 0.009518073976865	A: 0.010325090376673	A: 0.010544987182945	A: 0.010757597020150
	C: 0.009518073976838	C: 0.010325090376647	C: 0.010544987182921	C: 0.010757597020128
Shape: blade 2	A: 0.000535025241509	A: 0.000607314158464	A: 0.000618811948355	A: 0.000633736751875
	C: 0.000535025241508	C: 0.000607314158463	C: 0.000618811948355	C: 0.000633736751875
Flap frequency: blade 3	A: -0.004866399384562	A: -0.004825188859067	A: -0.004821787992149	A: -0.004810632891273
	C: -0.004866399384562	C: -0.004825188859067	C: -0.004821787992149	C: -0.004810632891273
Rotation rate: fuselage	A: 0.042649260159755	A: 0.044962632318017	A: 0.044947751807594	A: 0.044876653248215
	C: 0.042649260159807	C: 0.044962632318090	C: 0.044947751807680	C: 0.044876653248312
Translation rate: fuselage	A: 0.010034159304733	A: 0.010404514410124	A: 0.010284602229241	A: 0.010043806857134
	C: 0.010034159304771	C: 0.010404514410192	C: 0.010284602229293	C: 0.010043806857193
Shape: fuselage	A: 0.000087061995334	A: 0.000079589134812	A: 0.000082271937020	A: 0.000086753178814
	C: 0.000087061995336	C: 0.000079589134815	C: 0.000082271937019	C: 0.000086753178823

[a]Symbols A and C denote adjoint and complex-variable results, respectively.
[b]Discrepancies are shown in bold and underlined.

generations, occurring in the following ancestral order from oldest to youngest: inertial reference frame, fuselage motion, azimuthal blade motion, and flapping blade motion.

For the verification of the compressible implementation, the freestream Mach number is 0.1 and the Reynolds number is 4.2 million based on the blade tip speed and chord, and fully turbulent flow is assumed. A similarly scaled Reynolds number of 3.1 million is used for the incompressible verification. The angle of attack is 2 deg, and the advance ratio is 0.12. The physical time step corresponds to one deg of rotation in the azimuthal direction. All of the computations are performed using 128 processors.

Sensitivity derivatives of the lift coefficient for the entire vehicle after five physical time steps are computed using the discrete adjoint and complex-variable approaches. Although the coarse spatial resolution and brief duration of the simulation are not sufficient to resolve the flow physics of the problem, they are adequate to evaluate the discrete consistency of the implementation. Table 2 shows the compressible flow sensitivity derivatives with respect to angle of attack, variables characterizing the rigid-body motions, and parameters describing the blade and fuselage shapes. Results are shown for all of the temporal BDF schemes discussed in Sec. II and the Appendix. Analogous results for the incompressible formulation are shown in Table 3. The results from the discrete adjoint and complex-variable approaches are in very good agreement for all cases; nonmatching digits in the sensitivities are underlined.

X. Large-Scale Test Cases

To evaluate the proposed design methodology, aerodynamic optimizations are performed using three large-scale test cases. The goal is solely to demonstrate the ability of the implementation to successfully reduce each of the stated objective functions while satisfying any constraints present. While details pertaining to the underlying flow physics clearly may be of interest in each case, investigations of that nature are considered beyond the scope of the current effort and are not explored here.

For each case to be shown, the spatial and temporal grid resolutions have been chosen based on a suitable compromise between solution accuracy and computational efficiency. Each optimization is performed on an SGI ICE system using dual-socket hex-core nodes with Intel Xeon X5670 cores in a fully dense configuration. A single additional node is allocated for serial execution of the dynamic hole-cutting library. The computational environment also includes a Lustre-based parallel file system,[¶] and computational statistics include any disk I/O time required to read or write the complete flowfield solution.

As described previously, the implementation supports very general motions including the use of deforming bodies. However, physical models typically responsible for such effects, such as structural models, generally are strong functions of the aerodynamics and require a formal coupling procedure. While the flow solver used in the current study can accommodate such models, the adjoint formulation does not account for such effects at this time. Therefore, to evaluate the current methodology, all large-scale simulations described here rely on forced motions. Development of a more general adjoint formulation required for coupling aerodynamics with other disciplinary models is relegated to future work.

[¶]Data available online at http://wiki.lustre.org/index.php/Main_Page [retrieved 20 December 2011].

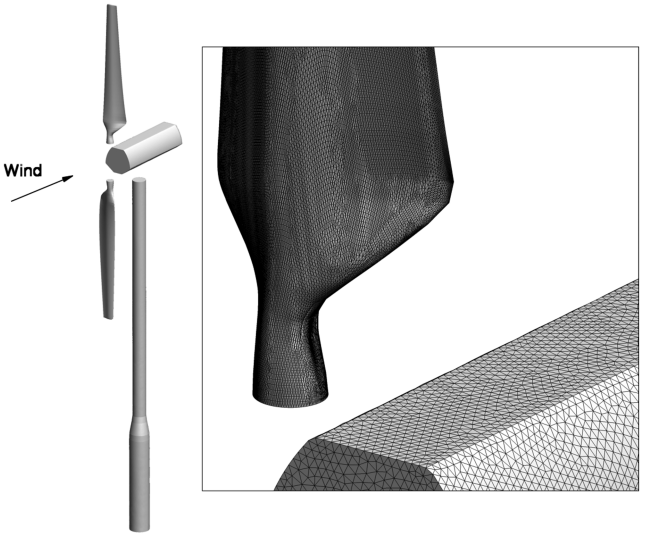


Fig. 4 Wind-turbine configuration and near-field view of surface mesh in hub region.

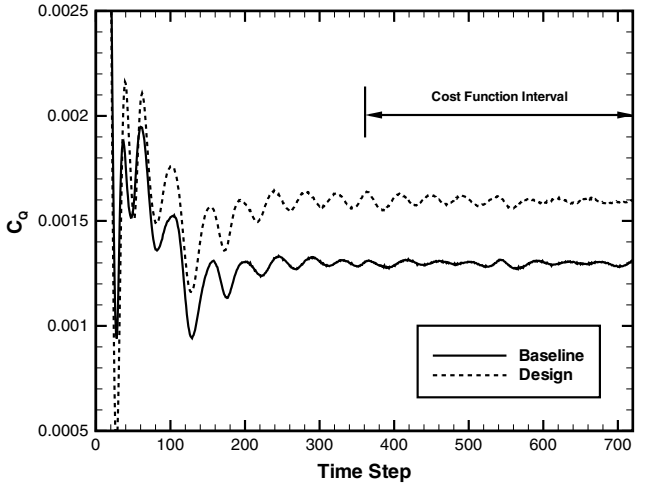


Fig. 5 Baseline and final torque profiles for wind-turbine configuration.

A. National Renewable Energy Laboratory Phase VI Wind Turbine

The first test case is based on the National Renewable Energy Laboratory phase VI wind turbine described in [59]. The geometry is a two-blade upwind configuration with a nacelle and tower. The grid system used here has been developed in [43]. The component grid for each blade consists of 4,510,177 nodes and 26,574,786 tetrahedral elements, and a separate component grid containing the nacelle and tower geometries consists of 971,059 nodes and 5,716,227 tetrahedral elements. The background mesh consists of 4,776,082 nodes and 28,278,639 tetrahedral elements. The resulting composite mesh system contains 14,767,495 nodes and 87,144,438 tetrahedral elements. Views of the configuration and surface meshes are shown in Fig. 4.

The simulation is fully turbulent and is performed using the incompressible form of the governing equations. Standard sea-level conditions are used with a freestream velocity of 15 m/s and aligned with the axis of rotation. The radius of the blades is 5.029 m, and the system rotates at a speed of 72 rpm. The BDF2opt time-integration scheme is used with 100 subiterations and a physical time step corresponding to 1 deg of blade rotation. Solutions are run for 720 time steps or two complete revolutions of the blades. The torque profile for the baseline geometry is shown as the solid line in Fig. 5. After a series of initial transients, the solution quickly settles into a quasi-steady state behavior. The mean value of the torque coefficient $\bar{C}_Q$ measured over the second revolution is 0.00130. An isosurface of the Q criterion [60] is included in Fig. 6.

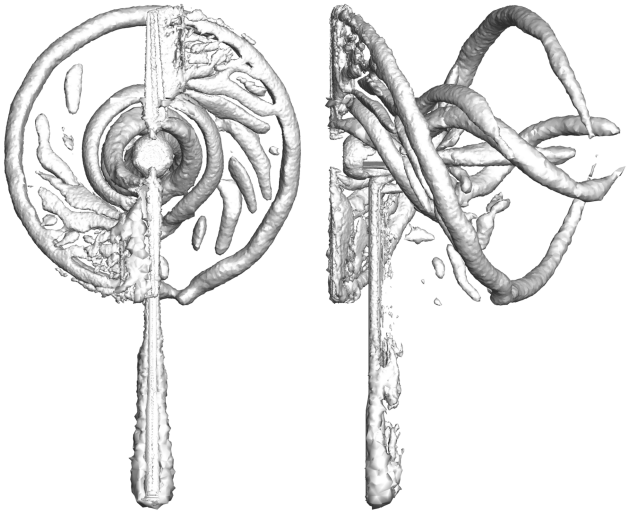


Fig. 6 Front and side views of an isosurface of the Q criterion for the baseline wind-turbine configuration.

The goal of the current test case is to maximize the torque acting on the turbine by altering the blade geometry. The objective function is based on torque values $\hat{C}_Q$, which do not include the nondimensionalization using the reference geometry, and is posed as a discrete summation of the intermediate torque value minus a constant target value over the second revolution:

$$f_{\text{obj}} = \sum_{n=361}^{720} (\hat{C}_Q^n - 2.0)^2 \Delta t \tag{44}$$

The target value of 2.0 has been chosen based on the initial $\hat{C}_Q$ profile, which has a mean value of about 0.5. The objective function could also be formulated in terms of nondimensional torque values; in this case, the target value should be rescaled accordingly. There are a total of 76 design variables, as shown in Fig. 7. These include seven twist values located at various stations along the span of the blade as well as 21 thickness and 48 camber variables distributed across the blade planform. Thinning of the blade is not allowed.

The optimization is performed using 240 computational nodes, or a total of 2,880 processing cores. In this environment, individual flowfield and adjoint solutions require 6.5 and 6 h of wall-clock time, respectively. Approximately 950 GB of disk space are required to store a complete flowfield solution and its associated domain-connectivity data. The package described in [61] is used to perform the optimization.

The convergence history for the optimization is shown in Fig. 8. The objective function has been reduced from its initial value of 69.4 to a final value of 58.7. The final profile for the torque coefficient is included as the dashed line in Fig. 5. The mean value $\bar{C}_Q$ measured over the second revolution is 0.00159, an increase of 22% over the baseline value. Cross sections of the baseline and final blade geometries are shown in Fig. 9. The optimization has increased the thickness across much of the span while also increasing the negative camber in the trailing-edge region.

The optimization procedure for the current test case required nine flow solutions and eight adjoint solutions, for a total of 307,000 CPU hours or 4.5 days of wall-clock time. Although not done for the wind-turbine demonstration, practical constraints such as root-bending moment or thrust constraints are straightforward to incorporate, as shown in Sec. X.C.

B. Biologically Inspired Flapping Wing

The next test case is based on a simple wing configuration undergoing a complex kinematic motion inspired by insects such as the Hawkmoth *manduca sexta* [62]. Such concepts are receiving considerable attention in applications to micro air vehicles [63]. The geometry consists of a rectangular flat plate with semicircular leading

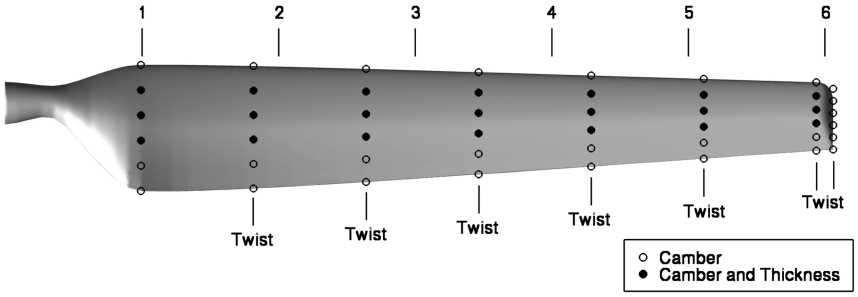


Fig. 7 Blade planform geometry, shape variable locations, and spanwise stations for wind-turbine configuration.

and trailing edges and an aspect ratio of 3.33. The mesh system used for this example has been generated using the approach outlined in [64]. The component mesh containing the wing geometry consists of 3,016,149 nodes and 17,642,078 tetrahedral elements. The background mesh containing the plane of symmetry and outer boundaries consists of 5,339,195 nodes and 31,446,042 tetrahedral elements, yielding a composite mesh with 8,355,344 nodes and 49,088,120 tetrahedral elements. A near-field view of the wing surface mesh is shown in Fig. 10.

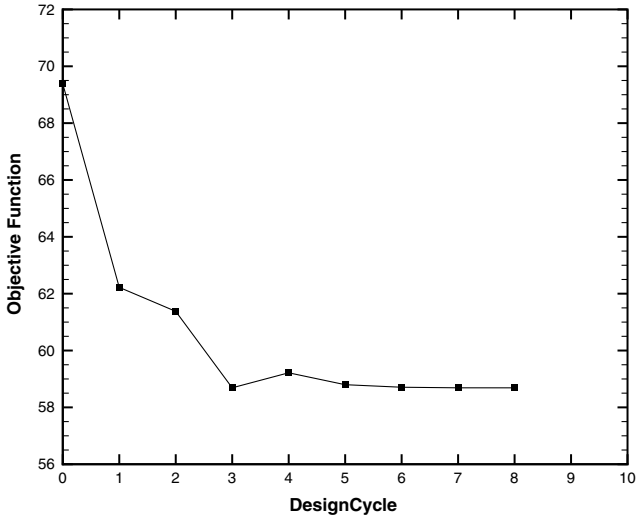


Fig. 8 Convergence of objective function for wind-turbine case.

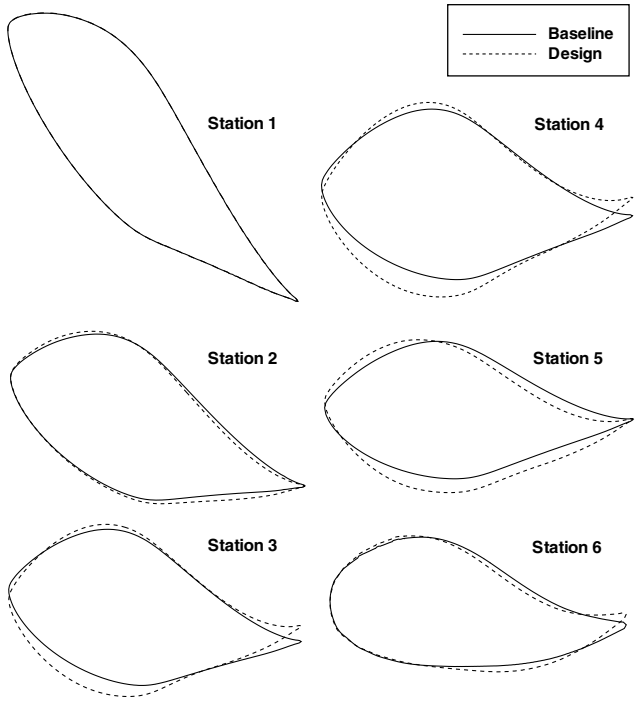


Fig. 9 Baseline and final blade section geometries for the wind-turbine configuration. Vertical scale has been exaggerated for clarity.

The baseline wing is offset 1.33 chord lengths from the plane of symmetry and is assumed to be operating in quiescent conditions. The imposed motion is achieved through the user-defined kinematics interface described previously. Here, time-varying angles describing rotations about the x, y, and z axes are specified in the following general form:

$$\theta_x = A_x[\cos(\omega_{1x}t) - 1] + B_x \sin(\omega_{2x}t)$$
$$\theta_y = A_y[\cos(\omega_{1y}t) - 1] + B_y \sin(\omega_{2y}t)$$
$$\theta_z = A_z[\cos(\omega_{1z}t) - 1] + B_z \sin(\omega_{2z}t) \quad (45)$$

where the amplitudes and frequencies are specified by the user. These angles are used to construct a series of rotation matrices of the form given by Eq. (20). These matrices are then multiplied together to form the final rotation matrix used to specify the current wing position.

In the current example, the baseline motion is a superposition of two oscillatory rotations, each occurring at 26 Hz. The first rotation is a sweeping motion that rotates the wing ± 60 deg about its root chord line. The second rotation is a feathering motion that rotates the wing ± 45 deg about its leading edge. The net effect of this composite motion is a thrust force in the direction from trailing edge to leading edge. Several snapshots of the wing undergoing a period of the baseline motion are shown in Fig. 11.

The Reynolds number based on the wing chord and maximum tip speed is 1280. The governing equations are the incompressible laminar Navier–Stokes equations. The BDF2opt time integration scheme is used with 50 subiterations and a physical time step corresponding to 250 steps per period of the baseline motion. Each simulation is run for 1250 time steps and is performed using 160 computational nodes or a total of 1920 processing cores. Approximately 850 GB of disk space are required to store a complete flowfield solution and its associated domain-connectivity data. Individual flowfield and adjoint solutions require roughly 4 and 3 h of wall-clock time, respectively. The baseline thrust profile exhibits a two-per-cycle periodic behavior, as shown by the solid line in Fig. 12. The mean value of the thrust coefficient C_T measured over the final period is 0.127.

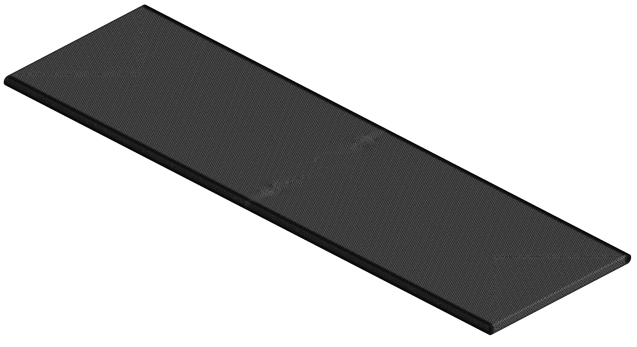
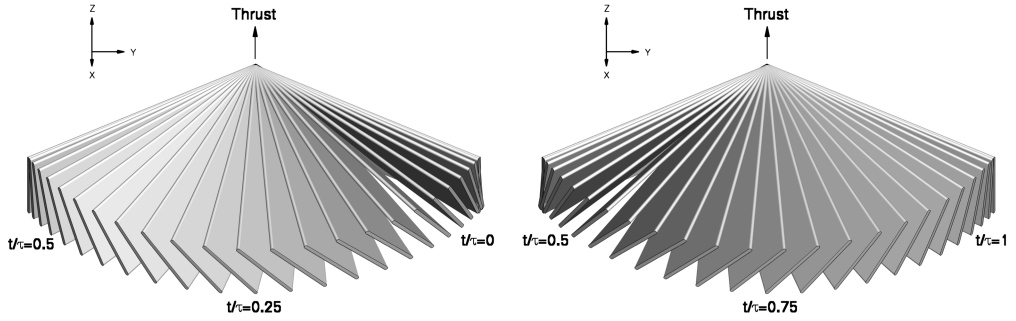


Fig. 10 Surface mesh for flapping-wing case.



a) First half of period b) Second half of period

Fig. 11 Snapshots of baseline flapping-wing motion.

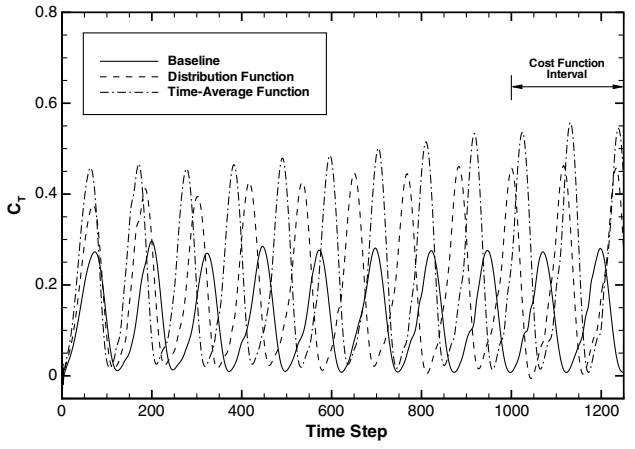


Fig. 12 Baseline and final thrust profiles for flapping-wing case.

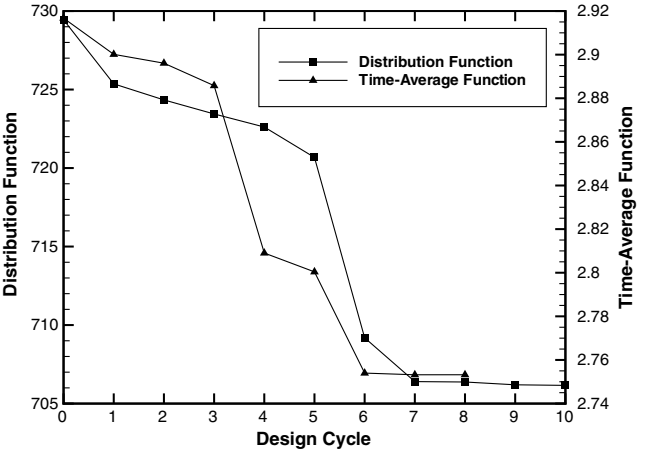


Fig. 13 Convergence of objective functions for flapping-wing case.

The goal of the two test cases presented here is to maximize the thrust coefficient over the final 250 time steps by optimizing the 15 design parameters describing the kinematic motion of the wing, namely, the frequencies, amplitudes, and coordinates of the center of rotation for the composite motion described previously. Both of the optimizations have been performed using the package described in [65]. The first test case uses an objective function based on a target thrust distribution:

$$f_{\text{obj}} = \sum_{n=1,001}^{1,250} (C_T^n - 5.0)^2 \Delta t \tag{46}$$

The second test case uses an objective function that aims to match a single target value for the time-average value of thrust:

$$f_{\text{obj}} = \left[\left(\frac{1}{250} \sum_{n=1,001}^{1,250} C_T^n \right) - 5.0 \right]^2 \Delta t \tag{47}$$

In each case, the target value of 5.0 has been chosen based on the initial thrust profile shown in Fig. 12. Although not shown, physical constraints such as power constraints can also be incorporated in a straightforward fashion.

The convergence history for the objective function based on a target distribution is shown by the square symbols in Fig. 13. The value has been steadily reduced from 729 to 706 over 10 design cycles. Inspection of the final values of the design variables shown in Table 4 reveals moderate changes to all parameters. The final thrust profile is included as the dashed line in Fig. 12. The optimization has not only increased the magnitude of the peaks, but it has also altered the frequency content such that three peaks now occur within the interval used to define the objective function. The mean value of the thrust coefficient over the final 250 time steps is 0.207, a 63% increase over the baseline value. For this test, the optimizer requested 22 flow solutions and 10 adjoint solutions, requiring approximately 227,000 CPU hours or five days of wall-clock time.

The results based on the time-average objective function are included in Fig. 12 as the dashed–dotted line. As in the previous case, the frequency of the signal has been altered to yield three peaks within the objective function interval. The mean value of the thrust coefficient over the final 250 time steps has been increased to 0.265, a 109% increase over the baseline value. The objective function history is plotted in Fig. 13, where it can be seen that the value has been reduced from 2.92 to 2.75 over eight design cycles. Here, the optimizer requested 25 flow solutions and eight adjoint solutions, requiring 238,000 CPU hours or just over five days of wall-clock time.

It should be noted that a series of shape optimizations were also attempted for the current test problem, but they are not presented here. A total of 88 shape parameters describing the twist, shear, thickness, and camber of the wing were used. In general, any shape modification

Table 4 Values of the initial and final design variables for the flapping-wing configuration

Variable	Baseline	Distribution target function	Time-average target function
x-COR[a]	0.000	$0.025c$	$0.027c$
y-COR	0.000	$-0.119c$	$-0.114c$
z-COR	0.000	$0.011c$	$0.012c$
A_x	0.00	0.77	−0.11
B_x	45.00	45.13	45.25
ω_{1x}	163.36	163.45	163.36
ω_{2x}	163.36	177.47	192.77
A_y	0.000	0.30	−0.99
B_y	0.000	−1.50	−0.26
ω_{1y}	163.36	162.76	163.15
ω_{2y}	163.36	163.10	162.97
A_z	−60.00	−62.71	−62.83
B_z	0.00	0.69	−1.55
ω_{1z}	163.36	173.59	189.57
ω_{2z}	163.36	164.41	163.55

[a]COR denotes the center of rotation.

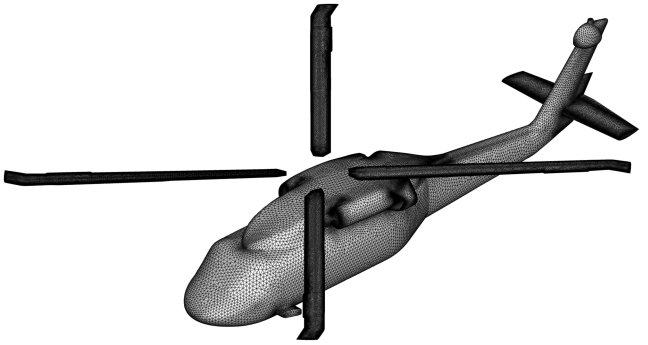


Fig. 14 Surface mesh for UH-60 configuration.

yielding a thrust improvement over one half of the period was seen to be equally detrimental to performance during the opposite half, as each wing surface alternates between pressure and suction conditions. Other forms of shape modification such as planform effects could prove beneficial, although such changes have not been explored here.

C. UH-60A Blackhawk Helicopter

The final test case is based on the UH-60A Blackhawk helicopter configuration [66]. Extensive analysis of this configuration has previously been performed using the solver employed in the current study [39]. The composite grid system used here consists of four identical blade component grids and a single component grid containing the fuselage and outer extent of the computational domain. Each of the blade grids consists of 1,266,525 nodes and 7,476,818 tetrahedral elements, while the fuselage grid contains 4,196,841 nodes and 24,735,227 tetrahedral elements. This results in a composite grid system consisting of 9,262,941 nodes and 54,642,499 tetrahedral elements. The surface mesh for the configuration is shown in Fig. 14.

The governing equations are the compressible Reynolds-averaged Navier–Stokes equations. The simulation is based on a forward flight condition with a blade tip Mach number equal to 0.6378 and a Reynolds number of 7.3 million based on the blade tip chord. The advance ratio is 0.37 and the angle of attack is 0 deg. The rotor blades are subjected to a time-dependent pitching motion that is modeled as a child of the azimuthal rotation and is governed by a sinusoidal variation based on collective and cyclic control inputs:

$$\theta = \theta_c + \theta_{1c} \cos \psi + \theta_{1s} \sin \psi \tag{48}$$

Here, θ is the current blade pitch setting, ψ is the current azimuth position for the blade, θ_c represents the collective control input, and θ_{1c} and θ_{1s} are the lateral and longitudinal cyclic control inputs, respectively. All three control inputs are set to 0 deg at the baseline condition; i.e., the vehicle is initially untrimmed.

The BDF2opt time integration scheme is used with 15 subiterations and a physical time step corresponding to 1 deg of rotor rotation. The simulation is run for two rotor revolutions using 160 computational nodes or a total of 1920 processing cores. In this environment, a single execution of the flow and adjoint solvers requires 2 and 3 h of wall-clock time, respectively. Approximately 650 GB of disk space are required to store a complete flowfield solution and its associated domain-connectivity data.

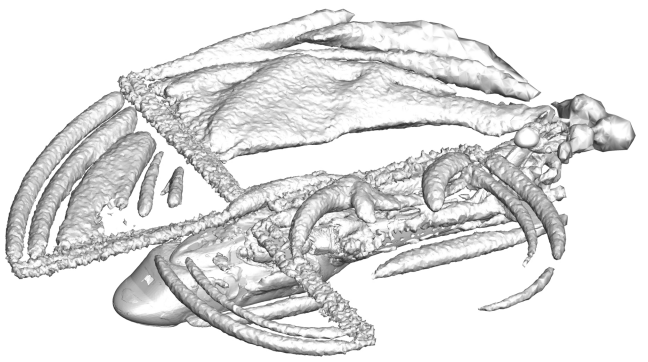


Fig. 15 Isosurface of the Q criterion for the baseline UH-60 configuration.

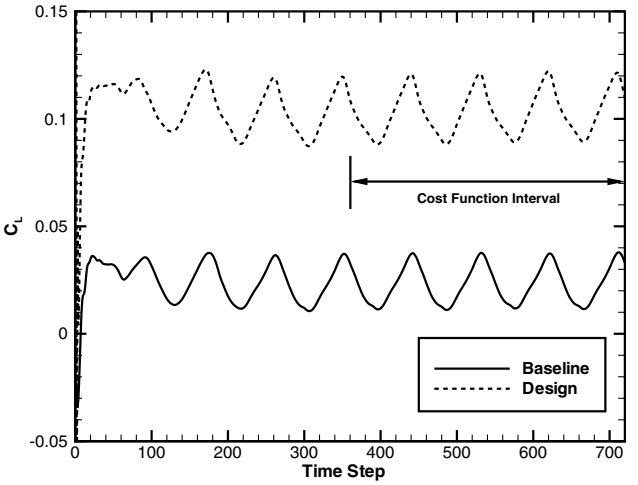


Fig. 16 Baseline and final lift coefficient profiles for the UH-60 configuration.

Figure 15 shows an isosurface of the Q criterion [60] after two rotor revolutions. The vortices emanating from each blade tip and other surfaces of the vehicle are clearly visible. Profiles of the baseline lift and lateral and longitudinal moment coefficients are shown as the solid lines in Figs. 16–18. The values quickly establish a four-per-revolution periodic behavior after 180 deg of blade rotation. The mean value of the lift coefficient over the second rotor revolution is 0.023. The untrimmed flight condition is clearly evident in the nonzero mean values for the two moment coefficients.

The objective for the current test case is to maximize the lift acting on the vehicle while satisfying explicit constraints on the lateral and longitudinal moments such that the final result is a trimmed flight condition. The design variables consist of 64 shape parameters describing the rotor blades, including an 8×4 matrix of 32 thickness variables and 32 camber variables, as shown in Fig. 19. While the camber is allowed to increase or decrease, no thinning of the blade is allowed. In addition, Eq. (48) and its relationship to the blade pitch transform matrix are also linearized, allowing the control variables θ_c, θ_{1c}, and θ_{1s} to also be used as design variables. These control angles are allowed to vary as much as ± 7 deg. Note that parameters describing geometric changes to the fuselage could also be applied; however, without guidance for practical constraints on such changes, such variables are not used here.

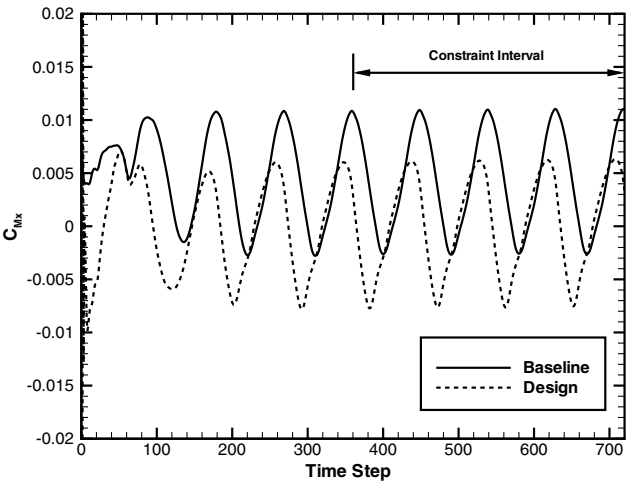


Fig. 17 Baseline and final C_{M_x} profiles for the UH-60 configuration.

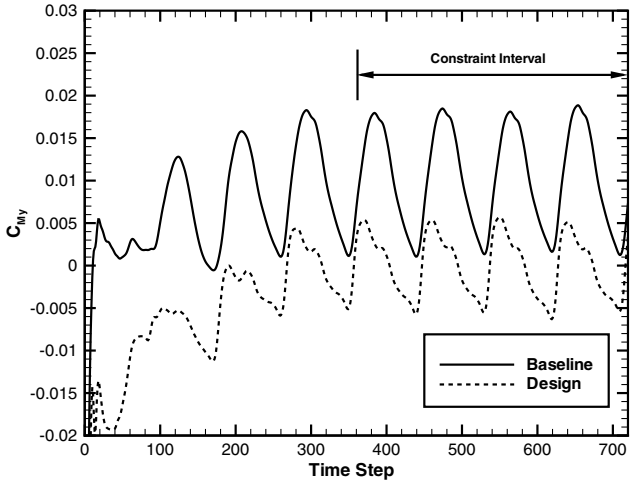


Fig. 18 Baseline and final C_{M_y} profiles for the UH-60 configuration.

The objective function to be minimized is based on the time-average value of the lift coefficient over the second rotor revolution:

$$f_{\text{obj}} = \left[\left(\frac{1}{360}\sum_{n=361}^{720} C_L^n\right) - 2.0\right]^2 \Delta t \tag{49}$$

The target value of 2.0 has been chosen based on the initial lift profile. The explicit constraints on the two moment coefficients are also based on time-average values over the same interval:

$$g_1 = \frac{1}{360}\sum_{n=361}^{720} C_{M_x}^n \Delta t \tag{50}$$

$$g_2 = \frac{1}{360}\sum_{n=361}^{720} C_{M_y}^n \Delta t \tag{51}$$

The constraints are considered satisfied if $g_1 = g_2 = 0$, within a feasibility tolerance of ± 0.0001. The optimization is performed using the package described in [61]. Note that the treatment of the moment constraints requires two additional adjoint solutions to compute the associated gradient vectors. These additional solutions are obtained simultaneously with the adjoint computation for the lift objective using the procedure outlined in [24] to accommodate multiple right-hand-side vectors in Eqs. (34)–(36).

1. *Design Results*

Figure 20 shows the convergence of the objective function and constraints after three design cycles. The optimization procedure quickly locates a feasible region in the design space based on the two moment constraints, and the value of the objective function is successfully reduced. The final unsteady lift profile is included as the dashed line in Fig. 16. The mean value has been substantially increased to a value of 0.103. The final unsteady profiles for the lateral and longitudinal moment coefficients are included as the dashed lines in Figs. 17 and 18, respectively. Each of the new profiles has the desired zero mean value, indicating that the final design is trimmed for level flight within the requested tolerance.

Based on the spanwise blade stations noted in Fig. 19, cross sections of the initial and final blade geometries are shown in Fig. 21. The shape changes are confined to the aft sections of the outer portion of the blade, where the camber has been increased. The final value of the collective input θ_c is 6.71 deg, while the final values for the cyclic inputs θ_{1c} and θ_{1s} are 2.58 and −7.00 deg, respectively. The entire optimization procedure requiring four flow solutions and four adjoint solutions took approximately 20 h of wall-clock time, or 38,400 CPU hours.

2. *Interpretation of the Adjoint Solution*

Typical qualitative features of unsteady adjoint solutions are shown in Fig. 22 for the objective function given by Eq. (49). The figure depicts centerline contours of the adjoint solution for the energy equation at time level $n = 420$. The contours represent the instantaneous sensitivity of the objective function to a source term

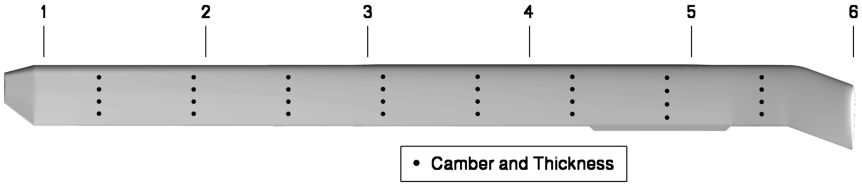


Fig. 19 Blade planform geometry, shape variable locations, and spanwise stations for UH-60 configuration.

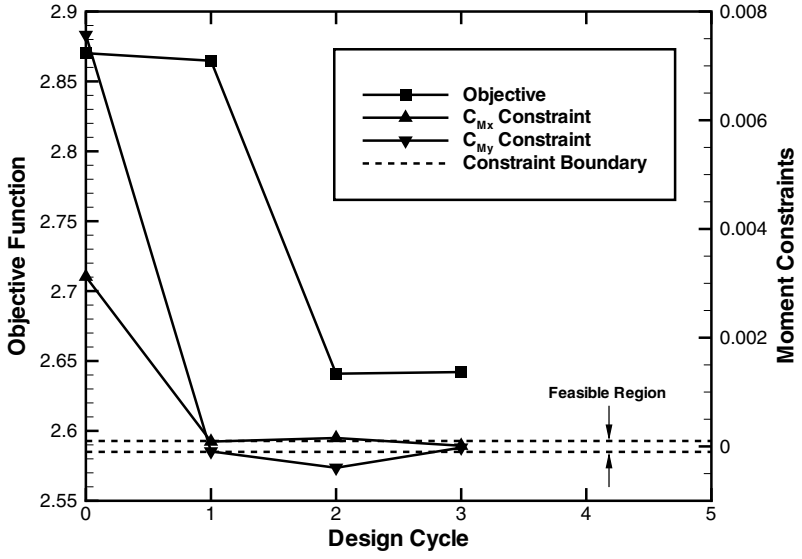


Fig. 20 Convergence of the objective function and constraints for the UH-60 configuration.

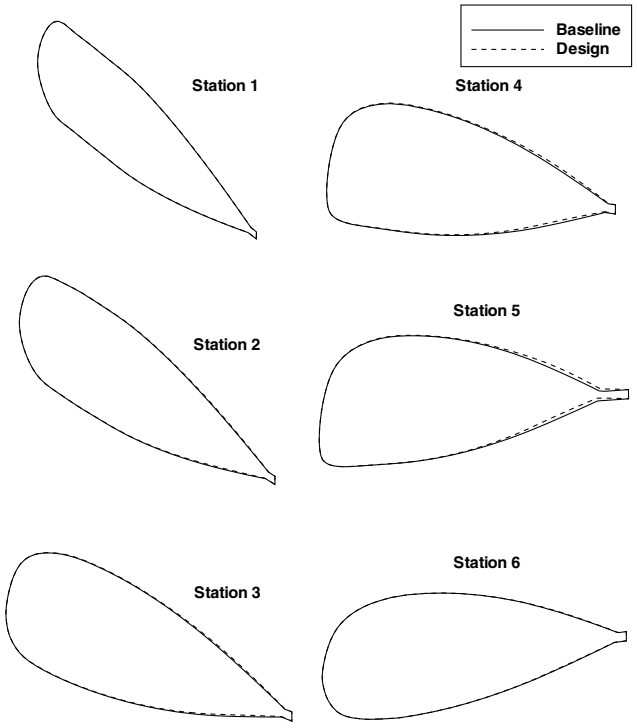


Fig. 21 Baseline and final blade section geometries for the UH-60 configuration. Vertical scale has been exaggerated for clarity.

applied to the energy equation at each point in the domain. Similar to steady-flow objective functions based on surface integrals [67–70], the time-average value of the lift is particularly sensitive to information propagating along the stagnation streamline and impacting the nose of the fuselage. In addition, Fig. 22 highlights several features emanating from the rotor blades as they pass through the cutting plane. These features are loosely analogous to unsteady-flow phenomena such as vortex sheets and tip vortices commonly seen in forward solutions for rotor flows, as shown in Fig. 15. However, unlike the forward problem, the features shown in the adjoint solution propagate in the upstream direction as the adjoint system is integrated in reverse physical time, indicating the sensitivity of the objective function to disturbances upstream.

In design optimization, the adjoint solutions are combined with the linearizations of the residual operators with respect to design variables to yield sensitivity derivatives. Alternatively, the adjoint solutions may be combined with local residuals to provide rigorous error estimates or with (local estimates of) the truncation errors to guide mesh adaptation. Although these applications are not the focus of the current work, adjoint-based adaptation methodologies [14] offer many compelling advantages over traditional feature-based mesh adaptation techniques that fail to identify important regions such as those containing the upstream features highlighted in Fig. 22.

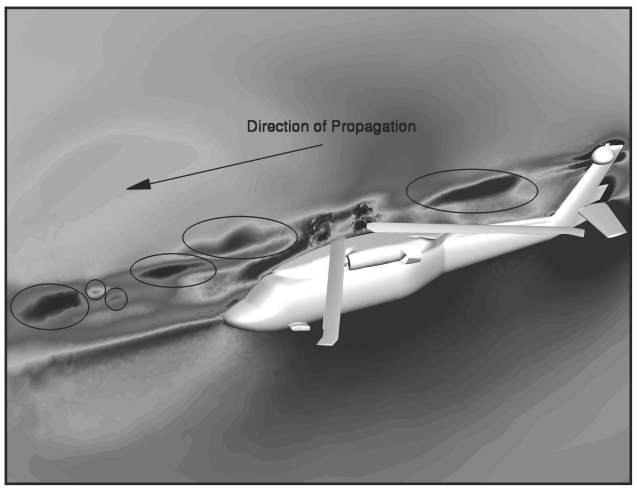


Fig. 22 Snapshot of adjoint solution for the energy equation using an objective function based on a time-average lift coefficient. Highlighted features originate on blade surfaces and propagate upstream.

XI. Conclusions

A general verified methodology for adjoint-based design optimization of unsteady turbulent flows on dynamic overset unstructured mesh systems has been presented. The formulation is valid for compressible and incompressible forms of the Reynolds-averaged Navier–Stokes equations. The implementation is amenable to massively parallel computing environments and has been verified through the use of an independent technique based on a complex-variable formulation. Several large-scale optimizations have been demonstrated for complex flowfields involving a wind-turbine configuration, a flapping wing, and a realistic helicopter geometry subject to trimming constraints. The objective functions have been successfully reduced in each case and all constraints present have been satisfied.

Although the demonstrated methodology provides a practical approach to optimization of general unsteady aerodynamic flows, a wide range of research topics remains to be explored. Locally optimal, reduced-order model, and checkpointing techniques offer the potential to greatly reduce storage requirements. Multifidelity optimization algorithms should be exploited where possible to reduce dependence on high-fidelity simulations. Convergence acceleration techniques can clearly have a direct impact on computational cost. Simultaneous adjoint-based error estimation and mesh adaptation approaches are very attractive in establishing rigorous gridding requirements and eliminating user interaction. Extension of adjoint-based methods to multidisciplinary optimization beyond the scope of computational fluid dynamics is essential for making significant impacts on the current paradigm for design of aerospace vehicles and other areas of applications. Finally, advancements in the fields of computer science, software development, and high-performance computing must continue to be leveraged to the greatest extent possible.

Acknowledgments

The first author is supported by the subsonic rotary-wing element of the Fundamental Aeronautics Project within the NASA Aeronautics Research Mission Directorate. The second author acknowledges the support from NASA through contract NNL07AA23C and cooperative agreement NNL09AA00A. Both authors wish to thank Robert Biedron of NASA Langley Research Center and Nail Yamaleev of North Carolina A&T State University for many helpful conversations pertaining to the current work. The enabling contributions of Elizabeth Lee-Rausch and William Jones of NASA Langley Research Center in the areas of grid generation and parameterization are very much appreciated. The work has also greatly benefitted from the high-performance computing research conducted by Dana Hammond of NASA Langley Research Center. Discussions with David Darmofal, Qiqi Wang, and Josh Krakos of the Massachusetts Institute of Technology (MIT) and Yuping Sun of Vestas Wind Systems were very useful. Ralph Noack of Pennsylvania State University is acknowledged for his support and assistance in using his overset grid libraries. Thanks are due to C. Eric Lynch of Naval Air Systems Command for providing the mesh used for the wind-turbine example. The authors are grateful to John Moore of MIT and the staff of the Air Force Research Laboratory for providing the mesh and problem definition for the flapping-wing case.

Appendix: Adjoint Equations for Higher-Order Backward Difference Schemes

Discrete conservation laws employing high-order temporal BDF schemes as introduced in Eq. (6) are defined as

$$\mathbf{C}_s^n \circ \left[a\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n + c\frac{\mathbf{I}_s^n\mathbf{Q}^{n-2} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n\mathbf{V}^{n-2} + d\frac{\mathbf{I}_s^n\mathbf{Q}^{n-3} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{I}_s^n\mathbf{V}^{n-3} \right] + \mathbf{R}^n + ((\mathbf{I}_s^n\mathbf{Q}^{n-1}) \circ \mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n) \circ \mathbf{R}_{\mathrm{GCL}}^n = 0 \tag{A1}$$

Proceeding as before, the Lagrangian can be written as

$$\begin{aligned} L(\mathbf{D},\mathbf{Q},\mathbf{X},\mathbf{\Lambda},\mathbf{\Lambda}_g) &= f\Delta t + \sum_{n=1}^{N} [\mathbf{\Lambda}_g^n]^T\mathbf{G}^n\Delta t \\ &+ \sum_{n=1}^{N}\Bigg\{ [\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n]^T\Bigg[a\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ \mathbf{V}_s^n \\ &+ c\frac{\mathbf{I}_s^n\mathbf{Q}^{n-2} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n\mathbf{V}^{n-2}) \\ &+ d\frac{\mathbf{I}_s^n\mathbf{Q}^{n-3} - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \circ (\mathbf{I}_s^n\mathbf{V}^{n-3}) \Bigg] \\ &+ [\mathbf{\Lambda}_s^n]^T[\mathbf{R}^n + ((\mathbf{I}_s^n\mathbf{Q}^{n-1})^{\circ}\mathbf{C}_s^n + \beta\bar{\mathbf{C}}_s^n)^{\circ}\mathbf{R}_{\mathrm{GCL}}^n] \\ &+ [\mathbf{\Lambda}_f^n]^T[\mathbf{A}^n\mathbf{Q}^n] + [\mathbf{\Lambda}_h^n]^T[\mathbf{P}^n\mathbf{Q}^n] \Bigg\}\Delta t \\ &+ (f^0 + [\mathbf{\Lambda}_g^0]^T\mathbf{G}^0 + [\mathbf{\Lambda}^0]^T\mathbf{R}^{\mathrm{in}})\Delta t \end{aligned} \tag{A2}$$

On time levels 1 and 2, the time derivatives are assumed to be discretized with the BDF1 and BDF2 schemes, respectively. Taking into account the dependencies on data at time levels $n-2$ and $n-3$, the adjoint equations are obtained as follows:

S:

$$\begin{aligned} &\frac{a}{\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T\mathbf{\Lambda}_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T \\ &- \mathbf{I}_s^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \Lambda_s^{n+1}\Bigg] + [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \Lambda_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \Lambda_s^{n+3}\right]\Bigg\} \end{aligned}$$

F:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T\Lambda_s^n + [\mathbf{A}_f^n]^T\Lambda_f^n + [\mathbf{P}_f^n]^T\Lambda_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T \\ &- \mathbf{I}_f^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \Lambda_s^{n+1}\Bigg] + [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \Lambda_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \Lambda_s^{n+3}\right]\Bigg\} \end{aligned}$$

H:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T\Lambda_s^n + [\mathbf{A}_h^n]^T\Lambda_f^n + [\mathbf{P}_h^n]^T\Lambda_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T \\ &- \mathbf{I}_h^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \Lambda_s^{n+1}\Bigg] + [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \Lambda_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \Lambda_s^{n+3}\right]\Bigg\} \end{aligned}$$

for $3 \le n \le N$ (A3)

S:

$$\begin{aligned} &\frac{3}{2\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T\mathbf{\Lambda}_h^n = \\ &-\left[\frac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\Bigg] \\ &+ [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right]\Bigg\} \end{aligned}$$

F:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_f^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_f^n]^T\mathbf{\Lambda}_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T \\ &- \mathbf{I}_f^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\Bigg] + [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right]\Bigg\} \end{aligned}$$

H:

$$\begin{aligned} &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T\mathbf{\Lambda}_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T \\ &- \mathbf{I}_h^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{a}{\Delta t}\mathbf{V}_s^{n+1} - \frac{c}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} - \frac{d}{\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-2} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\Bigg] + [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right]\Bigg\} \end{aligned}$$

for $n = 2$ (A4)

$$\begin{aligned} S:\ &\frac{1}{\Delta t}\mathbf{V}_s^n \circ \mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_s^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_s^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_s^n]^T\mathbf{\Lambda}_h^n \\ &= -\left[\frac{\partial f}{\partial \mathbf{Q}_s^n}\right]^T - \mathbf{I}_s^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \frac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\Bigg] \\ &+ [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right]\Bigg\} \\ F:\ &\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_f^n}\right]^T\mathbf{\Lambda}_s^n + [\mathbf{A}_f^n]^T\mathbf{\Lambda}_f^n + [\mathbf{P}_f^n]^T\mathbf{\Lambda}_h^n \\ &= -\left[\frac{\partial f}{\partial \mathbf{Q}_f^n}\right]^T - \mathbf{I}_f^n\Bigg\{[\mathbf{I}_s^{n+1}]^T\Bigg[\left(-\frac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \frac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{\mathrm{GCL}}^{n+1}\right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1}\Bigg] \\ &+ [\mathbf{I}_s^{n+2}]^T\left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}\right] \\ &+ [\mathbf{I}_s^{n+3}]^T\left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n\right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}\right]\Bigg\} \end{aligned}$$

H:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}_h^n}\right]^T \mathbf{\Lambda}_s^n + [\mathbf{A}_h^n]^T \mathbf{\Lambda}_f^n + [\mathbf{P}_h^n]^T \mathbf{\Lambda}_h^n = -\left[\frac{\partial f}{\partial \mathbf{Q}_h^n}\right]^T - \mathbf{I}_h^n \left\{ [\mathbf{I}_s^{n+1}]^T \left[\left(-\frac{3}{2\Delta t}\mathbf{V}_s^{n+1} - \frac{1}{2\Delta t}\mathbf{I}_s^{n+1}\mathbf{V}^{n-1} + \mathbf{R}_{\mathrm{GCL}}^{n+1} \right) \circ \mathbf{C}_s^{n+1} \circ \mathbf{\Lambda}_s^{n+1} \right] + [\mathbf{I}_s^{n+2}]^T \left[\left(\frac{c}{\Delta t}\mathbf{I}_s^{n+2}\mathbf{V}^n \right) \circ \mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2} \right] + [\mathbf{I}_s^{n+3}]^T \left[\left(\frac{d}{\Delta t}\mathbf{I}_s^{n+3}\mathbf{V}^n \right) \circ \mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3} \right] \right\}$$

for $n = 1$

$$\left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{Q}^0}\right]^T \mathbf{\Lambda}^0 = -\left[\frac{\partial f}{\partial \mathbf{Q}^0}\right]^T - \left\{ [\mathbf{I}_s^1]^T \left[\left(-\frac{1}{\Delta t}\mathbf{V}_s^1 + \mathbf{R}_{GCL}^1 \right) \circ \mathbf{C}_s^1 \circ \mathbf{\Lambda}_s^1 \right] + [\mathbf{I}_s^2]^T \left[\left(\frac{1}{2\Delta t}\mathbf{I}_s^2\mathbf{V}^0 \right) \circ \mathbf{C}_s^2 \circ \mathbf{\Lambda}_s^2 \right] + [\mathbf{I}_s^3]^T \left[\left(\frac{d}{\Delta t}\mathbf{I}_s^3\mathbf{V}^0 \right) \circ \mathbf{C}_s^3 \circ \mathbf{\Lambda}_s^3 \right] \right\}$$

for $n = 0$ (A5)

The corresponding grid adjoint equations are obtained as follows: Assuming $\mathbf{\Lambda}^{N+1} = \mathbf{\Lambda}^{N+2} = \mathbf{\Lambda}^{N+3} = 0$:

$$-\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[a\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \odot \frac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n} \right]^T (\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n) + \left[c\frac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}) + \left[d\frac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}) + \left[\frac{\partial (\mathbf{A}^n\mathbf{Q}^n)}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_f^n + \sum_{k=0}^{3} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + ((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^{n+k}}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_s^{n+k} + \left[\frac{\partial f}{\partial \mathbf{X}^n} \right]^T$$

for $3 \le n \le N$

$$-\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\frac{3}{2}\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \odot \frac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n} \right]^T (\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n) + \left[c\frac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}) + \left[d\frac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}) + \left[\frac{\partial (\mathbf{A}^n\mathbf{Q}^n)}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_f^n + \sum_{k=0}^{3} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + ((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^{n+k}}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_s^{n+k} + \left[\frac{\partial f}{\partial \mathbf{X}^n} \right]^T$$

for $n = 2$

$$-\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n}\right]^T \mathbf{\Lambda}_g^n = \left[\frac{\mathbf{Q}_s^n - \mathbf{I}_s^n\mathbf{Q}^{n-1}}{\Delta t} \odot \frac{\partial \mathbf{V}_s^n}{\partial \mathbf{X}^n} \right]^T (\mathbf{C}_s^n \circ \mathbf{\Lambda}_s^n) + \left[c\frac{\mathbf{I}_s^{n+2}\mathbf{Q}^n - \mathbf{I}_s^{n+2}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+2}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}) + \left[d\frac{\mathbf{I}_s^{n+3}\mathbf{Q}^n - \mathbf{I}_s^{n+3}\mathbf{Q}^{n+1}}{\Delta t} \odot \left(\mathbf{I}_s^{n+3}\frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \right) \right]^T (\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^{n+3}) + \left[\frac{\partial (\mathbf{A}^n\mathbf{Q}^n)}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_f^n + \sum_{k=0}^{3} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + ((\mathbf{I}_s^{n+k}\mathbf{Q}^{n+k-1}) \circ \mathbf{C}_s^{n+k} + \beta \bar{\mathbf{C}}_s^{n+k}) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^{n+k}}{\partial \mathbf{X}^n} \right]^T \mathbf{\Lambda}_s^{n+k} + \left[\frac{\partial f}{\partial \mathbf{X}^n} \right]^T$$

for $n = 1$

$$-\left[\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^0 = \sum_{n=1}^{N} \left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0}\right]^T \mathbf{\Lambda}_g^n + \left[\frac{1}{2}\frac{\mathbf{I}_s^2\mathbf{Q}^0 - \mathbf{I}_s^2\mathbf{Q}^1}{\Delta t} \odot \left(\mathbf{I}_s^2\frac{\partial \mathbf{V}^0}{\partial \mathbf{X}^0} \right) \right]^T (\mathbf{C}_s^{n+2} \circ \mathbf{\Lambda}_s^{n+2}) + \left[d\frac{\mathbf{I}_s^3\mathbf{Q}^0 - \mathbf{I}_s^3\mathbf{Q}^1}{\Delta t} \odot \left(\mathbf{I}_s^3\frac{\partial \mathbf{V}^0}{\partial \mathbf{X}^0} \right) \right]^T (\mathbf{C}_s^{n+3} \circ \mathbf{\Lambda}_s^3) + \left[\frac{\partial R^{\mathrm{in}}}{\partial \mathbf{X}^0} \right]^T \mathbf{\Lambda}^0 + \sum_{k=1}^{3} \left[\frac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + ((\mathbf{I}_s^k\mathbf{Q}^{k-1}) \circ \mathbf{C}_s^k + \beta \bar{\mathbf{C}}_s^k) \odot \frac{\partial \mathbf{R}_{\mathrm{GCL}}^k}{\partial \mathbf{X}^0} \right]^T \mathbf{\Lambda}_s^k + \left[\frac{\partial f}{\partial \mathbf{X}^0} \right]^T$$

for $n = 0$ (A6)

The sensitivity derivative for the higher-order BDF schemes is evaluated using Eq. (36).

References

[1] Newman, J., III, Taylor, A., III, Barnwell, R., Newman, P., and Hou, G.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96. doi:10.2514/2.2416

[2] Peter, J., and Dwight, R., "Numerical Sensitivity Analysis for Aerodynamic Optimization: A Survey of Approaches," *Computers and Fluids Journal*, Vol. 39, No. 3, 2010, pp. 373–391. doi:10.1016/j.compfluid.2009.09.013

[3] Nielsen, E., and Jones, W., "Integrated Design of an Active Flow Control System Using a Time-Dependent Adjoint Method," *Mathematical Modeling of Natural Phenomena*, Vol. 6, No. 3, 2011, pp. 1195–1206. doi:10.1051/mmnp/20116306

[4] Rumpfkeil, M., and Zingg, D., "General Framework for the Optimal Control of Unsteady Flows with Applications," AIAA Paper 2007-1128, 2007.

[5] Nielsen, E., Diskin, B., and Yamaleev, N., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, Vol. 48, No. 6, 2010, pp. 1195–1206. doi:10.2514/1.J050035

[6] Nadarajah, S., and Jameson, A., "Optimal Control of Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods," AIAA Paper 2002-5436, 2002.

[7] Muldoon, F., "Control of a Simplified Unsteady Film-Cooling Flow Using Gradient-Based Optimization," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2443–2458. doi:10.2514/1.34120

[8] Mavriplis, D., "Solution of the Unsteady Discrete Adjoint for Three-Dimensional Problems on Dynamically Deforming Unstructured Meshes," AIAA Paper 2008-0727, 2008.

[9] Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA Paper 2008-5810, 2008.

[10] Hicks, R., and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412. doi:10.2514/3.58379

[11] Baysal, O., and Eleshaky, M., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681–688. doi:10.1115/1.2926534

[12] Lyness, J. N., "Numerical Algorithms Based on the Theory of Complex Variables," *Proceedings of the 22nd ACM National Conference*, Thomas Book Company, Washington, D.C., 1967, pp. 124–134.

[13] Pironneau, O., "On Optimum Profiles in Stokes Flow," *Journal of Fluid Mechanics*, Vol. 59, No. 1, 1973, pp. 117–128. doi:10.1017/S002211207300145X

[14] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. doi:10.2514/1.J050073

[15] Wang, Q., *Uncertainty Quantification for Unsteady Fluid Flow Using Adjoint-Based Approaches*, Ph.D. Thesis, Stanford Univ., Stanford, CA, Dec. 2008.

[16] Rallabhandi, S., "Sonic Boom Adjoint Methodology and Its Applications," AIAA Paper 2011-3497, 2011.

[17] Lee, J.-D., and Jameson, A., "Natural-Laminar-Flow Airfoil and Wing Design by Adjoint Method and Automatic Transition Prediction," AIAA Paper 2009-0897, 2009.

[18] Nielsen, E., and Anderson, W., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419. doi:10.2514/2.640

[19] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155–1163. doi:10.2514/2.1765

[20] Nielsen, E., Lee-Rausch, E., and Jones, W., "Adjoint-Based Design of Rotors Using the Navier–Stokes Equations in a Noninertial Reference Frame," *Journal of Aircraft*, Vol. 47, No. 2, 2010, pp. 638–646. doi:10.2514/1.46044

[21] Nielsen, E. J., and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids Using Complex Variables," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 827–836. doi:10.2514/1.15830

[22] Nielsen, E. J., and Park, M. A., "Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953. doi:10.2514/1.16052

[23] Nielsen, E. J., *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier–Stokes Equations and a Discrete Adjoint Formulation*, Ph.D. Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, Dec. 1998.

[24] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids Journal*, Vol. 33, No. 9, 2004, pp. 1131–1155. doi:10.1016/j.compfluid.2003.09.005

[25] Lee, B., and Kim, C., "Aerodynamic Shape Optimization Using Discrete Adjoint Formulation Based on Overset Mesh Technique," *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, Egmond aan Zee, The Netherlands, Sept. 2006.

[26] Elliott, J., "Discrete Adjoint Analysis and Optimization with Overset Grid Modeling of the Compressible High-Re Navier–Stokes Equations," *Proceedings of the 6th Overset Grid and Solution Technology Symposium*, Oct. 2002.

[27] Lei, Z., and Makino, Y., "CFD-Based Aerodynamic Shape Optimization for Japanese Experimental Supersonic Transport," *Proceedings 24th International Congress of Aeronautical Sciences ICAS 2004*, ICAS Paper 2004-4.3.3, Optimage, Ltd., Sept. 2004.

[28] Liao, W., and Tsai, H., "Aerodynamic Design Optimization by the Adjoint Equation Method on Overset Grids," AIAA Paper 2006-0054, 2006.

[29] Cambier, L., and Veuillot, J.-P., "Status of the elsA CFD Software for Flow Simulation and Multidisciplinary Applications," AIAA Paper 2008-0664, 2008.

[30] O'Brien, D. M., Jr., *Analysis of Computational Modeling Techniques for Complete Rotorcraft Configurations*, Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, May 2006.

[31] Horn, R. A., and Johnson, C. R., *Matrix Analysis*, Cambridge Univ. Press, New York, 1985, p. 321.

[32] Chorin, A., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, 1967, pp. 12–26. doi:10.1016/0021-9991(67)90037-X

[33] Nyukhtikov, M., Smelova, N., Mitchell, B., and Holmes, D., "Optimized Dual-Time Stepping Technique for Time-Accurate Navier–Stokes Calculation," *Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines*, edited by Hall, K., Kielb, R., and Thomas, J., Springer, 2006.

[34] Vatsa, V., Carpenter, M., and Lockard, D., "Re-Evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications," AIAA Paper 2010-0122, 2010.

[35] Hirt, C., Amsden, A., and Cook, J., "An Arbitrary Lagrangian–Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics*, Vol. 14, No. 3, 1974, pp. 227–257. doi:10.1016/0021-9991(74)90051-5

[36] Biedron, R., and Thomas, J., "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications," AIAA Paper 2009-1360, 2009.

[37] Thomas, P., and Lombard, C., "Geometrical Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037. doi:10.2514/3.61273

[38] Noack, R., Boger, D., Kunz, R., and Carrica, P., "SUGGAR++: An Improved General Overset Grid Assembly Capability," AIAA Paper 2009-3992, 2009.

[39] Biedron, R., and Lee-Rausch, E., "Computation of UH-60A Airloads Using CFD/CSD Coupling On Unstructured Meshes," *American Helicopter Society 67th Annual Forum Proceedings*, 2011.

[40] Lee-Rausch, E., and Biedron, R., "Simulation of an Isolated Tiltrotor in Hover with an Unstructured Overset-Grid RANS Solver," *American Helicopter Society 65th Annual Forum Proceedings*, 2009.

[41] Biedron, R., and Lee-Rausch, E., "Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling," AIAA Paper 2008-7341, 2008.

[42] O'Brien, D., Jr., and Smith, M., "Analysis of Rotor-Fuselage Interactions Using Various Rotor Models," AIAA Paper 2005-0468, 2005.

[43] Lynch, E., *Advanced CFD Methods for Wind Turbine Analysis*, Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, Jan. 2011.

[44] Abras, J., *Enhancement of Aeroelastic Rotor Airload Prediction Methods*, Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, May 2009.

[45] Abras, J., Lynch, C., and Smith, M., "Advances in Rotorcraft Simulations with Unstructured CFD," *American Helicopter Society 63rd Annual Forum Proceedings*, 2007.

[46] Noack, R., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," AIAA Paper 2005-5116, 2005.

[47] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids Journal*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X

[48] Anderson, W., Rausch, R., and Bonhaus, D., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 391–408. doi:10.1006/jcph.1996.0219

[49] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5

[50] Neel, R., Godfrey, A., and McGrory, W., "Low-Speed, Time-Accurate Validation of GASP Version 4," AIAA Paper 2005-686, 2005.

[51] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, No. 1, 1994, pp. 5–21.

[52] Lee-Rausch, E., Hammond, D., Nielsen, E., Pirzadeh, S., and Rumsey, C., "Application of the FUN3D Unstructured-Grid Navier–Stokes Solver to the 4th AIAA Drag Prediction Workshop Cases," AIAA Paper 2010-4551, 2010.

[53] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. doi:10.1137/0907058

[54] Samareh, J. A., "A Novel Shape Parameterization Approach," NASA TM-1999-209116, May 1999.

[55] Yamaleev, N., Diskin, B., and Nielsen, E., "Adjoint-Based Methodology for Time-Dependent Optimization," AIAA Paper 2008-5857, 2008.

[56] Lyness, J., and Moler, C., "Numerical Differentiation of Analytic Functions," *SIAM Journal on Numerical Analysis*, Vol. 4, No. 2, 1967, pp. 202–210.
doi:10.1137/0704019

[57] Anderson, W., Newman, J., Whitfield, D., and Nielsen, E., "Sensitivity Analysis for the Navier–Stokes Equations on Unstructured Meshes Using Complex Variables," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63.
doi:10.2514/2.1270

[58] Kleb, W. L., Nielsen, E. J., Gnoffo, P. A., Park, M. A., and Wood, W. A., "Collaborative Software Development in Support of Fast Adaptive AeroSpace Tools (FAAST)," AIAA Paper 2003-3978, 2003.

[59] Hand, M., Simms, D., Fingersh, L., Jager, D., Cotrell, J., Schreck, S., and Larwood, S., "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," National Renewable Energy Lab. TP-500-29955, Golden, CO, 2001.

[60] Jeong, J., and Hussain, F., "On the Identification of a Vortex," *Journal of Fluid Mechanics*, Vol. 285, 1995, pp. 69–94.
doi:10.1017/S0022112095000462

[61] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL 5.0: A FORTRAN Package for Nonlinear Programming," TR-SOL-94, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1995.

[62] Willmott, A., and Ellington, C., "The Mechanics of Flight in the Hawkmoth *Manduca Sexta* I. Kinematics of Hovering and Forward Flight," *Journal of Experimental Biology*, Vol. 200, No. 21, 1997, pp. 2705–2722.

[63] McClung, A., "Influence of Structural Flexibility on Flapping Wing Propulsion," Ph.D. Thesis, Air Force Inst. of Technology, Wright–Patterson AFB, OH, Nov. 2009.

[64] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625.
doi:10.2514/3.12701

[65] Kaufman, L., and Gay, D., *PORT Library: Optimization and Mathematical Programming: User's Manual*, Bell Laboratories, Madison, WI, 1997.

[66] Bousman, W., and Kufeld, R., "UH-60A Airloads Catalog," NASA TM-2005-212827, Aug. 2005.

[67] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.
doi:10.1016/S0021-9991(03)00074-3

[68] Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003-3849, 2003.

[69] Lee-Rausch, E. M., Park, M. A., Jones, W. T., Hammond, D. P., and Nielsen, E. J., "Application of a Parallel Adjoint-Based Error Estimation and Anisotropic Grid Adaptation for Three-Dimensional Aerospace Configurations," AIAA Paper 2005-4842, 2005.

[70] Park, M., Lee-Rausch, E., and Rumsey, C., "FUN3D and CFL3D Computations for the First High Lift Prediction Workshop," AIAA Paper 2011-0936, 2011.

W. K. Anderson
Associate Editor