AIAA JOURNAL
Vol. 48, No. 6, June 2010

Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids

Eric J. Nielsen*
NASA Langley Research Center, Hampton, Virginia 23681
Boris Diskin†
National Institute of Aerospace, Hampton, Virginia 23666
and
Nail K. Yamaleev‡
North Carolina Agricultural and Technical State University, Greensboro, North Carolina 27411

DOI: 10.2514/1.J050035

An adjoint-based methodology for design optimization of unsteady turbulent flows on dynamic unstructured grids is described. The implementation relies on an existing unsteady three-dimensional unstructured grid solver capable of dynamic mesh simulations and discrete adjoint capabilities previously developed for steady flows. The discrete equations for the primal and adjoint systems are presented for the backward-difference family of time-integration schemes on both static and dynamic grids. The consistency of sensitivity derivatives is established via comparisons with complex-variable computations. The current work is believed to be the first verified implementation of an adjoint-based optimization methodology for the true time-dependent formulation of the Navier–Stokes equations in a practical computational code. Large-scale shape optimizations are demonstrated for turbulent flows over a tilt-rotor geometry and a simulated aeroelastic motion of a fighter jet.

Nomenclature

a, b, c, d	=	temporal coefficients
C	=	aerodynamic coefficient
C_T	=	rotor thrust coefficient
$\mathbf{D}$	=	vector of design variables
E	=	total energy per unit volume
$\mathbf{F}$	=	flux vector
$\mathbf{F}_i, \mathbf{F}_v$	=	inviscid and viscous flux vectors
f	=	cost function
f, s	=	general functions
$\mathbf{G}$	=	grid operator
i	=	$\sqrt{-1}$
i, j, k, n	=	indices
in	=	quantity at initial conditions
J	=	number of cost function components
$\boldsymbol{K}$	=	$m_x \times m_x$ linear elasticity coefficient matrix
L	=	Lagrangian function
m_q	=	size of vector $\mathbf{Q}$
m_x	=	size of vector $\mathbf{X}$
N	=	number of time levels
$\hat{\mathbf{n}}$	=	outward-pointing normal vector
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
$\mathbf{R}$	=	$m_q \times 1$ vector of spatial undivided residuals
$\boldsymbol{R}$	=	$m_x \times m_x$ block-diagonal rotation matrix
R	=	3×3 rotation matrix
$\boldsymbol{R}_{\mathrm{GCL}}$	=	$m_q \times m_q$ diagonal geometric conservation law residual matrix
S	=	control volume surface area
T	=	4×4 transform matrix
t	=	time
u, v, w	=	Cartesian components of velocity
V	=	control volume
$\boldsymbol{V}$	=	$m_q \times m_q$ 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
ε	=	perturbation
Θ	=	rotor blade collective setting
$\boldsymbol{\Lambda}_f$	=	$m_q \times 1$ flowfield adjoint variable
$\boldsymbol{\Lambda}_g$	=	$m_x \times 1$ grid adjoint variable
ρ	=	density
$\boldsymbol{\tau}$	=	3×1 translation vector
$\boldsymbol{\tau}$	=	$m_x \times 1$ translation vector
Ψ	=	rotor azimuth
ω	=	cost function component weight
∞	=	quantity at freestream conditions
$*$	=	target quantity

Received 30 June 2009; revision received 24 September 2009; accepted for publication 24 October 2009.

*Research Scientist, Computational AeroSciences Branch. Senior Member AIAA.

†Associate Fellow; also Visiting Associate Professor, Department of Mechanical and Aerospace Engineering, University of Virginia. Member AIAA.

‡Associate Professor, Department of Mathematics. Member AIAA.

Introduction

AS COMPUTATIONAL fluid dynamics (CFD) tools become more efficient, accurate, and robust, their role in the analysis and design of new aerospace configurations continues to increase. Computational methods have already become a major integrated component of industrial practices. The use of CFD has been traditionally confined to the steady regime; however, with recent algorithmic improvements and the persistent growth of computational power, CFD methods have begun to make substantial inroads in simulating unsteady flow phenomena. Target applications for these methods are widely abundant; typical examples might include the prediction of aeroelastic characteristics, maneuvering flight conditions, 6 degree-of-freedom simulations, specified motion problems, or flow control simulations, among many others.

In recent years, steady-state CFD methods have been targeted for use in automated design optimization frameworks. In gradient-based design approaches, one of the major challenges is to obtain sensitivity information for the flowfield at a reasonable cost. Conventional black-box finite difference methods [1] suffer from well-known step-size limitations and incur a computational expense that grows linearly with the number of design variables. Forward, or direct, differentiation methods [2] and techniques based on the use of complex variables [3] mitigate the step-size limitation but still suffer from excessive cost in the presence of many design variables, as is often the case with aerodynamic design applications.

Adjoint methods provide a powerful alternative for aerodynamic sensitivity analysis. In this approach, the sensitivities of an objective function are determined through the solution of an auxiliary, or adjoint, set of equations. Adjoint methods may be further categorized into either continuous or discrete approaches, depending on the order in which the governing equations are differentiated and discretized. One of the features of the discrete approach is that it allows one to account for mesh variation as well; a second adjoint system can be solved to linearize the relationship between the design variables and the mesh operator as described in [4]. The principal advantage of the adjoint approach is that the computational cost is independent of the number of design variables; a rigorous sensitivity analysis for hundreds of variables can be performed at a cost equivalent to the solution of the governing equations themselves. For examples of the use of such methods, see the references cited in [5].

The role of adjoint-based methodologies in mesh adaptation strategies should also be noted. Whereas many traditional mesh adaptation schemes rely on heuristic connections between solution gradient information and local mesh spacing requirements, the adjoint equations establish a rigorous mathematical connection between solution accuracy and the computational grid. The approach has proven quite powerful and has enjoyed success where traditional feature-based approaches have consistently failed. Fidkowski and Darmofal [6] provide a review of recent applications and an extensive list of references on the subject.

Some recent examples of adjoint-based strategies for unsteady aerospace applications are given in [7–14]. The goal of the current work is to extend the time-dependent adjoint formulation for static grids introduced in [14] and the steady-state discrete adjoint capability developed in [4,15–19] to the three-dimensional time-dependent Euler and Reynolds averaged Navier–Stokes equations. The present approach and implementation are valid for unsteady flows on various grids including static grids, dynamic grids undergoing rigid motion, and general morphing grids governed by a mesh deformation scheme based on a linear elasticity analog. This work is believed to be the first verified implementation of an adjoint-based optimization methodology for the true time-dependent formulation of the Navier–Stokes equations in a practical computational code. In the following sections, the unsteady governing equations are presented as well as various mesh motion strategies. These are followed by the derivation of the discrete adjoint equations for the flowfield and mesh, including details concerning their implementation. Examples demonstrating the discrete consistency of the implementation and applications of the design optimization framework to large-scale problems are also shown.

Flowfield Equations

Using the approach outlined in [20], the unsteady Euler and Navier–Stokes equations may be written in the following form for both moving and stationary control volumes:

$$\frac{\partial}{\partial t}\int_V \mathbf{q}\,\mathrm{d}V + \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v)\cdot\hat{\mathbf{n}}\,\mathrm{d}S = 0 \tag{1}$$

where V is the control volume bounded by the surface $\mathrm{d}V$. The vector $\mathbf{q}$ represents the conserved variables for mass, momentum, and energy, and the vectors $\mathbf{F}_i$ and $\mathbf{F}_v$ denote the inviscid and viscous fluxes, respectively. Note that, for a moving control volume, the inviscid flux vector must account for the difference in the fluxes due to the movement of control volume faces. Given a flux vector $\mathbf{F}$ on a static grid, the corresponding flux $\mathbf{F}_i$ on a moving grid can be defined as $\mathbf{F}_i = \mathbf{F} - \mathbf{q}(\mathbf{W}\cdot\hat{\mathbf{n}})$, where $\mathbf{W}$ is a local face velocity and $\hat{\mathbf{n}}$ is an outward-pointing unit face normal.

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

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

the conservation equations take the form

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v)\cdot\hat{\mathbf{n}}\,\mathrm{d}S = 0 \tag{3}$$

where the conserved variables and inviscid flux vectors are defined as $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, E]^T$ and

$$\begin{aligned}\mathbf{F}_i = &\begin{bmatrix} \rho(u - W_x) \\ \rho u(u - W_x) + p \\ \rho v(u - W_x) \\ \rho w(u - W_x) \\ (E + p)(u - W_x) + W_x p \end{bmatrix}\hat{\mathbf{i}} \\ &+ \begin{bmatrix} \rho(v - W_y) \\ \rho u(v - W_y) \\ \rho v(v - W_y) + p \\ \rho w(v - W_y) \\ (E + p)(v - W_y) + W_y p \end{bmatrix}\hat{\mathbf{j}} \\ &+ \begin{bmatrix} \rho(w - W_z) \\ \rho u(w - W_z) \\ \rho v(w - W_z) \\ \rho w(w - W_z) + p \\ (E + p)(w - W_z) + W_z p \end{bmatrix}\hat{\mathbf{k}}\end{aligned} \tag{4}$$

The viscous flux vector $\mathbf{F}_v$ is not explicitly shown here. The equations are closed with the perfect gas equation of state and an appropriate turbulence model for the eddy viscosity. Finally, it is worth noting that, for the special case of a spatially and temporally constant state vector, for example, $\mathbf{Q} = (1, 0, 0, 0, 0)^T$, the conservation equations reduce to the geometric conservation law (GCL) [21]:

$$\frac{\partial V}{\partial t} = \oint_{\mathrm{d}V} \mathbf{W}\cdot\hat{\mathbf{n}}\,\mathrm{d}S \tag{5}$$

In computational practice, the discrete GCL residual is added to the flow equations to preserve a constant solution on dynamic grids [20].

The flow solver used in the current work is described in [15,20,22].[§] 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 that may or may not contain overset grid topologies.

In the current study, the spatial 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 [23], and viscous fluxes are formed using an approach equivalent to a finite element Galerkin procedure. 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-

[§]Data available online at http://fun3d.larc.nasa.gov [retrieved 4 January 2010].

equation approach of Spalart and Allmaras [24]. Massively parallel scalability is achieved through domain decomposition and message passing communication.

An approximate solution of the linear system of equations formed within each time step is obtained through several iterations of a multicolor Gauss–Seidel point-iterative scheme. The turbulence model is integrated all the way to the wall without the use of wall functions and is solved separately from the mean flow equations at each time step with a time-integration and linear system solution scheme identical to that employed for the mean flow equations.

Grid Equations

The general grid equations can be defined in the form $\mathbf{G}^n(\mathbf{X}, \mathbf{D}) = 0$, where $\mathbf{X}$ is the 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. The specific formulations for different grid motions are introduced next.

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 [20]. This transform matrix enables general translations and rotations of the grid according to the relation

$$\mathbf{x} = T\mathbf{x}^0 \tag{6}$$

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

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \tau_x \\ R_{21} & R_{22} & R_{23} & \tau_y \\ R_{31} & R_{32} & R_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^0 \\ y^0 \\ z^0 \\ 1 \end{bmatrix} \tag{7}$$

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 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 [20] for more details. For this formulation, the grid operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{X}^0, \mathbf{D}) \equiv \boldsymbol{R}^n\mathbf{X}^0 + \boldsymbol{\tau}^n - \mathbf{X}^n \tag{8}$$

where $\mathbf{X}^0$ and $\mathbf{X}^n$ are the grid vectors at the initial and nth time levels, respectively; $\boldsymbol{R}^n$ 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 $\boldsymbol{R}^n$ and vector $\boldsymbol{\tau}^n$ may explicitly depend on $\mathbf{D}$.

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

$$\boldsymbol{K}\,\mathbf{X} = \mathbf{X}_{\text{surf}} \tag{9}$$

where $\boldsymbol{K}$ represents the elasticity coefficient matrix, $\mathbf{X}$ is the vector of grid coordinates being solved for, and $\mathbf{X}_{\text{surf}}$ is the vector of updated surface coordinates, complemented by zeros for all interior coordinates.

The coefficients of the matrix $\boldsymbol{K}$ depend on the coordinates of the grid. In the approach followed here, the elasticity equations are discretized on the grid corresponding to the initial time level. Thus, the grid at the initial level satisfies the nonlinear equations

$$\boldsymbol{K}^0(\mathbf{X}^0, \mathbf{D})\mathbf{X}^0 = \mathbf{X}^0_{\text{surf}} \tag{10}$$

The material properties of the system are chosen based on the local cell geometry and proximity to the surface, and the system is solved using a preconditioned generalized minimal residual algorithm. For further details on the approach, see [17,20,25].

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

$$\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv \mathbf{X}_{\text{surf}} - \boldsymbol{K}\mathbf{X} \tag{11}$$

where $\mathbf{X}_{\text{surf}}$ may explicitly depend on $\mathbf{D}$. There are situations in which time-dependent deforming grids are required, including aeroelastic deflections of the surface, for which the rigid motion as described in the previous section is not valid. Instead, a morphing mesh formulation is used. In this approach, the linear elasticity equations given by Eq. (9) are solved at each time level with the matrix $\boldsymbol{K} = \boldsymbol{K}^0$ computed at the initial time level and fixed throughout the time evolution; the vector $\mathbf{X}^n_{\text{surf}}$ represents the current body positions. For morphing grids, the operator at time level n is defined as

$$\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv \mathbf{X}^n_{\text{surf}} - \boldsymbol{K}^0\mathbf{X}^n \tag{12}$$

When the surface motion is governed by the rigid motion relations given by Eq. (6), $\mathbf{X}^n_{\text{surf}}$ can be further specified as $\mathbf{X}^n_{\text{surf}} = \boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n$.

Cost Functions

The steady-state adjoint implementation described in [4,15–19] permits multiple objective functions and explicit constraints of the following form, each containing a summation of individual components:

$$f_i = \sum_{j=1}^{J_i} \omega_j (C_j - C_j^*)^{p_j} \tag{13}$$

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 chosen so that f_i is a convex functional. 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.

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

$$f_i = \sum_{n=N_i^1}^{N_i^2} f_i^n \Delta t \tag{14}$$

where time levels N_i^1 and N_i^2 correspond to t_i^1 and t_i^2, respectively. The user now supplies time intervals over which the cost functions are to be used.

Derivation of the Time-Dependent Adjoint Equations

To derive the time-dependent form of the adjoint equations, the methodology developed in [14] is used. The governing equations given by Eq. (3) are rewritten as

$$\frac{\partial(\mathbf{Q}V)}{\partial t} + \mathbf{R} = 0, \qquad \mathbf{R} \equiv \oint_{\mathrm{d}V} (\mathbf{F}_i - \mathbf{F}_v) \cdot \hat{\mathbf{n}}\, \mathrm{d}S \tag{15}$$

Using a first-order backward difference (BDF1) in time, the equations can be evaluated at time level n as follows:

$$\boldsymbol{V}^n \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \boldsymbol{R}^n_{\mathrm{GCL}} \mathbf{Q}^{n-1} = 0 \tag{16}$$

Here, $\boldsymbol{V}^n$ and $\boldsymbol{R}^n_{\mathrm{GCL}}$ are $m_q \times m_q$ diagonal matrices, m_q is the length of vector $\mathbf{Q}^n$, the GCL is discretized in a consistent fashion as

$$\frac{1}{\Delta t}(\boldsymbol{V}^n - \boldsymbol{V}^{n-1}) = \boldsymbol{R}^n_{\mathrm{GCL}} \tag{17}$$

and $\mathbf{R}^n$ is the spatial undivided residual. Recall that $\mathbf{R}^n$ and $\boldsymbol{R}^n_{\mathrm{GCL}}$ depend on grids at the current and previous time levels. Note also that although the BDF1 scheme has been shown here for the sake of simplicity, the derivations for higher-order temporal schemes are similar and included in the Appendix.

The discrete adjoint-based optimization methodology is based on the method of Lagrange multipliers, which is used to enforce the governing equations as constraints. For the sake of simplicity in the following derivations, a single cost function is assumed; therefore, the index i is omitted. For the time-dependent equations, the Lagrangian functional is defined as follows:

$$\begin{aligned} L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \boldsymbol{\Lambda}_f, \boldsymbol{\Lambda}_g) = {} & \sum_{n=1}^{N} f^n \Delta t + \sum_{n=1}^{N} [\boldsymbol{\Lambda}_f^n]^T \left(\boldsymbol{V}^n \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} \right. \\ & \left. + \mathbf{R}^n + \boldsymbol{R}^n_{\mathrm{GCL}} \mathbf{Q}^{n-1} \right) \Delta t + \sum_{n=1}^{N} [\boldsymbol{\Lambda}_g^n]^T \mathbf{G}^n \Delta t \\ & + (f^0 + [\boldsymbol{\Lambda}_f^0]^T \mathbf{R}^{\mathrm{in}}) \Delta t + [\boldsymbol{\Lambda}_g^0]^T \mathbf{G}^0 \Delta t \end{aligned} \tag{18}$$

where $f^n \equiv 0$ for $n < N^1$ and $n > N^2$; $\mathbf{G}^n = 0$ are the grid equations at time level n; $\boldsymbol{\Lambda}_f^n$ and $\boldsymbol{\Lambda}_g^n$ are vectors of Lagrange multipliers associated with the flow and grid equations at time level n, respectively; $\mathbf{D}$ is a vector of design variables; and $\mathbf{R}^{\mathrm{in}} = 0$ is the initial condition for the flow equations.

The Lagrangian is differentiated with respect to $\mathbf{D}$, assuming that f^n depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, and $\mathbf{D}$; $\mathbf{R}^{\mathrm{in}}$ depends on $\mathbf{Q}^0$, $\mathbf{X}^0$, and $\mathbf{D}$; $\mathbf{R}^n$ depends on $\mathbf{Q}^n$, $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$; and $\boldsymbol{R}^n_{\mathrm{GCL}}$ depends on $\mathbf{X}^n$, $\mathbf{X}^{n-1}$, and $\mathbf{D}$. Regrouping terms to isolate the coefficients of $\partial \mathbf{Q}^n / \partial \mathbf{D}$ and equating the coefficients to zero yields the final form of the adjoint equations for the flowfield:

$$\begin{aligned} & \frac{1}{\Delta t}(\boldsymbol{V}^n \boldsymbol{\Lambda}_f^n - \boldsymbol{V}^{n+1} \boldsymbol{\Lambda}_f^{n+1}) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \boldsymbol{\Lambda}_f^n + \boldsymbol{R}^{n+1}_{\mathrm{GCL}} \boldsymbol{\Lambda}_f^{n+1} \\ & \quad = -\left[\frac{\partial f^n}{\partial \mathbf{Q}^n} \right]^T, \quad \text{for } 1 \le n \le N \end{aligned} \tag{19}$$

$$\begin{aligned} & -\frac{1}{\Delta t} \boldsymbol{V}^1 \boldsymbol{\Lambda}_f^1 + \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{Q}^0} \right]^T \boldsymbol{\Lambda}_f^0 + \boldsymbol{R}^1_{\mathrm{GCL}} \boldsymbol{\Lambda}_f^1 \\ & \quad = -\left[\frac{\partial f^0}{\partial \mathbf{Q}^0} \right]^T \quad \text{for the initial time level} \end{aligned} \tag{20}$$

where $\boldsymbol{\Lambda}^{N+1} = 0$. Collecting the coefficients of $\partial \mathbf{X}^n / \partial \mathbf{D}$ and equating them to zero leads to similar adjoint equations for the grid. Assuming that the grid operator at time level n, $\mathbf{G}^n$, depends on $\mathbf{X}^n$, $\mathbf{X}^0$, and $\mathbf{D}$, the grid adjoint equations are defined as

$$\begin{aligned} & -\left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^n} \right]^T \boldsymbol{\Lambda}_g^n = \left[\frac{\partial f}{\partial \mathbf{X}^n} \right]^T + \left[\frac{\partial \boldsymbol{V}}{\partial \mathbf{X}^n} \frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} \right]^T \boldsymbol{\Lambda}_f^n \\ & \quad + \sum_{k=0}^{1} \left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\mathrm{GCL}}}{\partial \mathbf{X}^n} \mathbf{Q}^{n+k-1} \right]^T \boldsymbol{\Lambda}_f^{n+k}, \quad \text{for } 1 \le n \le N \end{aligned} \tag{21}$$

$$\begin{aligned} & -\left[\frac{\partial \mathbf{G}^0}{\partial \mathbf{X}^0} \right]^T \boldsymbol{\Lambda}_g^0 = \sum_{n=1}^{N} \left[\frac{\partial \mathbf{G}^n}{\partial \mathbf{X}^0} \right]^T \boldsymbol{\Lambda}_g^n + \left[\frac{\partial f^0}{\partial \mathbf{X}^0} \right]^T + \left[\frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{X}^0} \right]^T \boldsymbol{\Lambda}_f^0 \\ & \quad + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^1_{\mathrm{GCL}}}{\partial \mathbf{X}^0} \mathbf{Q}^0 \right]^T \boldsymbol{\Lambda}_f^1 \end{aligned} \tag{22}$$

The specific form of these equations will be discussed in subsequent sections. With the adjoint coefficients satisfying the flowfield and grid adjoint equations, the sensitivity derivatives are calculated as follows:

$$\begin{aligned} \frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = {} & \sum_{n=1}^{N} \left(\frac{\partial f^n}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_f^n]^T \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + \frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}} \mathbf{Q}^{n-1} \right] + [\boldsymbol{\Lambda}_g^n]^T \frac{\partial \mathbf{G}^n}{\partial \mathbf{D}} \right) \Delta t \\ & + \left(\frac{\partial f^0}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_f^0]^T \frac{\partial \mathbf{R}^{\mathrm{in}}}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}_g^0]^T \frac{\partial \mathbf{G}^0}{\partial \mathbf{D}} \right) \Delta t \end{aligned} \tag{23}$$

Implementation

Flowfield Adjoint Equations

The implementation and solution of Eqs. (19) and (20) are based largely on the steady-state strategies described in [4,15–19]. In this manner, a great deal of software development effort is avoided because the steady and unsteady equations share many similar terms, namely, the details of the spatial discretization. However, some fundamental differences in the implementation must be addressed for time-dependent problems.

Implications of Reverse Time Integration

Although the discrete solution $\mathbf{Q}^n$ for Eq. (3) is determined by marching forward in physical time from $n = 0$ to N, due to the nature of the adjoint equations and their boundary conditions, the solution for $\boldsymbol{\Lambda}_f^n$ must instead be initiated from $n = N$ and proceed backward in physical time. Because Eqs. (19) and (20) involve the linearizations $\partial \mathbf{R}^n / \partial \mathbf{Q}$ and $\partial f^n / \partial \mathbf{Q}$, the flow solution $\mathbf{Q}^n$ at all time levels must be available during the reverse integration.

In practice, the most straightforward approach to meeting this requirement is to store $\mathbf{Q}^n$ to disk for all n during the solution of Eq. (16). In this case, the storage cost is significant, but the primary advantage is ease of implementation. This is the approach used for the current study. For problems in which the mesh is changing in time, the grid point coordinates and associated speeds are also stored. Although these mesh-related values could be recovered by performing the mesh motion in reverse, ease of the full storage implementation has been favored.

Solution Strategy

As described in [20], each solution vector $\mathbf{Q}^n$ is determined through a dual time-stepping procedure. In this approach, a sequence of subiterations is performed within each physical time step. The procedure relies on an approximate linearization of the discrete residual combined with a pseudotime term to achieve a scheme directly analogous to that used in [22] for steady flows. The same subiterative strategy is employed for the time-dependent adjoint equations, following an approach similar to that outlined in [18]. The Jacobian matrix used to relax the adjoint system is constructed once at each time step n based on the value of $\mathbf{Q}^n$ and does not change during the subiterative procedure.

A requirement for performing adjoint solutions is that the iteration scheme be linearly stable. It has been observed in some cases, more often for unsteady problems than for steady ones, that linear stability is not satisfactory. Suggested explanations [19,26–28] vary from physical instabilities to instabilities of the numerical schemes involved. The generalized conjugate residual scheme described in [29] has been used to wrap the multicolor Gauss–Seidel iteration as well as the temporal subiterative procedure. This approach has been found to work well in stabilizing otherwise problematic iterations.

Data Storage

For three-dimensional dynamic grid simulations using a one-equation turbulence model, the reverse time-integration and solution techniques outlined earlier require the storage of 12 floating-point variables per grid point at each time step: six flowfield variables, three mesh coordinates, and three mesh velocities. For large-scale

problems involving many time steps, this strategy can easily result in a storage requirement on the order of terabytes of disk space. Strategies for circumventing storage limitations have been suggested in the literature [9,30,31]; these may be the focus of future investigations once an initial capability has been established.

In the current implementation, each processor is responsible for reading and writing its local solution for the entire time history to a unique file on disk. Because each file may contain several gigabytes of data, requiring several hundred processors to parse sequential-access files at each time step can be very inefficient. For this reason, direct-access files are used so that the file pointer can be immediately placed at the record of interest. It has been found that this approach can decrease the time required for disk input/output (I/O) by as much as two orders of magnitude for large cases. The use of asynchronous file I/O was also examined, although it is not currently being used.

Grid Adjoint and Sensitivity Equations

Depending on the nature of the grid operator $\mathbf{G}$ and the design variables $\mathbf{D}$, the grid adjoint and sensitivity equations may need to be solved at each time level n, once at $n = 0$, or not at all. If solutions at each time step are required, they are performed at the completion of each step of the adjoint solver, rather than subsequently performing additional loops over the entire range of time levels. In this manner, $\mathbf{Q}^n$, $\mathbf{X}^n$, and the mesh velocities are the only vectors that must be stored for all n, whereas $\mathbf{\Lambda}_f^n$ and $\mathbf{\Lambda}_g^n$ may be discarded when no longer needed.

The predominant challenge in the discretization and solution of Eqs. (21–23) is the infrastructure required to simultaneously manage data from several time levels. An inspection of Eqs. (A7–A9) in the Appendix that are higher-order analogs to Eq. (21) shows that, for a given time step n, the solution for $\mathbf{\Lambda}_g^n$ may depend on values of $\mathbf{Q}$ from adjacent time levels both before and subsequent to level n. Values of $\mathbf{\Lambda}_f$ must also be available at time level n as well as later time levels. Moreover, this complexity increases with the temporal order of the scheme.

The summation term in Eq. (21) is ultimately due to the dependency of the mesh speeds on grid coordinates at multiple time levels, according to the BDF scheme being used. Rather than linearizing $\mathbf{R}$ and $\boldsymbol{R}_{\mathrm{GCL}}$ at several time levels with respect to the grid coordinates at the current time level as indicated in the summation, an inverse approach more amenable to the existing implementation of the spatial linearizations is used. The residual at time level n is linearized with respect to the grid coordinates at every time level in the temporal stencil by seeding the linearizations with the appropriate BDF coefficient. The results are then stored temporarily for use in evaluating the summation term at subsequent time levels within the stencil, after which the linearizations are discarded.

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 [32,33] and was first applied to a Navier–Stokes solver in [3]. 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 $i\varepsilon$:

$$\frac{\partial f}{\partial x} = \frac{\mathrm{Im}[f(x + i\varepsilon)]}{\varepsilon} + \mathcal{O}(\varepsilon^2) \tag{24}$$

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 divided differences. Through the use of an automated scripting procedure outlined in [34], 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^{-30}, which highlights the robustness of the complex-variable approach. For each verification test, all equations sets are converged to machine precision for both the complex-variable and adjoint approaches. When used, the elasticity matrix $\boldsymbol{K}$ is assumed to be constant throughout the verification.

Static Grid

Test Case

The first test case is used to verify the implementation for unsteady flows on static grids. For this example, fully turbulent flow over the ONERA M6 wing [35] shown in Fig. 1 is considered. The grid contains 16,391 nodes and 90,892 tetrahedral elements, and 16 processors are used for the simulation. The freestream Mach number is 0.3, the angle of attack is 1 deg, and the Reynolds number is 1×10^6 based on the mean aerodynamic chord (MAC). The simulation is initiated from freestream conditions $\mathbf{Q}^\infty$, which leads to $\mathbf{R}^{\mathrm{in}} \equiv \mathbf{Q}^\infty - \mathbf{Q}^0$. The solution is advanced five physical time steps using a nondimensional Δt of 0.1. Although this coarse spatial resolution, relatively large time step, 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.

Design Variables

For this test, two general classes of design variables are used. The first class of variables 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 useful in verifying the implementation of the flowfield adjoint equation, as the terms in Eq. (23) associated with these parameters are generally trivial to implement or identically zero, and solution of the mesh adjoint equations is not required.

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 all examples in the current study, the grid parameterization approach described in [36] 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 two or more design variables to define more general parameters. In the event that multiple bodies of the same shape are to be designed, 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.

Grid Adjoint Equation

For this case, there is only one grid operator, $\mathbf{G}(\mathbf{X}, \mathbf{D}) \equiv \mathbf{X}_{\mathrm{surf}} - \boldsymbol{K}\mathbf{X}$, which does not depend on time. As a result, the grid adjoint equation can be recast as

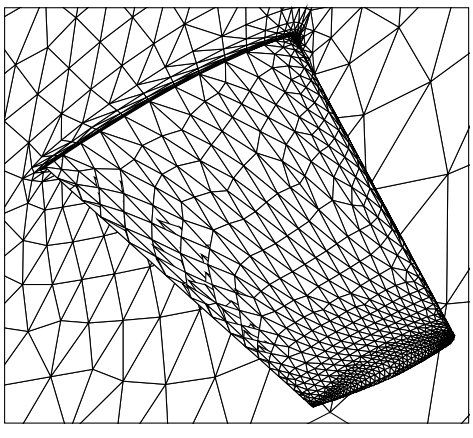

Fig. 1 Surface grid for ONERA M6 wing.

$$\left[\boldsymbol{K}+\frac{\partial \boldsymbol{K}}{\partial \mathbf{X}}\mathbf{X}\right]^T\boldsymbol{\Lambda}_g=\sum_{n=1}^{N}\left\{\left[\frac{\partial f^n}{\partial \mathbf{X}}\right]^T+\left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}_f^n+\sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}}\right]^T\boldsymbol{\Lambda}_f^{n+k}\right\}+\left[\frac{\partial f^0}{\partial \mathbf{X}}\right]^T+\left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}}\right]^T\boldsymbol{\Lambda}_f^1 \tag{25}$$

and the sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}\right)\Delta t+[\boldsymbol{\Lambda}_g]^T\left[\frac{\partial \mathbf{X}_{\mathrm{surf}}}{\partial \mathbf{D}}-\frac{\partial \boldsymbol{K}}{\partial \mathbf{D}}\mathbf{X}\right]\Delta t \tag{26}$$

Computational Results

The test has been performed using the BDF1 scheme and all other time-integration schemes described in the Appendix, and results are listed in Table 1. Sensitivity derivatives of the lift coefficient at the final time step with respect to the angle of attack and a camber variable located at the midspan of the wing are shown. The results for the adjoint implementation exhibit excellent agreement with the complex-variable approach, differing at most in the fifteenth digit.

Rigidly Moving Grid

Test Case

The next test case is used to verify the implementation for rigidly moving meshes. For this case, the grid and freestream conditions and computational environment are identical to those described for the preceding test; however, the mesh is now subjected to an oscillatory pitch-plunge motion based on the rigid mesh transform approach outlined earlier. The nondimensional pitching and plunging reduced frequencies are 0.5 and 0.1, respectively. The pitching amplitude is 5 deg and takes place about a vector normal to the symmetry plane located 0.47 MAC from the wing root leading edge. The amplitude of the plunging motion is 0.38 MAC. The baseline wing position at $t=0$ is as shown in Fig. 1. As in the preceding test, the simulation is initiated from freestream conditions $\mathbf{R}^{\mathrm{in}}\equiv\mathbf{Q}^\infty-\mathbf{Q}^0$ and is advanced five physical time steps using a nondimensional Δt of 0.1.

Design Variables

The design variables for the current test include those described earlier for the static grid example, as well as a third class of parameters governing the rigid motion procedure described earlier. These include translation and rotation frequencies, amplitudes, and directional vectors, as well as centers of rotation.

Grid Adjoint Equation

For this test case, the following grid operators are used: at the zeroth time level, the grid is either unchanged or governed by the elasticity equations $\mathbf{G}^0(\mathbf{X}^0,\mathbf{D})\equiv\mathbf{X}^0_{\mathrm{surf}}-\mathbf{K}^0\mathbf{X}^0$; grids at other time levels are governed by the rigid motion equation $\mathbf{G}^n(\mathbf{X}^n,\mathbf{D})\equiv\boldsymbol{R}^n\mathbf{X}^0+\boldsymbol{\tau}^n-\mathbf{X}^n$.

The grid adjoint equations are given by

$$\boldsymbol{\Lambda}_g^n=\left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T+\left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}_f^n+\sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n}+\frac{\partial \boldsymbol{R}^{n+k}_{\mathrm{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\boldsymbol{\Lambda}_f^{n+k},\quad \text{for } 1\le n\le N \tag{27}$$

Under the assumption that the shape does not change ($\mathbf{X}^0$ is constant), the sensitivity derivative is given by

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}+\frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right]+[\boldsymbol{\Lambda}_g^n]^T\frac{\partial(\boldsymbol{R}^n\mathbf{X}^0+\boldsymbol{\tau}^n)}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}\right)\Delta t \tag{28}$$

The formulation that would allow shape design is the following:

$$\left[\boldsymbol{K}^0+\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\boldsymbol{\Lambda}_g^0=\left(\sum_{n=1}^{N}[\boldsymbol{R}^n]^T\boldsymbol{\Lambda}_g^n\right)+\left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T+\left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\boldsymbol{\Lambda}_f^0+\left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0}+\frac{\partial \boldsymbol{R}^1_{\mathrm{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^\infty\right]^T\boldsymbol{\Lambda}_f^1 \tag{29}$$

and the corresponding sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}=\sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^n]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}+\frac{\partial \boldsymbol{R}^n_{\mathrm{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right]+[\boldsymbol{\Lambda}_g^n]^T\frac{\partial \boldsymbol{\tau}^n}{\partial \mathbf{D}}\right)\Delta t+\left(\frac{\partial f^0}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_f^0]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}}+[\boldsymbol{\Lambda}_g^0]^T\left[\frac{\partial \mathbf{X}^0_{\mathrm{surf}}}{\partial \mathbf{D}}-\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^0\right]\right)\Delta t \tag{30}$$

Computational Results

Results for the derivatives of the lift coefficient at the final time step are shown in Table 2 for the current case. In addition to the angle of attack and camber variables, derivatives with respect to the rigid motion pitching frequency are also shown. The agreement with the complex-variable formulation is excellent for each of the time-integration schemes considered.

Table 1 Results for static grid test case where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	$\mathrm{BDF2}_{\mathrm{opt}}$
Angle of attack	A: 0.004249541855867	A: 0.003734353591935	A: 0.003687377975335	A: 0.003708754474661
	C: 0.004249541855867	C: 0.003734353591935	C: 0.003687377975335	C: 0.003708754474661
Camber	A: 0.010713047647152	A: 0.013701437304586	A: 0.014574974114575	A: 0.014145698047604
	C: 0.010713047647155	C: 0.013701437304586	C: 0.014574974114577	C: 0.014145698047602

Table 2 Results for rigidly moving grid where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	$\mathrm{BDF2}_{\mathrm{opt}}$
Angle of attack	A: 0.004713138571667	A: 0.004293218571759	A: 0.004245785984455	A: 0.004267302756747
	C: 0.004713138571667	C: 0.004293218571759	C: 0.004245785984455	C: 0.004267302756681
Pitching frequency	A: −0.403740396501207	A: −0.527819225717431	A: −0.529833595955533	A: −0.528894917963836
	C: −0.403740396501207	C: −0.527819225717432	C: −0.529833595955533	C: −0.528894917963837
Camber	A: 0.011630821689945	A: 0.013925365539211	A: 0.014291228334440	A: 0.014071544549783
	C: 0.011630821689944	C: 0.013925365539206	C: 0.014291228334428	C: 0.014071544549783

Table 3 Results for morphing grid where A denotes adjoint result and C denotes complex-variable result

Design variable	BDF1	BDF2	BDF3	BDF2_{opt}
Angle of attack	A: 0.004713528355526	A: 0.004298221887378	A: 0.004250753632738	A: 0.004272205860974
	C: 0.004713528355526	C: 0.004298221887378	C: 0.004250753632738	C: 0.004272205860974
Pitching frequency	A: −0.403961428430834	A: −0.528263525075847	A: −0.530205775809711	A: −0.529295291075346
	C: −0.403961428430834	C: −0.528263525075847	C: −0.530205775809710	C: −0.529295291075346
Camber	A: 0.011680362720549	A: 0.013922237526691	A: 0.014268675858452	A: 0.014055458873064
	C: 0.011680362720548	C: 0.013922237526686	C: 0.014268675858435	C: 0.014055458873058

Morphing Grid

Test Case

To evaluate the accuracy of the implementation for morphing grids, the test case used for rigid motion described earlier is repeated with slight modifications. For the current test, the surface grid of the wing is moved using rigid motion, whereas the interior of the mesh is determined using the elasticity relation given by Eq. (9). All other input parameters remain unchanged.

Design Variables

The current test case uses the same design variables as the rigid motion test case described earlier.

Grid Adjoint Equation

At all time levels, the grids are governed by the elasticity equations $\mathbf{G}^n(\mathbf{X}^n, \mathbf{D}) \equiv \mathbf{X}^n_{\text{surf}} - \boldsymbol{K}^0\mathbf{X}^n$, and the surface coordinates are governed by the rigid motion equation $\mathbf{X}^n_{\text{surf}} = \boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n$.

The grid adjoint equations are given by

$$[\boldsymbol{K}^0]^T\boldsymbol{\Lambda}^n_g = \left[\frac{\partial f}{\partial \mathbf{X}^n}\right]^T + \left[\frac{\partial \boldsymbol{V}^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t}\right]^T\boldsymbol{\Lambda}^n_f + \sum_{k=0}^{1}\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\text{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\boldsymbol{\Lambda}^{n+k}_f, \quad \text{for } 1 \le n \le N \tag{31}$$

$$\left[\boldsymbol{K}^0 + \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\boldsymbol{\Lambda}^0_g = \left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T + \left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\boldsymbol{\Lambda}^0_f + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^1_{\text{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^\infty\right]^T\boldsymbol{\Lambda}^1_f - \sum_{n=1}^{N}\left[\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^n\right]^T\boldsymbol{\Lambda}^n_g \tag{32}$$

The sensitivity derivative is

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = \sum_{n=1}^{N}\left(\frac{\partial f^n}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^n_f]^T\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} + \frac{\partial \boldsymbol{R}^n_{\text{GCL}}}{\partial \mathbf{D}}\mathbf{Q}^{n-1}\right] + [\boldsymbol{\Lambda}^n_g]^T\left[\frac{\partial(\boldsymbol{R}^n\mathbf{X}^0_{\text{surf}} + \boldsymbol{\tau}^n)}{\partial \mathbf{D}} - \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^n\right]\right)\Delta t + \left(\frac{\partial f^0}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^0_f]^T\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{D}} + [\boldsymbol{\Lambda}^0_g]^T\left[\frac{\partial \mathbf{X}^0_{\text{surf}}}{\partial \mathbf{D}} - \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{D}}\mathbf{X}^0\right]\right)\Delta t \tag{33}$$

Two observations can be made. First, note that in the absence of any surface motion, that is, $\boldsymbol{R}^n$ is the identity matrix and $\boldsymbol{\tau}^n = 0$, the morphing grid formulation is equivalent to the static grid formulation. Also, with a constant transformation matrix T applied to all computational boundaries, the morphing and rigidly moving grid formulations are equivalent.

Computational Results

The results for the current test case are shown in Table 3. Derivatives of the lift coefficient at the final time step with respect to each of the design variables exhibit excellent agreement for the adjoint implementation and complex-variable formulation.

Large-Scale Design Cases

Two large-scale design optimization examples are presented. Although the grid motion in both cases is prescribed, a more realistic treatment would involve the use of additional coupled computational models such as 6 degrees of freedom or structural simulations. Although such capabilities are available for use with the flow solver [20], their effects have not been accounted for in the derivation and implementation of the adjoint equations. This important development is relegated to future work.

Both of the example cases shown next have been performed using 128 dual-socket quad-core nodes with 3.0 GHz Intel Xeon processors in a fully dense fashion for a total of 1024 computational cores. This environment has been chosen to maximize computational efficiency for the chosen test problems; numerical experiments have shown that the solvers used in the current study scale well in this range for the grid sizes selected.

The computational grid sizes and time steps for the examples presented here have been chosen merely to demonstrate optimization capability for typical problems using immediately available resources. Spatial and/or temporal refinement could be readily performed if desired. Although the formulation places no restrictions on initial conditions, all solutions are started from freestream conditions. The grids have been generated using the method in [37], and the optimizations have been performed using a trust region method from the package described in [38].

Tilt-Rotor Configuration

The first large-scale example is a three-bladed tilt-rotor configuration similar to that used by the V-22 aircraft and is based on the tilt-rotor aeroacoustics model (TRAM) geometry described in [39,40]. The grid used for this computation is designed for a blade collective setting of $\Theta = 14$ deg and consists of 5,048,727 nodes and 29,802,252 tetrahedral elements. The rotational speed of the rotor is held constant at a value corresponding to a tip Mach number of 0.62 in a hover condition. The Reynolds number is 2.1×10^6 based on the blade tip chord. The physical time step is chosen to

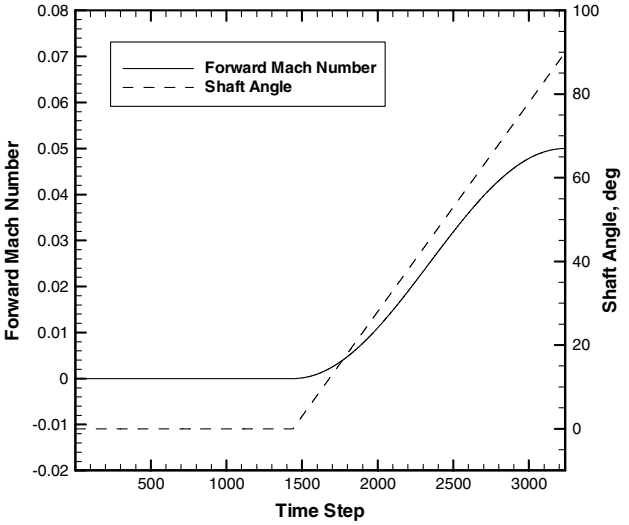

Fig. 2 Forward Mach number and shaft angle schedule for TRAM rotor simulation.

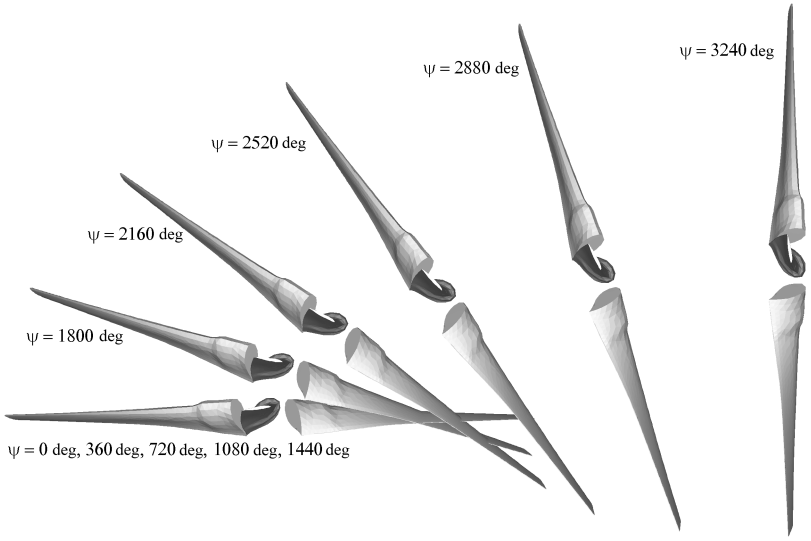

Fig. 3 View of TRAM rotor motion.

correspond to 1 deg of rotor azimuth, for a total of 360 time steps per revolution. The $BDF2_{opt}$ formulation outlined in [41] is used with 10 subiterations per time step.

For this test, the prescribed rigid mesh motion consists of four initial revolutions of the geometry designed to reach a quasi-steady hover condition, followed by five additional revolutions during which a 90 deg constant-rate pitch-up maneuver into a forward-flight mode is performed. A more realistic pitch-up scenario might consist of many more revolutions; however, the prescribed motion was chosen to keep the cost of the computation affordable given the current resources. During the pitch-up phase of the motion, an assumed forward-flight velocity profile based on a simple sine function is imposed through the mesh speed terms. The schedule for the shaft angle and forward-flight velocity is shown in Fig. 2, in which the shaft angle is defined to be 0 deg in the hover condition and 90 deg in forward flight. The resulting motion is shown in Fig. 3, in which a snapshot of the rotor is shown every 360 deg during the course of the motion. An isosurface of the second invariant of the velocity-gradient tensor, also known as the Q criterion from [42], at the time step corresponding to $\Psi = 1440$ deg is shown in Fig. 4. The tip vortex system is maintained for 2–3 revolutions of the rotor.

The objective function for the current test case is to maximize the rotor thrust coefficient over the time interval corresponding to the pitch-up maneuver, $1441 \text{ deg} \leq \Psi \leq 3240 \text{ deg}$:

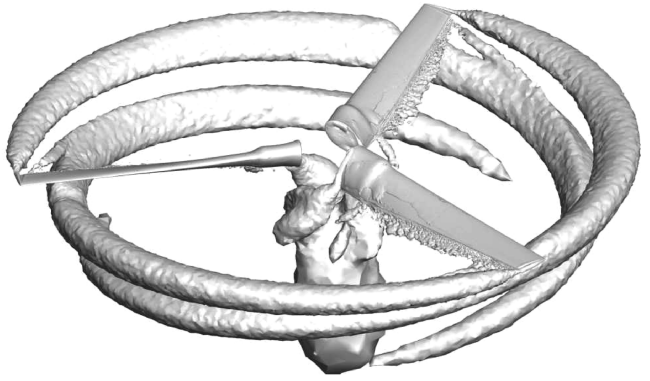

Fig. 4 Isosurface of Q criterion for TRAM rotor at $\Psi = 1440$ deg.

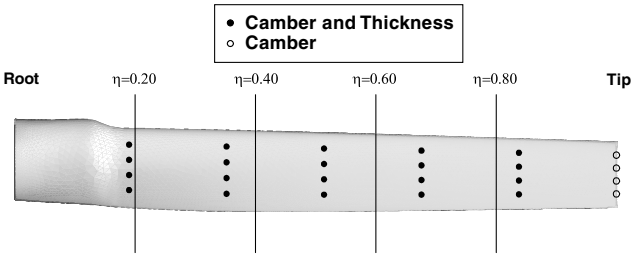

Fig. 6 Spanwise blade and design variable locations for TRAM rotor.

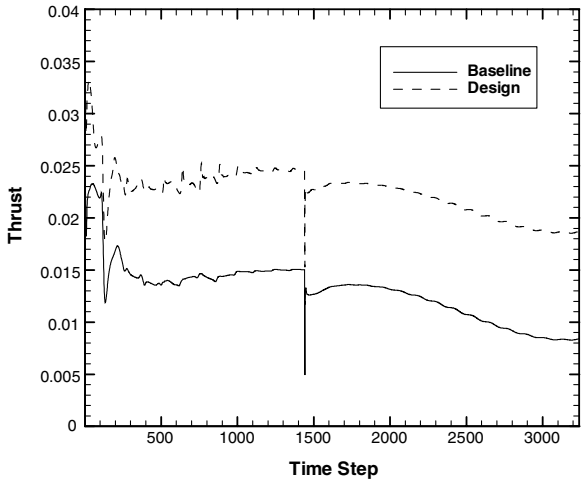

Fig. 5 Thrust for TRAM rotor before and after design optimization.

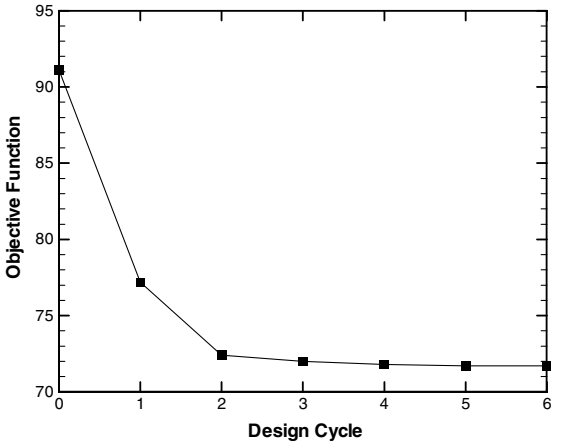

Fig. 7 Objective function history for TRAM rotor.

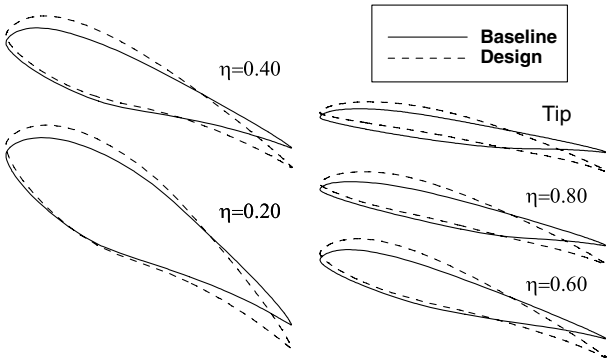

Fig. 8 Spanwise blade cross sections before and after optimization of TRAM rotor.

Fig. 9 Modified F-15 with engine duct geometry.

$$f = \sum_{n=1441}^{3240} (C_T^n - 0.1)^2 \Delta t \tag{34}$$

Here, the target thrust coefficient value of 0.1 has been chosen to sufficiently exceed the baseline thrust profile shown as the solid line in Fig. 5. After the first four rotor revolutions, the thrust coefficient has reached a quasi-steady value of approximately 0.015, which is in good agreement with experimental data given in [39,40]. The thrust coefficient shows a discontinuous behavior at the impulsive start of the pitch-up motion ($n = 1441$) and gradually decreases to a lower constant value in the forward-flight condition. A subtle 3/rev oscillation in the thrust coefficient during the pitch-up maneuver can also be seen.

The surface grid has been parameterized as described in [43]. This approach yields a set of 44 active design variables describing the

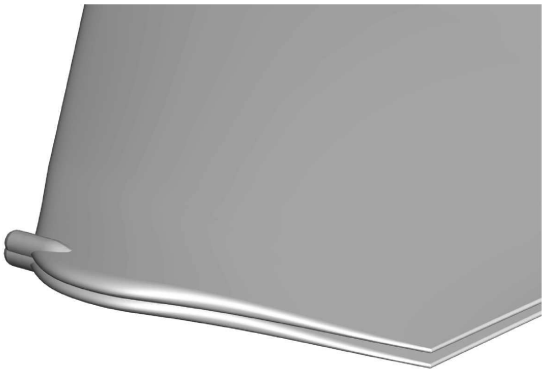

Fig. 10 Range of prescribed motion for modified F-15 wing tip.

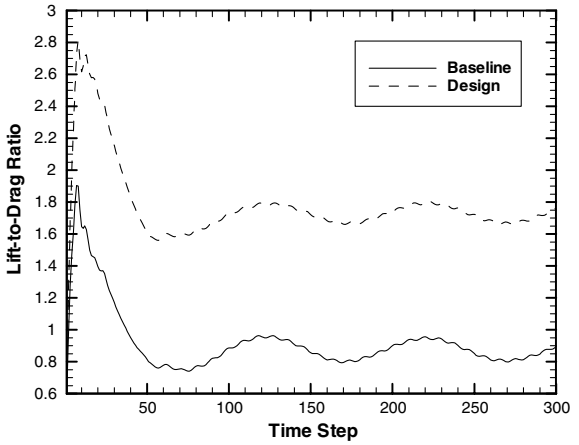

Fig. 11 Lift-to-drag ratio for modified F-15 before and after design optimization.

thickness and camber of the blade geometry as shown in Fig. 6; thinning of the blade is not allowed. Additional bound constraints have been specified based on previous experience in avoiding nonphysical geometries. In addition, a single twist variable is used to modify the blade collective setting during the design.

The convergence history for six design cycles is shown in Fig. 7. The optimizer quickly reduces the value of the objective function over the first two design cycles, after which further improvements are minimal. Closer inspection of the design variables indicates that the majority of values have reached their bound constraints, preventing any further reduction in the objective function. The final thrust coefficient profile is included as the dashed line in Fig. 5. Cross sections of the baseline blade geometry are compared with the optimized geometry in Fig. 8. The optimization has increased the camber of the blade across the span, as well as the blade collective setting.

The cost of each solution to the unsteady flow and adjoint equations for the current example is approximately 3.5 and 10.5 wall-clock hours, respectively; however, due to frequent file I/O, this estimate varies with file system load. The optimization procedure requires 12 calls to the flow solver and 6 calls to the adjoint solver, for a total runtime of approximately 4.5 days of wall-clock time or 110,000 h of CPU time. The disk storage required for one complete flow solution is approximately 1.5 terabytes.

Fighter Jet with Simulated Aeroelastic Effects

The second example uses a deforming grid approach to simulate aeroelastic motion of the modified F-15 fighter jet configuration known as NASA research aircraft 837, shown in Fig. 9.[¶] The computational model assumes half-plane symmetry in the spanwise direction. The grid consists of 4,715,852 nodes and 27,344,343 tetrahedral elements and includes detailed features of the external airframe as well as the internal ducting upstream of the engine fan face and the plenum/nozzle combination downstream of the turbine. For the current test, the freestream Mach number is 0.90, the angle of attack is 0 deg, and the Reynolds number based on the MAC is 1×10^6. The static pressure ratio at the engine fan face is set to 0.9, and the total pressure ratio at the plenum face is ramped linearly from 1.0 to its final value of 5.0 over the first 50 time steps.

The prescribed grid motion consists of 5 Hz 0.3 deg oscillatory rotations of the canard, wing, and tail surfaces about their root chord lines, with the wing oscillations 180 deg out of phase with the canard and tail motion. In addition, the main wing is also subjected to a 5 Hz oscillatory twisting motion for which the amplitude decays linearly from 0.5 deg at the wing tip to 0 deg at the wing root and takes place about the quarter-chord line. This composite motion

[¶]Data available online at http://www.nasa.gov/centers/dryden/aircraft/F-15B-837/index.html [retrieved 4 January 2010].

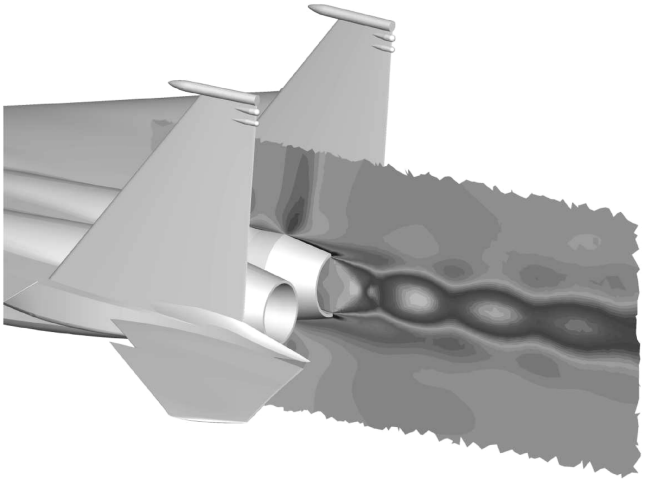

Fig. 12 Cross-section of engine plume contours for modified F-15.

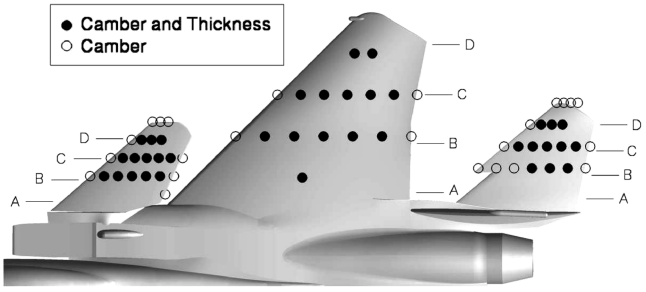

Fig. 13 Spanwise and design variable locations for modified F-15.

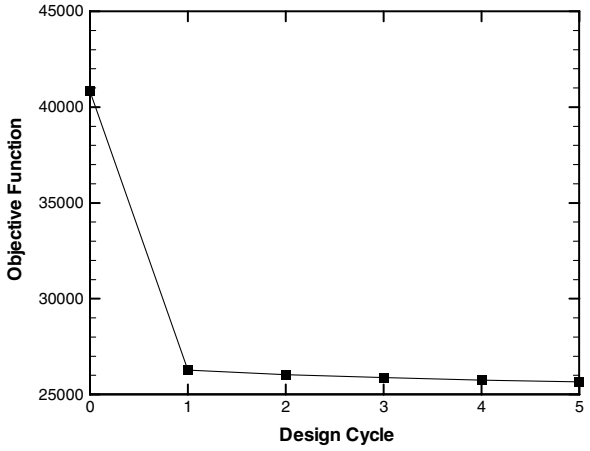

Fig. 14 Objective function history for modified F-15.

results in a maximum wing tip deflection of approximately 1.3% MAC, as shown in Fig. 10. The $BDF2_{opt}$ scheme is used with 10 subiterations and a physical time step corresponding to 100 steps per cycle of grid motion.

The unsteady lift-to-drag ratio (L/D) for the baseline configuration undergoing the specified motion for 300 time steps is shown as the solid line in Fig. 11. The L/D behavior begins to exhibit a periodic response after approximately 100 time steps. The high-frequency oscillations in the profile are believed to be due to a small unsteadiness in the engine plume shown in Fig. 12; this behavior is also present when the mesh is held fixed.

The objective function for the current test case is to maximize L/D for the interval $201 \leq n \leq 300$:

$$f = \sum_{n=201}^{300} [(L/D)^n - 5.0]^2 \Delta t \tag{35}$$

where the target L/D value of 5.0 has been chosen to provide sufficient room for optimization over the baseline profile. The surface grids for the canard, wing, and tail have been parameterized as shown in Fig. 13, resulting in a set of 98 active design variables describing the thickness and camber of each surface. Thinning of the geometry is not permitted, and other bound constraints are chosen to avoid nonphysical geometries.

Convergence of the objective function is shown in Fig. 14. A large reduction in the function is obtained after a single design cycle, after which further improvements are minimal due to many of the design variables having reached their bound constraints. The final L/D profile is included as the dashed line in Fig. 11. The resulting shape changes at various spanwise stations on the canard, wing, and tail are shown in Fig. 15, in which the vertical scale has been exaggerated for clarity. The design procedure has increased the thickness of the wing and canard, as well as the camber across all three elements. Closer inspection shows that the trailing edges of each surface have also been deflected in a downward fashion.

The wall-clock times required for single flow and adjoint solutions for the current problem are approximately 1 and 1.5 h, respectively. For the five design cycles shown in Fig. 14, the optimizer requires 10 flow solutions and 5 adjoint solutions, or a total wall-clock time of approximately 18 h or 18,400 h of CPU time. The disk space necessary to store a single unsteady flow solution is 136 gigabytes.

Conclusions

A discrete adjoint-based methodology for optimization of unsteady flows governed by the three-dimensional Reynolds averaged Navier–Stokes equations on dynamic unstructured grids has been formulated and implemented. The methodology accounts for mesh motion based on both rigid movement as well as deforming grids. The accuracy of the implementation has been verified using comparisons with an independent approach based on the use of complex variables. The methodology has been successfully used in a massively parallel environment to perform two large-scale design optimization examples: one for a tilt rotor in a pitch-up maneuver into a forward-flight regime and another for a fighter jet with simulated aeroelastic effects.

Although the approach outlined in the current study represents significant progress toward the goal of performing routine optimization of unsteady turbulent flows, a number of research areas remain to be explored. The extension of the present formulation to overset grid topologies is ongoing and will allow for the treatment of multiple

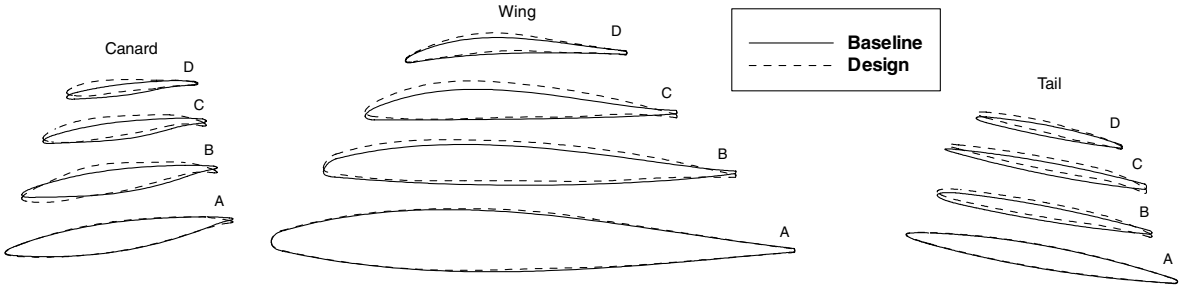

Fig. 15 Canard, wing, and tail cross sections before and after optimization of modified F-15.

bodies undergoing large relative motion. Methods aimed at reducing the storage costs associated with the flow solution have the potential to drastically reduce disk requirements. Techniques based on variable or adaptive time steps as well as alternate time-integration schemes should be examined. The effects of related computational disciplines such as 6 degrees of freedom and structural models should also be properly accounted for. Finally, the use of the unsteady flowfield adjoint solution holds tremendous potential for performing mathematically rigorous mesh adaptation to specified error bounds.

Appendix A: Adjoint Equations for Higher-Order Backward-Difference-Formula Schemes

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

$$\frac{\partial s}{\partial t} = \frac{1}{\Delta t}[as^n + bs^{n-1} + cs^{n-2} + ds^{n-3}] \tag{A1}$$

where n is a time level, and the coefficients are given in Table A1. The coefficients listed for the BDF2_{opt} scheme are a linear combination of the BDF2 and BDF3 coefficients taken from [41]. The resulting scheme is second-order-accurate but has a leading truncation error term less than that of the BDF2 scheme. Although usually found to be stable in practice, stability of the BDF2_{opt} and third-order BDF3 scheme are not guaranteed. Discrete conservation laws are defined as

$$aV^n\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t} + cV^{n-2}\frac{\mathbf{Q}^{n-2}-\mathbf{Q}^{n-1}}{\Delta t} + dV^{n-3}\frac{\mathbf{Q}^{n-3}-\mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n + \boldsymbol{R}^n_{\text{GCL}}\mathbf{Q}^{n-1} = 0 \tag{A2}$$

Because the morphing grid formulation includes static meshes and rigid motion as special cases, the derivation is provided only for this formulation. Taking into account that $\mathbf{R}^n$ and $\boldsymbol{R}^n_{\text{GCL}}$ are dependent on $\mathbf{X}^{n-2}$ and $\mathbf{X}^{n-3}$, the procedure applied to the BDF1 scheme may also be used to derive the following adjoint equations for the flowfield:

$$\frac{a}{\Delta t}(V^n\boldsymbol{\Lambda}_f^n - V^{n+1}\boldsymbol{\Lambda}_f^{n+1}) + \frac{c}{\Delta t}(V^n\boldsymbol{\Lambda}_f^{n+2} - V^{n-1}\boldsymbol{\Lambda}_f^{n+1}) + \frac{d}{\Delta t}(V^n\boldsymbol{\Lambda}_f^{n+3} - V^{n-2}\boldsymbol{\Lambda}_f^{n+1}) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n}\right]^T\boldsymbol{\Lambda}_f^n + \boldsymbol{R}^{n+1}_{\text{GCL}}\boldsymbol{\Lambda}_f^{n+1} = -\left[\frac{\partial f^n}{\partial \mathbf{Q}^n}\right]^T, \quad \text{for } 3 \le n \le N \tag{A3}$$

where $\Lambda^{N+1} = \Lambda^{N+2} = \Lambda^{N+3} = 0$; for $n = 2$:

$$\frac{1}{\Delta t}\left(\frac{3}{2}V^2\boldsymbol{\Lambda}_f^2 - aV^3\boldsymbol{\Lambda}_f^3\right) + \frac{c}{\Delta t}(V^2\boldsymbol{\Lambda}_f^4 - V^1\boldsymbol{\Lambda}_f^3) + \frac{d}{\Delta t}(V^2\boldsymbol{\Lambda}_f^5 - V^{\text{in}}\boldsymbol{\Lambda}_f^3) + \left[\frac{\partial \mathbf{R}^2}{\partial \mathbf{Q}^2}\right]^T\boldsymbol{\Lambda}_f^2 + \boldsymbol{R}^3_{\text{GCL}}\boldsymbol{\Lambda}_f^3 = -\left[\frac{\partial f^2}{\partial \mathbf{Q}^2}\right]^T \tag{A4}$$

for $n = 1$:

$$\frac{1}{\Delta t}\left(V^1\boldsymbol{\Lambda}_f^1 - \frac{3}{2}V^2\boldsymbol{\Lambda}_f^2\right) + \frac{1}{\Delta t}\left(cV^1\boldsymbol{\Lambda}_f^3 - \frac{1}{2}V^0\boldsymbol{\Lambda}_f^2\right) + \frac{d}{\Delta t}V^1\boldsymbol{\Lambda}_f^4 + \left[\frac{\partial \mathbf{R}^1}{\partial \mathbf{Q}^1}\right]^T\boldsymbol{\Lambda}_f^1 + \boldsymbol{R}^2_{\text{GCL}}\boldsymbol{\Lambda}_f^2 = -\left[\frac{\partial f^1}{\partial \mathbf{Q}^1}\right]^T \tag{A5}$$

and for the initial conditions:

Table A1 Coefficients for higher-order BDF schemes

Scheme	a	b	c	d
BDF2	3/2	−2	1/2	0
BDF3	11/6	−3	3/2	−1/3
BDF2_{opt}	5.08/3	−2.58	1.08	−0.58/3

$$-\frac{1}{\Delta t}V^1\boldsymbol{\Lambda}_f^1 + \frac{1}{2\Delta t}V^0\boldsymbol{\Lambda}_f^2 + \frac{d}{\Delta t}V^0\boldsymbol{\Lambda}_f^3 + \left[\frac{\partial \mathbf{R}^{\text{in}}}{\partial \mathbf{Q}^0}\right]^T\boldsymbol{\Lambda}_f^0 + \boldsymbol{R}^1_{\text{GCL}}\boldsymbol{\Lambda}_f^1 = -\left[\frac{\partial f^0}{\partial \mathbf{Q}^0}\right]^T \tag{A6}$$

The corresponding mesh adjoint equations are obtained as follows. Assuming $\mathbf{R}^{N+1} = \mathbf{R}^{N+2} = \mathbf{R}^{N+3} = 0$ and $\boldsymbol{R}^{N+1}_{\text{GCL}} = \boldsymbol{R}^{N+2}_{\text{GCL}} = \boldsymbol{R}^{N+3}_{\text{GCL}} = 0$:

$$[\boldsymbol{K}^0]^T\boldsymbol{\Lambda}_g^n = \left[\frac{\partial f^n}{\partial \mathbf{X}^n}\right]^T + \left[a\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n-1}}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^n + c\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n+1}}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^{n+2} + d\left(\frac{\partial V^n}{\partial \mathbf{X}^n}\frac{\mathbf{Q}^n-\mathbf{Q}^{n+2}}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^{n+3}\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{n+k}}{\partial \mathbf{X}^n} + \frac{\partial \boldsymbol{R}^{n+k}_{\text{GCL}}}{\partial \mathbf{X}^n}\mathbf{Q}^{n+k-1}\right]^T\boldsymbol{\Lambda}_f^{n+k}\right\}, \quad \text{for } 3 \le n \le N \tag{A7}$$

$$[\boldsymbol{K}^0]^T\boldsymbol{\Lambda}_g^2 = \left[\frac{\partial f^2}{\partial \mathbf{X}^2}\right]^T + \left[\frac{3}{2}\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^1}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^2 + c\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^3}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^4 + d\left(\frac{\partial V^2}{\partial \mathbf{X}^2}\frac{\mathbf{Q}^2-\mathbf{Q}^4}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^5\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{2+k}}{\partial \mathbf{X}^2} + \frac{\partial \boldsymbol{R}^{2+k}_{\text{GCL}}}{\partial \mathbf{X}^2}\mathbf{Q}^{1+k}\right]^T\boldsymbol{\Lambda}_f^{2+k}\right\}, \quad \text{for } n = 2 \tag{A8}$$

$$[\boldsymbol{K}^0]^T\boldsymbol{\Lambda}_g^1 = \left[\frac{\partial f^1}{\partial \mathbf{X}^1}\right]^T + \left[\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^0}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^1 + c\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^2}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^3 + d\left(\frac{\partial V^1}{\partial \mathbf{X}^1}\frac{\mathbf{Q}^1-\mathbf{Q}^3}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^4\right] + \sum_{k=0}^{3}\left\{\left[\frac{\partial \mathbf{R}^{1+k}}{\partial \mathbf{X}^1} + \frac{\partial \boldsymbol{R}^{1+k}_{\text{GCL}}}{\partial \mathbf{X}^1}\mathbf{Q}^k\right]^T\boldsymbol{\Lambda}_f^{1+k}\right\}, \quad \text{for } n = 1 \tag{A9}$$

and for the initial conditions, $\mathbf{R}^{\text{in}} \equiv \mathbf{Q}^\infty - \mathbf{Q}^0$:

$$\left[\boldsymbol{K}^0 + \frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^0\right]^T\boldsymbol{\Lambda}_g^0 = \left[\frac{\partial f^0}{\partial \mathbf{X}^0}\right]^T + \left[\frac{1}{2}\left(\frac{\partial V^0}{\partial \mathbf{X}^0}\frac{\mathbf{Q}^0-\mathbf{Q}^1}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^2 + d\left(\frac{\partial V^0}{\partial \mathbf{X}^0}\frac{\mathbf{Q}^0-\mathbf{Q}^2}{\Delta t}\right)^T\boldsymbol{\Lambda}_f^3\right] + \left[\frac{\partial \mathbf{Q}^\infty}{\partial \mathbf{X}^0}\right]^T\Lambda_f^0 + \sum_{k=1}^{3}\left\{\left[\frac{\partial \mathbf{R}^k}{\partial \mathbf{X}^0} + \frac{\partial \boldsymbol{R}^k_{\text{GCL}}}{\partial \mathbf{X}^0}\mathbf{Q}^{k-1}\right]^T\boldsymbol{\Lambda}_f^k\right\} - \sum_{n=1}^{N}\left[\frac{\partial \boldsymbol{K}^0}{\partial \mathbf{X}^0}\mathbf{X}^n\right]^T\boldsymbol{\Lambda}_g^n \tag{A10}$$

The sensitivity derivative for the higher-order BDF schemes is evaluated using Eq. (23).

Acknowledgments

The authors wish to thank Robert Biedron of NASA Langley Research Center for many useful discussions pertaining to the current work. Geometric parameterizations provided by Bill Jones of NASA Langley Research Center are also appreciated. Jan-Renee Carlson of NASA Langley Research Center is acknowledged for her assistance with plenum geometry modifications and boundary conditions for the fighter jet example. The second and third authors acknowledge the support from NASA under grant NNL07AA23C.

References

[1] Vanderplaats, G. N., Hicks, R. N., and Murman, E. M., "Application of Numerical Optimization Techniques to Airfoil Design," *NASA Conference on Aerodynamic Analysis Requiring Advanced Computers*, NASA SP-347, Part 2, March 1975.

[2] Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids*

Engineering, Vol. 113, 1991, pp. 681–688.
doi:10.1115/1.2926534

[3] Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen, E. J., "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

[4] 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

[5] Newman, J. C. III, Taylor, A. C. III, Barnwell, R. W., Newman, P. A., and Hou, G. J.-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

[6] Fidkowski, K. J., and Darmofal, D. L., "Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results," AIAA Paper 2009-1303, 2009.

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

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

[9] Gunzburger, M. D., *Perspectives in Flow Control and Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

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

[11] Mani, K., and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Journal*, Vol. 46, No. 6, 2008, pp. 1351–1364.
doi:10.2514/1.29924

[12] 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

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

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

[15] Nielsen, E. J., "Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier–Stokes Equations and a Discrete Adjoint Formulation," Ph.D. Dissertation, Dept. of Aerospace and Ocean Engineering, Virginia Polytechnic Inst. and State Univ., Dec. 1998.

[16] Nielsen, E. J., and Anderson, W. K., "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

[17] 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

[18] 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*, Vol. 33, No. 9, 2004, pp. 1131–1155.
doi:10.1016/j.compfluid.2003.09.005

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

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

[21] Thomas, P. D., and Lombard, C. K., "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

[22] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.
doi:10.1016/0045-7930(94)90023-X

[23] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.
doi:10.1016/0021-9991(81)90128-5

[24] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 1992-439, 1992.

[25] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal 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

[26] Elliott, J., "Aerodynamic Optimization Based on the Euler and Navier–Stokes Equations Using Unstructured Grids," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, June 1998.

[27] Campobasso, M., and Giles, M., "Effects of Flow Instabilities on the Linear Analysis of Turbomachinery Aeroelasticity," *Journal of Propulsion and Power*, Vol. 19, No. 2, 2003, pp. 250–259.
doi:10.2514/2.6106

[28] Krakos, J., and Darmofal, D. L., "Effect of Small-Scale Unsteadiness on Adjoint-Based Output Sensitivity," AIAA Paper 2009-4274, 2009.

[29] Eisenstat, S. C., Elman, H. C., and Schultz, M., "Variational Iterative Methods for Nonsymmetric Systems of Linear Equations," *SIAM Journal on Numerical Analysis*, Vol. 20, No. 2, 1983, pp. 345–357.
doi:10.1137/0720023

[30] Hinze, M., Walther, A., and Sternberg, J., "Discrete Approximation Schemes for Reduced Gradients and Reduced Hessians in Navier–Stokes Control Utilizing an Optimal Memory-Reduced Procedure for Calculating Adjoints," *Optimal Control Applications and Methods*, Vol. 27, No. 1, 2006, pp. 19–40.
doi:10.1002/oca.771

[31] Yamaleev, N., Diskin, B., and Nielsen, E., "Local-in-Time Adjoint-Based Method for Design Optimization of Unsteady Compressible Flows," AIAA Paper 2009-1169, January 2009.

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

[33] Lyness, J. N., and Moler, C. B., "Numerical Differentiation of Analytic Functions," *SIAM Journal on Numerical Analysis*, Vol. 4, 1967, pp. 202–210.
doi:10.1137/0704019

[34] 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.

[35] Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," *Experimental Database for Computer Program Assessment*, AGARD AR-138, May 1979, pp. B1-1–B1-44.

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

[37] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing Front Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.
doi:10.2514/3.13019

[38] Kaufman, L., and Gay, D., "PORT Library: Optimization and Mathematical Programming—User's Manual," Bell Lab., Paris, 1997.

[39] Young, L. A., Booth, E. R., Jr., Yamauchi, G. K., Botha, G., and Dawson, S., "Overview of the Testing of a Small-Scale Proprotor," *American Helicopter Society 55th Annual Forum*, American Helicopter Society, Alexandria, VA, May 1999.

[40] Swanson, S. M., McCluer, M. S., Yamauchi, G. K., and Swanson, A. A., "Airloads Measurements from a $\frac{1}{4}$-Scale Tiltrotor Wind Tunnel Test," *25th European Rotorcraft Forum*, Sept. 1999.

[41] Nyukhtikov, M., Smelova, N., Mitchell, B. E., and Holmes, D. G., "Optimized Dual-Time Stepping Technique for Time-Accurate Navier–Stokes Calculation," *Proceedings of the 10th International Symposium on Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines*, Springer, Dordrecht, The Netherlands, 2003, pp. 449–462.

[42] 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

[43] Nielsen, E. J., Lee-Rausch, E. M., and Jones, W. T., "Adjoint-Based Design of Rotors Using the Navier–Stokes Equations in a Noninertial Reference Frame," *American Helicopter Society 65th Annual Forum*, American Helicopter Society, Alexandria, VA, May 2009.

Z. Wang
Associate Editor