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

- \(A\) = interpolation matrix
- \(A, B\) = amplitudes of rotation in degrees
- \(a, b, c, d\) = temporal coefficients
- \(C\) = aerodynamic coefficient
- \(C = m_0 \times 1\) vector of zeros and ones, indicator of time derivatives
- \(c\) = wing chord
- \(C_L\) = lift coefficient
- \(C_{M_1}, C_{M_2}\) = lateral and longitudinal moment coefficients
- \(C_Q\) = torque coefficient
- \(C_T\) = thrust coefficient
- \(D\) = vector of design variables
- \(Diag\) = diagonal matrix operator
- \(E\) = total energy per unit volume
- \(F\) = \(m_q \times 3\) flux matrix
- \(f, s\) = general functions
- \(F_{inv}, F_{visc}\) = inviscid and viscous fluxes
- \(f_{obj}\) = objective function
- \(G\) = grid operator
- \(g_1, g_2\) = explicit constraint functions
- \(I\) = projector operator
- \(i\) = \(\sqrt{-1}\)
- \(J\) = number of cost function components
- \(K\) = \(m_x \times m_y\) linear elasticity coefficient matrix
- \(L\) = Lagrangian function
- \(m_d\) = size of vector \(D\)
- \(m_f\) = size of solution vector at fringe points
- \(m_h\) = size of solution vector at hole points
- \(m_q\) = size of solution vector \(Q\)
- \(m_s\) = size of solution vector at solve points
- \(m_s\) = size of vector \(X\)
- \(N\) = number of time levels
- \(n\) = time level
- \(\mathbf{n}\) = \(3 \times 1\) outward-pointing normal vector
- \(P\) = \(m_0 \times m_q\) pseudo-Laplacian matrix
- \(p\) = pressure; also cost function exponent
- \(Q\) = \(m_q \times 1\) vector of volume-averaged conserved variables
- \(q\) = \(m_q \times 1\) vector of conserved variables
- \(R\) = \(3 \times 3\) rotation matrix
- \(R\) = \(m_x \times 1\) vector of spatial undivided residuals
- \(R\) = \(m_x \times m_y\) block-diagonal rotation matrix
- \(R_{GCL}\) = residual of static geometric conservation law
- \(R_{GCL}\) = \(m_x \times 1\) vector of \(R_{GCL}\)
- \(S\) = control volume surface area
- \(T\) = \(4 \times 4\) transform matrix
- \(t\) = time
- \(u, v, w\) = Cartesian components of velocity
- \(V\) = control volume
- \(V\) = \(m_q \times m_q\) diagonal matrix of cell volumes
- \(W\) = \(3 \times 1\) face velocity vector
- \(X\) = \(m_x \times 1\) vector of grid coordinates
- \(x\) = \(3 \times 1\) position vector
- \(x\) = independent variable
- \(x, y, z\) = Cartesian coordinate directions
- \(\alpha\) = interpolation coefficient
- \(\beta\) = scaling parameter for incompressible continuity equation
- \(\epsilon\) = perturbation
- \(\theta\) = angle of rotation; also blade pitch
- \(\theta_{ci}\) = collective input
- \(\theta_{ci}\) = lateral cyclic input
- \(\Lambda\) = \(m_q \times 1\) flowfield adjoint variable
- \(\Lambda_x\) = \(m_x \times 1\) grid adjoint variable
- \(\rho\) = density
- \(\tau\) = \(m_x \times 1\) translation vector
- \(\psi\) = blade azimuth
- \(\omega\) = 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. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1533-385X/13 and $10.00 in correspondence with the CCC.

*Research Engineer, Computational AeroSciences Branch, MS 128.
†Fellow, 100 Exploration Way; currently Visiting Associate Professor, Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA 22904. Associate Fellow AIAA.

Senior Member AIAA.
\[ \omega_1, \omega_2 = \text{frequencies of rotation, rad/s} \]
\[ * = \text{Hadamard vector multiplication operator} \]
\[ \circ = \text{extension of } * \text{ to a vector-matrix product} \]

**Subscripts/Superscripts**
- \( c \): child motion
- \( f \): fringe point
- \( h \): hole point
- \( i, j, k, \ldots \): indices
- \( m, n \): dimensions
- \( \text{in} \): quantity at initial conditions
- \( \text{nb} \): quantity at simply connected neighbor
- \( p \): parent motion
- \( s \): solve point
- \( x, y, z \): axis of rotation
- \( \infty \): quantity at freestream conditions
- \( \ast \): target quantity
- \( \bar{\cdot} \): 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], noniterative 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 \( C \) and a Hadamard vector multiplication operator \( [31] \). The vector \( C \) 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 \( C \), \( \bar{C} \), is a vector of the same dimension in which zeros are replaced by ones and vice versa. The Hadamard operator is denoted as \( * \) 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 \( C \) with the vector of time derivatives. The resulting equations can be written in the following form for both moving and stationary control volumes:

\[
C \frac{\partial}{\partial t} \int_V \mathbf{q} \, dV + \int_{\partial V} (F_{\text{inv}} - F_{\text{visc}}) \mathbf{n} \, dS = 0
\]

where \( V \) is the control volume bounded by the surface \( \partial V \), and \( \mathbf{n} \) is an outward-pointing unit normal. The vector \( \mathbf{q} \) represents the conserved variables for mass, momentum, and energy. The matrices \( F_{\text{inv}} \) and \( F_{\text{visc}} \) denote the inviscid and viscous fluxes, respectively; the fluxes
matrices are composed of three columns, each representing the flux components along coordinate directions. The product of a flux matrix and \( n \) 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
\[
\mathbf{F}_{\text{inv}} = \mathbf{F} - (C + \mathbf{q} + \mathbf{q}) W^T, \quad \text{where} \quad \mathbf{W} = [W_x, W_y, W_z]^T \quad \text{is a local face velocity. In other words, for an equation with a time derivative}
\]

By defining a volume-averaged quantity \( \mathbf{\bar{q}} \) within each control volume,
\[
\mathbf{\bar{q}} = \frac{1}{V} \int_{V} \mathbf{q} \, dV
\]

the conservation equations given by Eq. (1) take the form
\[
C \cdot \frac{\partial (\mathbf{\bar{q}V})}{\partial t} + \oint_{\partial V} (\mathbf{F}_{\text{inv}} - \mathbf{F}_{\text{visc}}) \mathbf{n} \, dS = 0
\]

where \( \mathbf{F}_{\text{inv}} \) and \( \mathbf{F}_{\text{visc}} \) are a control volume and the corresponding solution point, respectively. For an equation without a time derivative, \( \partial (\mathbf{\bar{q}V})/\partial t \) = 0, and

\[
\mathbf{F}_{\text{inv}} = \begin{bmatrix}
\rho(u-W_x) & \rho(v-W_y) & \rho(w-W_z) \\
\rho(u-W_x) + p & \rho(v-W_y) + p & \rho(w-W_z) + p \\
\rho v(u-W_x) & \rho v(v-W_y) + p & \rho v(w-W_z) + p \\
E(u-W_x) & E(v-W_y) + p & E(w-W_z) + p
\end{bmatrix}
\]

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

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

where \( \beta \) is a scaling parameter analogous to the artificial compressibility parameter \( \alpha \). 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
\[
\frac{\partial s}{\partial t} = \frac{1}{\Delta t} (as^n + bs^{n-1} + cs^{n-2} + ds^{n-3})
\]

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>
<thead>
<tr>
<th>Scheme</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>( d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDF1</td>
<td>1.0</td>
<td>−1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>BDF2</td>
<td>3/2</td>
<td>−2.0</td>
<td>1/2</td>
<td>0.0</td>
</tr>
<tr>
<td>BDF3</td>
<td>11/6</td>
<td>−3.0</td>
<td>3/2</td>
<td>−1/3</td>
</tr>
<tr>
<td>BDF2opt</td>
<td>1.66</td>
<td>−2.48</td>
<td>0.98</td>
<td>−0.16</td>
</tr>
</tbody>
</table>

Using a BDF1 scheme, the discrete form of the flow equations at time level \( n \) is given as
\[
C \cdot \mathbf{\bar{q}}^n V^n - \mathbf{\bar{q}}^{n-1} V^{n-1} + \frac{R^n}{\Delta t} = 0
\]

where \( V^n \) and \( \mathbf{\bar{q}}^n \) are a control volume and the corresponding solution vector at time level \( n \), and \( R^n \) 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:
\[
C \cdot \mathbf{\bar{q}}^n - \mathbf{\bar{q}}^{n-1} \frac{V^n - V^{n-1}}{\Delta t} + R^n_{\text{GCL}} (C \cdot \mathbf{\bar{q}}^{n-1} + \beta \mathbf{C}) = 0
\]

Here,
\[
R^n_{\text{GCL}} = \oint_{\partial V} \mathbf{W}^n F \mathbf{n} \, dS
\]

for an equation with a time derivative and
\[
-\beta R^n_{\text{GCL}} = 0
\]

for the incompressible continuity equation. Equation (8) is obtained by subtracting the GCL residual, multiplied by \( \mathbf{\bar{q}}^{n-1} \) 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, \( \mathbf{\tilde{q}}_f \), is defined as a linear combination of solution values at solve points, \( \mathbf{\bar{q}}_s \),
\[
\mathbf{\tilde{q}}_f = \sum_k a^k \mathbf{\bar{q}}_s = 0, \quad \sum_k a^k = 1
\]

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, \( \mathbf{\tilde{q}}_h \), is set to the average of the solution values at its simply connected neighbors, \( \mathbf{\bar{q}}_n \). This averaging procedure is equivalent to a discrete pseudo Laplacian, which is an elliptic operator:
\[
\sum_j (\mathbf{\bar{q}}_j - \mathbf{\bar{q}}_h) = 0
\]

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 overset 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 \( I_f^s \), \( I_f^h \), and \( I_h^s \) at time level \( n \). These operators are rectangular matrices of respective dimensions \( m_f \times m_s \), \( m_f \times m_f \), and \( m_h \times m_f \), and for which the rows contain a single unity entry complemented by zeros. The values \( m_f, m_s \), and \( m_h \) are the solution dimensions at all solve, fringe, and hole points, respectively, and \( m_f = m_s + m_f + m_h \) is the solution dimension at all grid points. The projectors are used to extract solutions at grid points of a specific type: \( Q^s = I_f^s Q^s \), \( Q^f = I_f^f Q^f \), and \( Q^h = I_h^h Q^h \), where \( Q^s \) is the vector of solution values at all grid points and \( Q^f, Q^f, Q^h \) 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

\[
C^n_s \cdot V^n_s - I_f^n Q^{n-1} + R^n + (I_f^n Q^n \cdot C^n_s + \beta C^n_h) \cdot R^{GCL}_{GC} = 0
\]

(14)

In Eq. (14) and all discussions to follow, \( R^n \) and \( R^{GCL}_{GC} \) are \( m_f \times 1 \) vectors that include residuals at solve points, \( V^n \) is an \( m_f \times 1 \) vector of control volumes for all equations at time level \( n \), \( V^n_s = I_f^n V^n \) is a subset of \( V^n \) corresponding to solve points, \( C^n_s \) is an \( m_f \times 1 \) vector indicator of a time derivative restricted to solve points at time level \( n \), and \( C^n_h \) is the complement of \( C^n_s \). 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

\[
A^n Q^n = 0
\]

(15)

where \( A^n \) is the \( m_f \times m_f \) 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

\[
P^n Q^n = 0
\]

(16)

where \( P^n \) is the \( m_h \times m_f \) matrix of the pseudo Laplacian given by Eq. (13).

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

\[
\begin{bmatrix}
\frac{1}{\Delta t} \text{Diag}(C^n_s \cdot V^n_s) + \frac{\partial R^n}{\partial X^n_s} & \frac{\partial R^n}{\partial X^n_f} & \frac{\partial R^n}{\partial X^n_h} \\
\frac{\partial R^n}{\partial X^n_f} & A^n_s & A^n_h \\
\frac{\partial R^n}{\partial X^n_h} & A^n_h & P^n_f & P^n_h
\end{bmatrix}
\]

(17)

where \( \text{Diag}(C^n_s \cdot V^n_s) \) is a diagonal \( m_f \times m_f \) matrix with the vector \( C^n_s \cdot V^n_s \) on the main diagonal; \( A^n_s \) is an \( m_f \times m_f \) diagonal matrix describing interpolation at fringe points; \( A^n_s \) and \( A^n_h \) are matrices with respective dimensions \( m_f \times m_h \) and \( m_f \times m_f \), describing interpolation from solve and hole points; and \( P^n_f \) and \( P^n_h \) are matrices with respective dimensions \( m_f \times m_f \) and \( m_f \times m_f \), 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 \( \partial R^n / \partial X^n_h = 0 \), \( A^n_h = 0 \), 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

\[
G^n(X, D) = 0
\]

(18)

where the \( m_f \times 1 \) vector \( X \) represents the coordinates of the composite overset mesh (meshes at several time levels may be involved), \( D \) 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 \( G^n \) may be specified for different component grids. The specific formulations for different grid motions are introduced next.
A. Grids Undergoing Rigid Motion

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

$$ x = Tx^0 \tag{19} $$

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

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

In an expanded form, $x = Rx^0 + \tau$. Here, the $3 \times 3$ matrix $R$ defines a general rotation and the vector $\tau$ 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 [36] for more details. For a rigid-motion formulation, the grid operator at time level $n$ is defined as

$$ G^n(X^n, \bar{X}^n, D) \equiv R^n X^n + \tau^n - X^n \tag{21} $$

or in abbreviated notation,

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

Here, $X^n$ and $\bar{X}^n$ are the grid vectors at the initial time level and time level $n$, respectively; $R^n$ is an $m_t \times m_t$ block-diagonal matrix with $3 \times 3$ blocks representing rotation, and $m_t$ being the size of vector $X^n$; and $\tau^n$ is an $m_t \times 1$ translation vector. The matrix $R^n$ and vector $\tau^n$ may explicitly depend on $D$.

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

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

where $K$ represents the elasticity coefficient matrix; $X$ is the vector of grid coordinates being solved for; $\bar{X}$ is the vector of coordinates of a reference grid; and $X_{\text{bound}}$ and $\bar{X}_{\text{bound}}$ are the vectors of corresponding boundary coordinates, complemented by zeros for all interior coordinates. The coefficients of the matrix $K$ depend on $X$. 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 general 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

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

where $X_{\text{bound}}$ may explicitly depend on $D$, $\bar{X}$ is an independent grid obtained either from a grid generator or from the previous optimization iteration, and $\bar{X}_{\text{bound}}$ 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 $K = K^0$ computed at the initial time level and fixed throughout the time evolution; $X_{\text{bound}}$ includes the description of the current body positions. The grid operator at time level $n$ is defined as

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

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 $T_p T_c$, where $T_p$ is the collective parent transform matrix (which itself can be a chain of parent–child products), and $T_c$ 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, $X^n$, is designated as rigid, then all nodes of this grid undergo the same motion described as

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

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

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

or computed from the elasticity equations; Eq. (25). The corresponding body surface undergoes the $T_p T_c$ motion, the external boundary and the initial (reference) grid undergo the $T_p$ motion, and the grid at time level $n$, $X^n$, satisfies the elasticity relations

$$ G^n(X^n, X^0, D) \equiv -K^n(X^n - T_p X^0) + X_{\text{bound}} - T_p \bar{X}_{\text{bound}} \quad \text{where} \quad X^n = X^n \quad \text{is equivalent to the rigidly moving one; Eq. (26).} \tag{28} $$

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

$$ K^n T_p = T_p K^n \tag{29} $$

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, $T_p T_c$, the boundary variations in Eq. (28) become zero, and the obtained grid, $X^n$, 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:

$$ f_i = \sum_{j=1}^{l_i} \omega_i (C_j - C_j^p)^p_i \tag{30} $$

Here, $\omega_i$ 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 exponent value of \( C_1 \), and \( p_1 \) 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 \( f_n^i \) 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 \([t^*, t^* + \Delta t]\):

\[
f_i = \sum_{n=N_1^i}^{N_2^i} f_n^i
\]

where time levels \( N_1^i \) and \( N_2^i \) correspond to \( t^* \) and \( t^* + \Delta t \), respectively. The corresponding time integral is approximated as \( f_i \cdot \Delta t \). The current study also introduces an additional cost function of the following form, which is based on the time-average value of an output:

\[
f_i = \left[ \left( \frac{1}{(N_2^i - N_1^i + 1)} \sum_{n=N_1^i}^{N_2^i} C_n^i \right) - C_f^i \right]^{p_i}
\]

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 class 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 [34] 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 \( D \) to minimize an objective function, \( f_{obj} = f \Delta t \), 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

\[
L(D, Q, X, \Lambda, A_s) = f \Delta t + \left[ [\Lambda_0^i]^T G^0 + [\Lambda_0^0]^T R^0 \right] \Delta t
\]

\[
+ \sum_{n=1}^{N} \left[ [\Lambda_0^i]^T G^n + [\Lambda_0^n]^T A^n Q^n + [\Lambda_0^n]^T [P^n Q^n] \right]
\]

\[
+ [\Lambda_0^n]^T \left( C_0^n \ast V_0^n + \frac{Q_0^n - I^n Q_{0^n -1}}{\Delta t} + R^n \right)
\]

\[
+ \left( (I^n Q_{0^n - 1}) \ast C_0^n + \beta C_0^n \ast R_{GCL} \right) \right] \Delta t
\]

(33)

Here, \( A_0^i \), \( A_0^n \), \( A_0^n \), and \( A_0^n \) are \( m_x, 1 \), \( m_y, 1 \), and \( m_z, 1 \) vectors of Lagrange multipliers associated with the solve, fringe, hole, and grid equations, respectively; \( [\Lambda_0^i]^T = [\Lambda_0^i]^T, [\Lambda_0^n]^T, [\Lambda_0^n]^T \); \( A_0^i = I^n A_0^n \), \( A_0^n = I^n A_0^n \), and \( A_0^n = I^n A_0^n \); and \( R^n = 0 \) represents the initial conditions. A typical form of the initial conditions is \( R^n \equiv V^n \ast (Q_0^n - Q^n) \), where \( Q_0^n \) 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 \( D \), assuming that \( V^n \) depends on \( X^n; G^n \) depends on \( X^n, X^0 \), and \( D; R^n \) depends on \( Q^n, X^n, X^{n-1} \), and \( D; R_{GCL} \) depends on \( X^n, X^{n-1} \), and \( D; A^n \) depends on \( X^n, G^n \) depends on \( X^n, X^0 \), and \( D; R^n \) depends on \( Q^n, X^n, X^{n-1} \), and \( D; P^n, C^n, I^n, I^n, I^n, I^n, I^n, I^n \), and \( I^n \) are independent of grid coordinates, solutions, and design parameters.

Regrouping terms to collect the coefficients of \( \partial Q^n / \partial D \) and equating those coefficients to zero yields the adjoint equations:

\[
S:\]

\[
\frac{1}{\Delta t} C^n_0 \ast V^n_0 \ast A^n_0 + \left[ \frac{\partial R^n}{\partial Q^n} \right]^T A^n_0 + \left[ \frac{\partial A^n}{\partial Q^n} \right]^T A^n_0 + \left[ \frac{\partial P^n}{\partial Q^n} \right]^T A^n_0
\]

\[
= - \left( \frac{\partial f}{\partial Q^n} \right)^T \left[ -I^n \left[ V^{n+1}_0 + R^{n+1}_{GCL} \right] \ast A^{n+1}_0 \right]
\]

\[
F:\]

\[
\frac{\partial R^n}{\partial Q^n} \ast A^n_0 + \left[ \frac{\partial A^n}{\partial Q^n} \right]^T A^n_0 + \left[ \frac{\partial P^n}{\partial Q^n} \right]^T A^n_0
\]

\[
= - \left( \frac{\partial f}{\partial Q^n} \right)^T \left[ -I^n \left[ I^n + V^{n+1}_0 + R^{n+1}_{GCL} \right] \ast A^{n+1}_0 \right]
\]

\[
H:\]

\[
\frac{\partial R^n}{\partial Q^n} \ast \Lambda^n_0 + \left[ \frac{\partial A^n}{\partial Q^n} \right]^T \Lambda^n_0 + \left[ \frac{\partial P^n}{\partial Q^n} \right]^T \Lambda^n_0
\]

\[
= - \left( \frac{\partial f}{\partial Q^n} \right)^T \left[ -I^n \left[ I^n + V^{n+1}_0 + R^{n+1}_{GCL} \right] \ast A^{n+1}_0 \right]
\]

for \( 1 \leq n \leq N \)

\[
\left( \frac{\partial R^n}{\partial Q^n} \right)^T \Lambda^n_0 = - \left( \frac{\partial f}{\partial Q^n} \right)^T \left[ -I^n \left[ V^{n+1}_0 + R^{n+1}_{GCL} \right] \ast A^1 \right]
\]

for \( n = 0 \)

(34)

where \( \Lambda^{n+1}_0 = 0 \). 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 \( \partial X^n / \partial D \) and equating those coefficients to zero in a similar fashion yields the grid adjoint equations:
computations for time level $n$. The sensitivity derivates [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 $Q_{t}^{n-1}$ is taken to be an initial approximation for $Q_t$. 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):

$$F:
A_t^s\delta Q_t^{n,m} = -[A_t^sQ_t^{n,m} + A_t^sQ_t^{n,m} + A_h^sQ_h^{n,m}]
Q_f^{n,m+1} = Q_f^{n,m} + \Delta Q_f^{n,m}
$$

$$S:
\begin{align*}
\frac{1}{\Delta t} \text{Diag}(V_t^n) + \frac{1}{\Delta t} \text{Diag}(C_t^n + \beta C_t^n) + \frac{\partial R_{\text{dGL}}^{n,m}}{\partial \Omega_t} & \Delta Q_t^{n,m} \\
= \left[C_t^n + \beta C_t^n \right] \Delta Q_t^{n,m} + \frac{\partial R_{\text{dGL}}^{n,m}}{\partial \Omega_t} & \\
+ \left(1 - Q_t^{n-1} \right) \left[C_t^n + \beta C_t^n \right] R_{\text{dGL}}^{n,m} & \\
Q_t^{n,m+1} = Q_t^{n,m} + \Delta Q_t^{n,m}
\end{align*}
$$

$$H:
P_h^s \Delta Q_h^{n,m} = -\left[ P_h^sQ_h^{n,m+1} + P_h^sQ_h^{n,m+1} + P_h^sQ_h^{n,m} \right] \\
Q_h^{n,m+1} = Q_h^{n,m} + \Delta Q_h^{n,m}
$$

Here, the second superscript $m$ is the iteration count, $R_{\text{dGL}}^{n,m}$ is the spatial nonlinear residual computed for the most recent solution that involves $Q_{t}^{n,m+1}$ and $Q_{t}^{n,m}$, $\Delta t$ is a pseudo-time step, and $\partial R_{\text{dGL}}^{n,m}/\partial Q_t^n$ is the Jacobian of a first-order spatial discretization.

At each iteration, Eq. (37) is solved exactly because $A_t^n$ 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 $Q_{t}^{n,m+1}$. 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 Eqs. (34) at time level $n$ is performed in precisely the reverse order as the iterations given by Eqs. (37)–(39):

$$H:
\begin{align*}
[P_h^s]^T \Delta A_h^{n,m} & = \left[ \frac{\partial f}{\partial Q_h^n} \right]^T
- \left[ C_t^n + \beta C_t^n \right] \left( -\frac{1}{\Delta t} V_t^{n+1} + R_{\text{dGL}}^{n+1} \right) \Delta A_t^{n+1} \\
- \left[ P_h^s\right]^T A_h^{n,m} - \left[ \frac{\partial R_{\text{dGL}}^{n,m}}{\partial \Omega_t} \right] A_h^{n,m} - \left[ A_h^s\right]^T \Delta A_h^{n,m} \\
A_h^{n,m+1} = A_h^{n,m} + \Delta A_h^{n,m}
\end{align*}
$$
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.

\[
S: \quad \left[ \frac{1}{\Delta t} \text{Diag}(V^*) + \frac{1}{\Delta t} \text{Diag}(C^o + V) + \frac{\partial R^m}{\partial Q^*} \right] \Delta \lambda^{n,m}
= -\left[ \frac{\partial f}{\partial Q^*} \right]^T \left[ \frac{1}{\Delta t} \text{Diag}(C^o + V) + \frac{\partial R^m}{\partial Q^*} \right] \Delta \lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m+1}
\]
\[\Lambda^{n,m+1} = \Lambda^{n,m} + \Delta \lambda^{n,m} \tag{41}\]

\[
F:\quad \left[ \Lambda^f \right]^T \Delta \lambda^{n,m}
= -\left[ \frac{\partial f}{\partial Q^*} \right]^T \left[ \frac{1}{\Delta t} \text{Diag}(C^o + V) + \frac{\partial R^m}{\partial Q^*} \right] \Delta \lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m} - \left[ \frac{\partial R^m}{\partial Q^*} \right]^T \Lambda^{n,m+1}
\]
\[\Lambda^f \Lambda^{n,m+1} = \Lambda^f \Lambda^{n,m} + \Delta \lambda^{n,m} \tag{42}\]

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

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 \( i\varepsilon \):

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

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 \( 10^{-5} \), 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 \( \psi \) 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.
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 are shown in Table 3. The results from the discrete adjoint and complex-variable approaches are in very good agreement 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 very good agreement for all cases; nonmatching digits in the sensitivities are underlined.

<table>
<thead>
<tr>
<th>Variable</th>
<th>BDF1</th>
<th>BDF2</th>
<th>BDF2opt</th>
<th>BDF3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle of attack</td>
<td>A: 0.116458691618373</td>
<td>A: 0.10209965021956</td>
<td>A: 0.102091575231413</td>
<td>A: 0.103785048456802</td>
</tr>
<tr>
<td>C: 0.390973864</td>
<td>C: 0.390973864</td>
<td>C: 0.390973864</td>
<td>C: 0.390973864</td>
<td></td>
</tr>
<tr>
<td>Rotation rate: blade 1</td>
<td>A: 0.619149219921508</td>
<td>A: 0.609270815829788</td>
<td>A: 0.592456231908979</td>
<td>A: 0.570591540947799</td>
</tr>
<tr>
<td>C: 0.619149219921508</td>
<td>C: 0.609270815829788</td>
<td>C: 0.592456231908979</td>
<td>C: 0.570591540947799</td>
<td></td>
</tr>
<tr>
<td>Shape: blade 2</td>
<td>A: 0.056440771725091</td>
<td>A: 0.06438237311983</td>
<td>A: 0.062734653842921</td>
<td>A: 0.060943525612814</td>
</tr>
<tr>
<td>C: 0.056440771725091</td>
<td>C: 0.06438237311983</td>
<td>C: 0.062734653842921</td>
<td>C: 0.060943525612814</td>
<td></td>
</tr>
<tr>
<td>Flap frequency: blade 3</td>
<td>A: -0.414712919056299</td>
<td>A: -0.33225902074676</td>
<td>A: -0.34455551267488</td>
<td>A: -0.35241956884976</td>
</tr>
<tr>
<td>Translation rate: fuselage</td>
<td>A: 0.42030051382122</td>
<td>A: 0.40037176530665</td>
<td>A: 0.390738649649789</td>
<td>A: 0.379952931745697</td>
</tr>
<tr>
<td>Shape: fuselage</td>
<td>A: 0.007890949237659</td>
<td>A: -0.009509444345683</td>
<td>A: -0.009613538492229</td>
<td>A: -0.009705401931792</td>
</tr>
<tr>
<td>C: 0.007890949237659</td>
<td>C: -0.009509444345683</td>
<td>C: -0.009613538492229</td>
<td>C: -0.009705401931792</td>
<td></td>
</tr>
</tbody>
</table>

*Symbols A and C denote adjoint and complex-variable results, respectively.
*Discrepancies are shown in bold and 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.

<table>
<thead>
<tr>
<th>Variable</th>
<th>BDF1</th>
<th>BDF2</th>
<th>BDF2opt</th>
<th>BDF3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle of attack</td>
<td>A: 0.000195945789030</td>
<td>A: 0.000234143173131</td>
<td>A: 0.000218182269639</td>
<td>A: 0.000191641169710</td>
</tr>
<tr>
<td>Rotation rate: blade 1</td>
<td>A: 0.000518073976865</td>
<td>A: 0.010329090370672</td>
<td>A: 0.010549871829212</td>
<td>A: 0.010757597021050</td>
</tr>
<tr>
<td>Shape: blade 2</td>
<td>A: 0.000535202541590</td>
<td>A: 0.000607314158464</td>
<td>A: 0.00061881998355</td>
<td>A: 0.000637367518752</td>
</tr>
<tr>
<td>Flap frequency: blade 3</td>
<td>A: 0.004866399384562</td>
<td>A: -0.004825188590676</td>
<td>A: -0.004825188590676</td>
<td>A: -0.004825188590676</td>
</tr>
<tr>
<td>Rotation rate: fuselage</td>
<td>A: 0.420396201952597</td>
<td>A: 0.449692632180217</td>
<td>A: 0.44947751807594</td>
<td>A: 0.44876653248215</td>
</tr>
<tr>
<td>Translation rate: fuselage</td>
<td>A: 0.010054193904733</td>
<td>A: 0.010404541410124</td>
<td>A: 0.010284602229431</td>
<td>A: 0.010438086875134</td>
</tr>
<tr>
<td>Shape: fuselage</td>
<td>A: 0.000870619953334</td>
<td>A: 0.000795891348124</td>
<td>A: 0.000822719730679</td>
<td>A: 0.000867531788234</td>
</tr>
</tbody>
</table>

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 non-dimensionalization 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:

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

(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 shown 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
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.

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:

$$
\theta_x = A_x[\cos(\omega_{1x}t) - 1] + B_x \sin(\omega_{2x}t)
$$

$$
\theta_y = A_y[\cos(\omega_{1y}t) - 1] + B_y \sin(\omega_{2y}t)
$$

$$
\theta_z = A_z[\cos(\omega_{1z}t) - 1] + B_z \sin(\omega_{2z}t)
$$

(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 $\pm 60$ deg about its root chord line. The second rotation is a feathering motion that rotates the wing $\pm 45$ 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.
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 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Baseline</th>
<th>Distribution target function</th>
<th>Time-average target function</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-COR</td>
<td>0.000</td>
<td>0.025c</td>
<td>0.027c</td>
</tr>
<tr>
<td>y-COR</td>
<td>0.000</td>
<td>−0.119c</td>
<td>−0.114c</td>
</tr>
<tr>
<td>z-COR</td>
<td>0.000</td>
<td>0.011c</td>
<td>0.012c</td>
</tr>
<tr>
<td>A_x</td>
<td>0.000</td>
<td>0.77</td>
<td>−0.11</td>
</tr>
<tr>
<td>B_x</td>
<td>45.00</td>
<td>45.13</td>
<td>45.25</td>
</tr>
<tr>
<td>(\omega_{x1})</td>
<td>163.36</td>
<td>163.45</td>
<td>163.36</td>
</tr>
<tr>
<td>(\omega_{x2})</td>
<td>163.36</td>
<td>177.47</td>
<td>192.77</td>
</tr>
<tr>
<td>A_y</td>
<td>0.000</td>
<td>0.30</td>
<td>−0.99</td>
</tr>
<tr>
<td>B_y</td>
<td>0.000</td>
<td>−1.50</td>
<td>−0.26</td>
</tr>
<tr>
<td>(\omega_{y1})</td>
<td>163.36</td>
<td>162.76</td>
<td>163.15</td>
</tr>
<tr>
<td>(\omega_{y2})</td>
<td>163.36</td>
<td>163.10</td>
<td>162.97</td>
</tr>
<tr>
<td>A_z</td>
<td>−60.00</td>
<td>−62.71</td>
<td>−62.83</td>
</tr>
<tr>
<td>B_z</td>
<td>0.000</td>
<td>0.69</td>
<td>−1.55</td>
</tr>
<tr>
<td>(\omega_{z1})</td>
<td>163.36</td>
<td>173.59</td>
<td>189.57</td>
</tr>
<tr>
<td>(\omega_{z2})</td>
<td>163.36</td>
<td>164.41</td>
<td>163.55</td>
</tr>
</tbody>
</table>

\(^{4}\)COR denotes the center of rotation.

The convergence history for the objective function based on a target thrust distribution:

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

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

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

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 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.
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:

\[ \theta = \theta_c + \theta_{1c} \cos \psi + \theta_{1s} \sin \psi \]  
(48)

Here, \( \theta \) is the current blade pitch setting, \( \psi \) is the current azimuth position for the blade, \( \theta_c \) represents the collective control input, and \( \theta_{1c} \) and \( \theta_{1s} \) 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.

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 \( \times \) 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 \( \theta_{1c} \), \( \theta_{1s} \), and \( \theta_{1l} \) to also be used as design variables. These control angles are allowed to vary as much as \( \pm 7 \) 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.
The objective function to be minimized is based on the time-average value of the lift coefficient over the second rotor revolution:

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

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:

\[ g_1 = \frac{1}{360} \sum_{n=361}^{720} C_{M_x}^n \Delta t \quad (50) \]

\[ g_2 = \frac{1}{360} \sum_{n=361}^{720} C_{M_y}^n \Delta t \quad (51) \]

The constraints are considered satisfied if \( g_1 = g_2 = 0 \), within a feasibility tolerance of \( \pm 0.0001 \). 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 \( \theta_c \) is 6.71 deg, while the final values for the cyclic inputs \( \theta_{1c} \) and \( \theta_{1s} \) 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. 18 Baseline and final \( C_{M_x} \) profiles for the UH-60 configuration.

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.
features originate on blade surfaces and propagate upstream. A general verified methodology for adjoint-based design optimization of unsteady turbulent flows on dynamic 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.

Figure 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.

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

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.

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 benefited 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 Yiping 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 description 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
\[\begin{align*}
C^n_j &= \left[ a \frac{Q^n_j - L_i Q^{n-1}_j}{\Delta t} + c \frac{L_i Q^{n-2} - L_i Q^{n-1}_j}{\Delta t} + L_i^r V^n_{j+2} \right] + \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = 0 \quad (A1)
\end{align*}\]

Proceeding as before, the Lagrangian can be written as

\[\begin{align*}
L(D, Q, X, \Lambda, \Lambda_x) &= f + \sum_{n=1}^{N} \left( \left[ C^n_j + \Lambda^n_j \right] \left[ a \frac{Q^n_j - L_i Q^{n-1}_j}{\Delta t} + c \frac{Q^n_j - L_i Q^{n-1}_j}{\Delta t} + L_i^r V^n_{j+2} \right] \right) \\
&+ \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = 0
\end{align*}\]  

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:

\[\begin{align*}
S: \quad \frac{a}{\Delta t} V^n_j + C^n_j + \Lambda^n_j + [A^n_j]^{T} + \left[ \delta Q^n_j \right] \right] + [A^n_j]^{T} + \left[ \delta Q^n_j \right] \right] + \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = - \left[ \frac{df}{\partial Q^n_j} \right] \\
&+ \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = 0
\end{align*}\]

\[\begin{align*}
F: \quad \frac{dR^n_j}{Q^n_j} \Lambda^n_j + [A^n_j]^{T} \Lambda^n_j + [P^n_j]^{T} \Lambda^n_j = - \left[ \frac{df}{\partial Q^n_j} \right] \\
&+ \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = 0
\end{align*}\]

\[\begin{align*}
H: \quad \frac{dR^n_j}{Q^n_j} \Lambda^n_j + [A^n_j]^{T} \Lambda^n_j + [P^n_j]^{T} \Lambda^n_j = - \left[ \frac{df}{\partial Q^n_j} \right] \\
&+ \left( \beta \frac{C^n_j}{\Delta t} \right) + R^n_{\text{GCL}} = 0
\end{align*}\]  

for \(3 \leq n \leq N\)  

(A3)

(A4)
\[ H: \]
\[ \begin{align*}
\frac{\partial \mathbf{u}^n}{\partial \mathbf{Q}^n} & = [\mathbf{Q}^n - I_n \mathbf{Q}^n - I_n \mathbf{Q}^{n-1} \circ \frac{\partial \mathbf{V}^n}{\partial \mathbf{X}^n} \circ (\mathbf{C}_s^3 \circ \Lambda^3)] \\
& + \frac{1}{2} \left( I_n^{+2} \mathbf{Q} - I_n^{+2} \mathbf{Q}^{n+1} \right) \circ \left( \frac{d}{\Delta t} I_n^{+2} \mathbf{V} \cdot \mathbf{C}_s^2 \circ \Lambda^2 \right) \\
& + \frac{d}{\Delta t} I_n^{+3} \mathbf{Q} - I_n^{+3} \mathbf{Q}^{n+1} \right) \circ \left( \frac{d}{\Delta t} I_n^{+3} \mathbf{V} \cdot \mathbf{C}_s^3 \circ \Lambda^3 \right) \\
& - \frac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n} \Lambda^n \\
& + \frac{\partial f}{\partial \mathbf{X}^n} \\
\end{align*} \]

for \( n = 1 \)

\[ \frac{\partial \mathbf{u}^n}{\partial \mathbf{Q}^n} \Lambda^n = \left[ \frac{\partial \mathbf{u}^n}{\partial \mathbf{Q}^n} \Lambda^n \right] \]

The corresponding grid adjoint equations are obtained as follows: Assuming \( \Lambda^{n+1} = \Lambda^{n+2} = \Lambda^{n+3} = 0 \):

\[ \begin{align*}
\frac{\partial \mathbf{u}^n}{\partial \mathbf{Q}^n} \Lambda^n & = \left[ \frac{\partial \mathbf{u}^n}{\partial \mathbf{Q}^n} \Lambda^n \right] \\
& - \frac{\partial f}{\partial \mathbf{X}^n} \\
& + \frac{\partial (\mathbf{A}^n \mathbf{Q}^n)}{\partial \mathbf{X}^n} \Lambda^n \\
& + \frac{\partial f}{\partial \mathbf{X}^n} \\
\end{align*} \]

for \( n = 0 \) (A5)

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

References


