AIAA Journal
Vol. 48, No. 12, December 2010

Adjoint Sensitivity Formulation for Discontinuous Galerkin Discretizations in Unsteady Inviscid Flow Problems

Li Wang*
University of Tennessee at Chattanooga, Chattanooga, Tennessee 37403
Dimitri J. Mavriplis†
University of Wyoming, Laramie, Wyoming 82071
and
W. Kyle Anderson‡
University of Tennessee at Chattanooga, Chattanooga, Tennessee 37403

DOI: 10.2514/1.J050444

This paper presents an unsteady discrete adjoint algorithm for high-order implicit discontinuous Galerkin discretizations in time-dependent inviscid flow problems. The major function of the adjoint approach is to obtain the sensitivity information in a time-dependent functional output, which in turn is used to drive an unsteady shape-optimization process to deliver a minimum of the objective functional. A gradient-based optimization strategy is investigated, in which the sensitivity derivatives of the objective functional with respect to input variables are formulated in the context of high-order discontinuous Galerkin discretizations, while special emphasis is given to the variations and linearizations of curvilinear boundary elements. Implicit temporal discretizations consisting of a second-order backward Euler scheme and a fourth-order implicit Runge–Kutta scheme are considered exclusively in this work, where the corresponding adjoint problem is required to be solved in a backward time-integration manner due to the associated transpose operation. Two numerical examples for the unsteady shape design techniques are presented to verify the derived sensitivity formulations and to demonstrate the performance of the adjoint approach; the first involves an inverse shape-optimization case by matching a time-dependent target pressure profile for a two-dimensional periodic vortical gust impinging on a RAE-2822 airfoil, and the second considers minimization of the acoustic noise produced by subsonic flow over a NACA0012 airfoil with a 0.03*c* thick blunt trailing edge.

Nomenclature

a = coefficient matrix for a six-stage explicit singly diagonally implicit Runge–Kutta scheme
a_∞ = freestream speed of sound
b = Hicks–Henne bump function
C_l = lift coefficient
c = chord length
$\mathbf{D}$ = design variables
e = specific total energy per unit mass
$\mathbf{f}_1$, $\mathbf{f}_2$ = inviscid Cartesian flux components
J = volume Jacobian
K = stiffness matrix
$\mathbf{k}$ = wave number vector (k_1, k_2)
L = scalar-valued time-dependent objective functional
$\mathbf{M}$ = mass matrix
M_∞ = freestream Mach number
N, N^* = number of time steps for flow and objective
$\mathbf{n}$ = outward unit normal vector (n_x, n_y)
p = pressure
p' = pressure disturbance
$\mathbf{R}$ = spatial flow residual
$\mathbf{R}_e$ = unsteady flow residual
$\mathbf{u}$ = conservative flow variables
u, v = Cartesian flow velocity components
$(u, v)_\parallel$ = tangential velocity vector at physical boundary
$\mathbf{u}_p$ = Galerkin solution approximation
$\tilde{\mathbf{u}}_p$ = time-dependent modal solution coefficients
$\mathbf{u}^b$ = time-dependent boundary conditions
$\tilde{\mathbf{u}}^0$ = initial flow condition
$\mathbf{x}$ = Cartesian coordinate (x, y)
$\tilde{\mathbf{x}}$ = geometric mapping coefficients
$\mathbf{x}_q$ = coordinates of additional surface quadrature points
$\mathbf{x}_s$ = coordinates of surface nodes
γ = ratio of specific heats
Δt = time-step size
$\Delta \mathbf{x}$ = displacements of mesh points
$\Delta \mathbf{x}_q$ = displacements of additional surface quadrature points
$\Delta \mathbf{x}_s$ = displacements of surface mesh points
ε = gust amplitude parameter, $\varepsilon \ll 1$
$\boldsymbol{\lambda}_u$ = flow-adjoint variables
$\boldsymbol{\lambda}_x$ = mesh-adjoint variables
ξ, η = reference triangle coordinates
ρ = density
$\boldsymbol{\Phi}$ = projection mapping matrix
ϕ = basis function
Ω = two-dimensional bounded domain
ω = gust angular frequency

I. Introduction

HIGH-ORDER discontinuous Galerkin (DG) methods have become a popular approach over the last decade in solving a variety of computational fluid dynamics problems [1–6]. The advantage of high-order DG discretizations is that they alter the asymptotic relationship between solution accuracy and resolution in a beneficial manner. Over the same period, the incorporation of sensitivity analysis techniques [7–13] has become the subject of

---

Presented as Paper 2010-367 at the 48th AIAA Aerospace Science Meeting, Orlando, FL, 4–7 January 2010; revision received 19 July 2010; accepted for publication 7 August 2010.

*Research Assistant Professor, Graduate School of Computational Engineering, 701 East M. L. King Boulevard. Member AIAA.

†Professor, Department of Mechanical Engineering, 1000 East University Avenue. Associate Fellow AIAA.

‡Professor, Graduate School of Computational Engineering, 701 East M. L. King Boulevard. Associate Fellow AIAA.

increased interest in many numerical simulations. The sensitivities of specific simulation outputs with respect to simulation inputs deliver important information in a simulation, which in turn can be used to drive a design-optimization process [14,15] to obtain a minimum of an objective functional. In addition, adjoint methods in gradient-based design-optimization techniques enable efficient calculation of the sensitivity of an objective with respect to any number of design variables at a cost that is essentially independent of the number of design variables and roughly equivalent to an additional flow analysis problem [16,17]. While the majority of work in adjoint-based aerodynamic shape optimizations has been focused on the use of finite volume methods in a steady flow environment [16,18–20], relatively little work has been done with regard to the use of high-order DG methods for shape designs in an unsteady flow environment [21–24]. Therefore, the goal of this work is to develop an unsteady discrete adjoint algorithm for high-order discontinuous Galerkin discretizations in time-dependent flow problems and to apply this technique to unsteady shape-optimization problems.

It is known that high-order methods require the use of curved boundary elements to deliver an overall high-accuracy solution [4,25,26]. This is especially the case for problems with complicated geometries in which high-order methods typically use relatively coarse or large boundary elements, compared with lower-order methods. Therefore, this work uses additional surface quadrature points [4,12] on physical (wall) boundaries to determine the geometry mappings of high-order-accurate geometric surface shapes. In a shape-optimization process of this circumstance, when design variables produce changes in the shape of the domain boundaries, the additional face quadrature points must be deformed in the same manner as the surface nodes. Accordingly, the linearizations of the discretized unsteady flow equations as well as the objective functionals with respect to the mesh geometry must consider the variation from both grid points and extra surface quadrature points in the element mappings. The present work focuses on the development of the techniques required to incorporate these effects and results are demonstrated in two unsteady shape-optimization problems.

In each typical optimization iteration, numerical approximations for the time-dependent flow problem and adjoint problem are required, where the unsteady flow solution is solved in a forward integration in time. Because of the transpose operator in the discrete adjoint formulations, the unsteady adjoint solution that corresponds to a series of linear problems spanning the entire time domain is solved after the flow analysis problem, but in a backward time-integration manner. To avoid the restriction of time-step sizes of explicit schemes [27,28], we make exclusive use of implicit temporal schemes, consisting of a second-order backward-difference Euler (BDF2) scheme and a fourth-order implicit Runge–Kutta (IRK4) scheme [29]. Note that the difference of the adjoint solution procedure between the IRK4 scheme and the BDF2 scheme lies in the fact that the adjoint variables in the IRK4 scheme are required to be computed at both the discrete time-step locations and all the intermediate stages for each implicit time step.

Additionally, one of the flow problems of interest in the present work consists of a single-airfoil gust-response case taken from the computational aeroacoustics (CAA) benchmark problems [30], in which a highly-accurate time-dependent solution is required to capture the airfoil unsteady pressure, the noise generation and radiation produced by a periodic vortical gust impinging on a Joukowski airfoil. This type of the gust interaction problem has many practical applications, such as turbomachines or helicopter rotors, and hence a series of computational aeroacoustics solvers [31–33] have been developed in the past. This work investigates current capabilities of the high-order implicit DG solver for the aeroacoustics problem and applies the aeroacoustics problem to an unsteady airfoil shape-optimization case.

The outline of this paper is as follows. In Sec. II the governing equations are introduced and the spatial discontinuous Galerkin discretizations as well as implicit time-integration schemes are formulated. Sec. III focuses on the formulation of discrete adjoint-based sensitivity derivatives in the context of high-order discontinuous Galerkin methods for unsteady shape-optimization problems, with special emphasis given on the variation and re-creation of high-order curved boundary or surface elements. The validation of the present DG solver for the CAA benchmark problem is first provided in Sec. IV, followed by two numerical examples for verifying the unsteady discrete adjoint formulations derived in this work and for demonstrating the performance of the unsteady adjoint techniques. Finally, Sec. V summarizes the conclusions of the current work and discusses some possible directions for future work.

II. Governing Equations and Discretizations

The governing equations that we consider exclusively in this work are the two-dimensional compressible Euler equations that can be written in the following conservative form:

$$\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{u}(\mathbf{x},t))}{\partial x} + \frac{\partial \mathbf{f}_2(\mathbf{u}(\mathbf{x},t))}{\partial y} = 0 \quad \text{in } \Omega \tag{1}$$

where Ω is a two-dimensional bounded domain. The vector of conservative flow variables $\mathbf{u}$ and the inviscid Cartesian flux components $\mathbf{f}_1$ and $\mathbf{f}_2$ are defined, respectively, by

$$\mathbf{u} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{Bmatrix}, \quad \mathbf{f}_1 = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho e + p)u \end{Bmatrix}, \quad \mathbf{f}_2 = \begin{Bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho e + p)v \end{Bmatrix} \tag{2}$$

where the notations ρ, p, and e denote the fluid density, pressure, and specific total energy per unit mass, respectively, and u and v represent the flow velocity components in the x and y-coordinate directions. This system of equations is completed by the perfect gas equation of state given as

$$p = (\gamma - 1)[\rho e - \tfrac{1}{2}\rho(u^2 + v^2)] \tag{3}$$

where γ is defined as the ratio of specific heats, which is 1.4 for air.

A. Discontinuous Galerkin Discretizations

The computational domain Ω is partitioned into a tessellation of nonoverlapping triangular elements, such that

$$\Omega = \bigcup_k \Omega_k$$

where Ω_k refers to the volume of an element k in the computational mesh. The discontinuous Galerkin discretization proceeds by formulating a weak statement of the governing equations, by multiplying Eq. (1) by a set of test functions, $\{\phi_j, j = 1, \ldots, M\}$, with the maximum polynomial order of p, and integrating within each element, e.g., k, as

$$\int_{\Omega_k} \phi_j \left[\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}_1(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{f}_2(\mathbf{u})}{\partial y} \right] \mathrm{d}\Omega_k = 0 \tag{4}$$

Integrating this equation by parts, the weak statement of the problem becomes

$$\int_{\Omega_k} \phi_j \frac{\partial \mathbf{u}_p}{\partial t}\,\mathrm{d}\Omega_k - \int_{\Omega_k} \left[\frac{\partial \phi_j}{\partial x} \mathbf{f}_1(\mathbf{u}_p) + \frac{\partial \phi_j}{\partial y} \mathbf{f}_2(\mathbf{u}_p) \right] \mathrm{d}\Omega_k + \int_{\partial\Omega_k \backslash \partial\Omega} \phi_j \mathbf{H}(\mathbf{u}_p^+, \mathbf{u}_p^-, \mathbf{n})\,\mathrm{d}S + \int_{\partial\Omega_k \cap \partial\Omega} \phi_j \mathbf{H}^b(\mathbf{u}_p^-(\mathbf{u}_p^+, \mathbf{n}), \mathbf{n})\,\mathrm{d}S = 0 \tag{5}$$

where the unit normal vector $\mathbf{n} = (n_x, n_y)$ is outward to the boundary, and $\mathbf{u}_p^+$ and $\mathbf{u}_p^-$ refer, respectively, to the interior and exterior Galerkin solution approximations at shared interelement interfaces. Current implementations of the interior boundary flux function $\mathbf{H}(\mathbf{u}_p^+, \mathbf{u}_p^-, \mathbf{n})$ include the Riemann flux approximation of Harten–Lax–van Leer for contact wave (HLLC) [34] and Lax–Friedrichs [35]. For edges coinciding with the computational domain boundaries, the approximate flux function $\mathbf{H}^b(\mathbf{u}_p^-(\mathbf{u}_p^+,\mathbf{n}),\mathbf{n})$ is required to be explicitly dependent on exterior traces to satisfy the dual consistency condition [36]. Here, we set

$$\mathbf{H}^b(\mathbf{u}_p^-(\mathbf{u}_p^+,\mathbf{n}),\mathbf{n}) = \mathbf{f}_1(\mathbf{u}_p^-)n_x + \mathbf{f}_2(\mathbf{u}_p^-)n_y$$

where $\mathbf{u}_p^-$ is determined by the interior flow approximation, $\mathbf{u}_p^+$, as well as the given boundary conditions. In particular, total temperature and pressure are prescribed on the inflow boundary and static pressure is prescribed on the outflow boundary, and at solid walls $\mathbf{u}_p^-$ is set to have the same density, total energy, and tangential velocity $(u,v)_{\parallel}^t$ as $\mathbf{u}_p^+$, given by

$$(u,v)_{\parallel}^t = (u^+,v^+)^t - (u^+n_x + v^+n_y)(n_x,n_y)^t \tag{6}$$

The Galerkin finite element approximation in the weak formulation is expanded as a series of truncated basis functions [29] and solution coefficients as

$$\mathbf{u}_p = \sum_{i=1}^{M} \tilde{\mathbf{u}}_{p_i}\phi_i(\mathbf{x}) \tag{7}$$

Since the set of basis functions is defined in a reference triangle $\bar{\Omega}$ spanning between $\{0 < \xi, \eta < 1\}$ [37], a coordinate mapping from the reference to a physical triangle, illustrated by Fig. 1, is required for the computation of the first-order derivatives and integrals denoted in Eq. (5). For isoparametric elements considered in this work (in that they are generally sufficient for delivering as accurate a solution as superparametric elements when $p > 1$ [25]), the reference-to-physical transformation and the corresponding Jacobian J_k associated with each element k are given by

$$\mathbf{x}_k = \sum_{i=1}^{M} \tilde{\mathbf{x}}_{k_i}\phi_i(\xi,\eta) \qquad J_k = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{8}$$

In the simple case of straight-sided elements, the transformation is linear; thus, the geometric mapping $\tilde{\mathbf{x}}_k$ can be evaluated only by using the element vertex coordinates. However, in the more complex cases of high-order curved elements, which are typically required at curved wall boundaries even for inviscid flows, additional surface nodes [38] must be included for determining the higher-order modes ($p > 1$) of the geometric mapping coefficients, obtained by

$$\tilde{\mathbf{x}}_k = \mathbf{\Phi}^{-1}\hat{\mathbf{x}}_{p_k} \tag{9}$$

where $\hat{\mathbf{x}}_{p_k} = \{\mathbf{x}_{c_k}, \mathbf{x}_{q_k}\}$ refers to the coordinates of physical quadrature points, consisting of the element vertices $\mathbf{x}_{c_k}$ and additional surface points $\mathbf{x}_{q_k}$ for the element k, and $\mathbf{\Phi}$ denotes the projection mapping matrix, which is constituted by the basis functions evaluated at the aforementioned quadrature points ($\hat{\boldsymbol{\xi}}_{p_k} \leftarrow \hat{\mathbf{x}}_{p_k}$) in the reference triangle. The evaluation of the volume integrals in Eq. (5) is computed by use of Gaussian quadrature rules [39], which are exact for polynomial degree $2p$, and the surface integrals use Gauss–Legendre–Lobatto quadrature rules [37], which are exact for polynomial degree of $2p + 1$ [4,29].

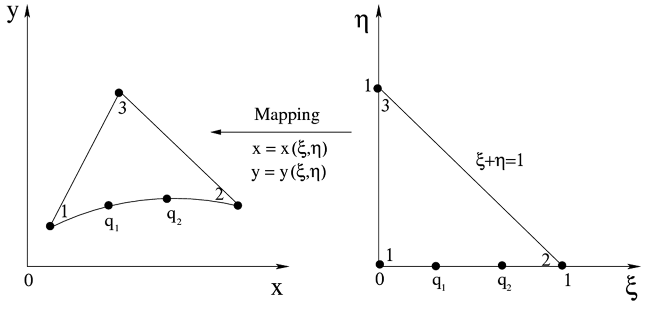


Fig. 1 Geometric mapping between a physical curvilinear triangle and the reference triangle.

Equation (5) can be written in the following ordinary differential equation (ODE) form as

$$\mathbf{M}\frac{\mathrm{d}\tilde{\mathbf{u}}_p}{\mathrm{d}t} + \mathbf{R}(\tilde{\mathbf{u}}_p) = 0 \tag{10}$$

where $\mathbf{R}$ represents discretized spatial residual and $\mathbf{M}$ denotes the mass matrix [29]. Next, we proceed to integrate in time using implicit temporal schemes.

B. Temporal Discretizations

To avoid the restriction of time-step sizes of explicit schemes, we make exclusive use of implicit temporal schemes, consisting of a BDF2 and an IRK4 [29,40]. The formulations for the BDF2 and IRK4 schemes can be derived by starting from the set of ODEs given by Eq. (10), written as follows:

BDF2:

$$\mathbf{R}_e^{n+1}(\tilde{\mathbf{u}}_p^{n+1}) = \frac{\mathbf{M}}{\Delta t}\left(\frac{3}{2}\tilde{\mathbf{u}}_p^{n+1}\right) + \mathbf{R}(\tilde{\mathbf{u}}_p^{n+1}) - \frac{\mathbf{M}}{\Delta t}\left(2\tilde{\mathbf{u}}_p^{n} - \frac{1}{2}\tilde{\mathbf{u}}_p^{n-1}\right) = 0 \tag{11}$$

IRK4:

$$\mathbf{R}_e^{n+1}(\tilde{\mathbf{u}}_p^{(s),n+1}) = \frac{\mathbf{M}}{\Delta t}\tilde{\mathbf{u}}_p^{(s),n+1} + a_{ss}\mathbf{R}(\tilde{\mathbf{u}}_p^{(s),n+1}) - \left[\frac{\mathbf{M}}{\Delta t}\tilde{\mathbf{u}}_p^{n} - \sum_{j=1}^{s-1} a_{sj}\mathbf{R}(\tilde{\mathbf{u}}_p^{(j),n+1})\right] = 0 \qquad (s = 1,\ldots,6) \tag{12}$$

where $\mathbf{R}_e^{n+1}$ represents the unsteady flow residual at time step $n + 1$. The current work uses a six-stage, fourth-order-accurate ESDIRK (explicit singly diagonally implicit Runge–Kutta) scheme, and the values of the set of coefficients a_{sj} can be found in [29].

Since one or multiple nonlinear problems are required to be solved in each implicit time step of the respective BDF2 scheme or the IRK4 scheme, in order to make these implicit schemes competitive, we employ efficient solution strategies including a nonlinear p-multigrid approach driven by a linearized Gauss–Seidel smoother. Details about the solution methods can be found in [29,40].

III. Adjoint-Based Unsteady Shape Optimization

The adjoint sensitivity derivation starts with the formulation of the forward linearization problem, in which the discretized governing equations [cf. Eq. (11) or Eq. (12)] are linearized and the sensitivity derivatives of an objective functional are formulated. The discrete adjoint formulation is then derived by transposing each matrix of the forward linearization problem and performing the operations in a reverse order. Here, we first provide a preliminary description of the shape-optimization techniques, and then we provide a detailed derivation for the calculation of sensitivity derivatives in the context of high-order discontinuous Galerkin discretizations on meshes involving curved boundary elements.

A. Preliminary Descriptions

1. Design Variables and Shape Parametrization

Assume that an initial shape geometry, represented by the coordinates of boundary nodes $\mathbf{x}_s$ and possibly additional surface quadrature points $\mathbf{x}_q$ for curved elements, and a computational mesh are given. The surface geometry is modified through surface node displacements that are determined by a set of design variables, $\mathbf{D} = \{D_m, m = 1,\ldots,N_d\}$ (where N_d represents the total number of design variables). The Hicks–Henne sine bump function [41] is employed in order to ensure smooth surface shapes, and the design variables are set to be the magnitudes of the bump functions placed at the surface nodes, expressed as

$$b_i(x_{si}, D_m) = D_m \sin^4\left(\pi x_{si}^{\ln\frac{1}{2}/\ln x_{sm}}\right), \qquad x_{si}, x_{sm} \in [0,1] \tag{13}$$

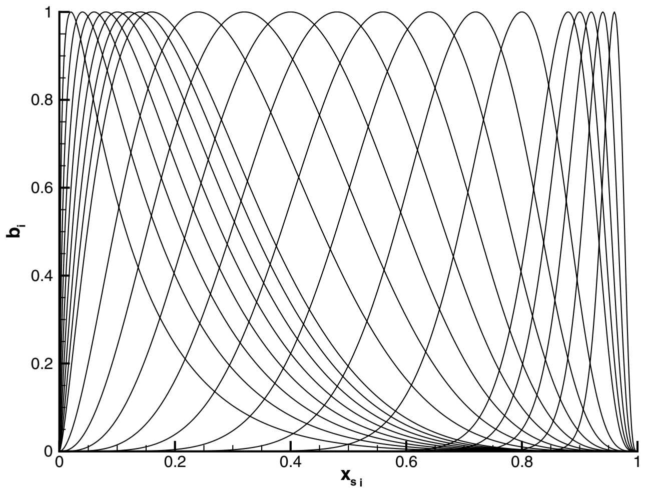


Fig. 2 Illustration of the Hicks–Henne bump functions for various x_{sm} parameters varying from 0.02 to 0.96.

where x_{sm} denotes the x coordinate of the surface node where the bump function is placed, b_i denotes the surface node displacement at x_{si} due to the displacement of the surface node at x_{sm}, and D_m is the mth component of the design variables associated with the surface node at x_{sm}. Figure 2 depicts a sample plot of the bump functions with a magnitude of 1, as x_{sm} takes on a range of 0.02 to 0.96. For multiple design variables ($N_d > 1$), the surface displacement at a particular node location of x_{si} is the superposition of all the bump functions placed at each designed surface node, and new surface coordinates are computed based on all the surface displacements, denoted as

$$\Delta \mathbf{x}_{si} = \mathbf{n}_{si} \sum_{m=1}^{N_d} b_i(x_{si}, D_m) \qquad \mathbf{x}_{si}^{\text{new}} = \mathbf{x}_{si}^{\text{old}} + \Delta \mathbf{x}_{si} \tag{14}$$

where $\Delta \mathbf{x}_{si}$ represents the displacement at the surface node i; $\mathbf{n}_{si} = (n_{six}, n_{siy})$ denotes the normal direction at the surface node i; and $\mathbf{x}_{si}^{(\text{new})}$ and $\mathbf{x}_{si}^{(\text{old})}$ represent, respectively, the new surface coordinates and the old ones from the initial design step. It is also noted that in many cases of aerodynamic airfoil design, surface deformation may occur only in the y-coordinate direction rather than in normal direction, since it may be desirable to keep the chord length of the designed airfoil fixed. Then the resulting displacement direction vector denoted by $\mathbf{n}_{si}$ in Eq. (14) is simply replaced by $(0, 1 \cdot \text{sign}(n_{siy}))$, and thus the surface displacements in the x-coordinate direction vanish.

2. *Deformation of Additional Quadrature Points for Surface Shape*

As described in Sec. II, the integrals in Eq. (5) are evaluated in the physical triangle but the basis set is defined in the reference triangle; thus, a reference-to-physical coordinate transformation is required, as illustrated in Fig. 1. Additional quadrature points are required to determine higher-order geometric mapping modes $\tilde{\mathbf{x}}_s$ for curved surface elements [cf. Eq. (9)], which must be employed in the presence of higher-order solutions ($p \geq 1$). Moreover, surface quadrature points must deform with surface nodes to define new surface mappings. This is achieved by using the same smooth bump functions used for the surface node displacements, shown as

$$\Delta \mathbf{x}_{qi} = \mathbf{n}_{qi} \sum_{m=1}^{N_d} b_i(x_{qi}, D_m) \qquad \mathbf{x}_{qi}^{\text{new}} = \mathbf{x}_{qi}^{\text{old}} + \Delta \mathbf{x}_{qi} \tag{15}$$

where $\Delta \mathbf{x}_{qi}$ represents the displacement at the quadrature point i; $\mathbf{n}_{qi} = (n_{qi_x}, n_{qi_y})$ denotes its normal direction obtained by the initial surface geometry [however, it may be replaced by $(0, 1 \cdot \text{sign}(n_{qiy}))$ to remain consistent with the surface node deformation]; and $\mathbf{x}_{qi}^{(\text{new})}$ and $\mathbf{x}_{qi}^{(\text{old})}$ represent, respectively, the new surface quadrature coordinates for quadrature node i and the old ones from the initial design step. Therefore, as illustrated in Fig. 3, the displacements at the entire set of surface quadrature points are performed similarly to those occurring at the surface grid points.

3. *Interior Mesh Deformation*

Once the deformed surface mesh is obtained, the interior mesh is deformed to prevent generation of overlapping elements. Here, we employ the linear tension spring analogy [42], in which each edge of the mesh is represented by a spring whose stiffness is related to the length of the edge. The governing equations are expressed as

$$[K]\Delta \mathbf{x} = \Delta \mathbf{x}_s \tag{16}$$

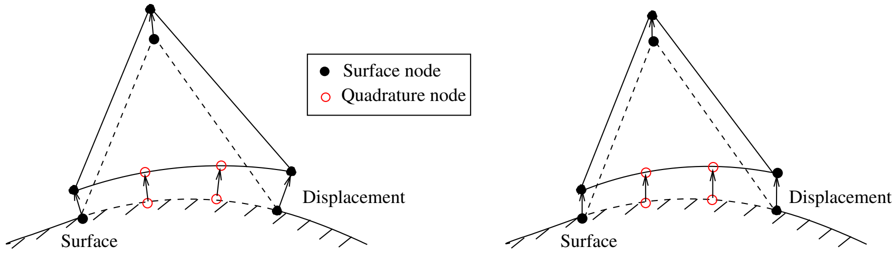
where $\Delta \mathbf{x}$ and $\Delta \mathbf{x}_s$ denote the entire mesh displacements and the surface mesh point displacements, respectively, and $[K]$ denotes the stiffness matrix obtained from the discrete mesh motion equations. This set of equations is solved using several hundred sweeps of a Gauss–Seidel scheme, since these equations are inexpensive to solve due to the relatively coarse meshes that are generally employed with high-order discretizations. Given the deformation of surface grid points, extra surface quadrature points and the interior mesh points, the corresponding coordinate mapping modal coefficients, $\tilde{\mathbf{x}}$, for the reference-to-physical mapping can be obtained. Note that the interior mesh contains only straight-sided, isotropic elements (due to the inviscid nature of the problem being considered), and the linear tension spring method is found to be capable of avoiding generation of negative-Jacobian elements if moderate displacements are performed in the surface deformation. Additionally, the geometric mapping of the interior linear elements can be simply determined by the use of element vertex coordinates.

4. *Unsteady Flow Equations*

The discretized implicit flow equations on the deformed mesh are expressed as

$$\mathbf{R}_e(\tilde{\mathbf{u}}(\tilde{\mathbf{x}}), \tilde{\mathbf{u}}^0(\tilde{\mathbf{x}}), \mathbf{u}^b(\tilde{\mathbf{x}}), \tilde{\mathbf{x}}) = 0 \tag{17}$$

where $\mathbf{R}_e = \{\mathbf{R}_e^n, n = 1, 2, \ldots, N\}$ represents the unsteady residual over all time steps; $\tilde{\mathbf{u}} = \{\tilde{\mathbf{u}}^n, n = 1, 2, \ldots, N\}$ represents the unsteady flow solution obtained by employing either the BDF2 scheme or the IRK4 scheme (N denotes the total number of time steps and the absorption of the subscript p is for simplicity); and $\tilde{\mathbf{u}}^0$ and $\mathbf{u}^b$ denote the initial condition and the unsteady freestream boundary



a) Normal direction displacements **b) Y-coordinate direction displacements**

Fig. 3 Illustration of deformation of curved surface elements with additional quadrature points. The dashed line represents the initial element shape and the solid line represents the deformed element shape.

condition, which may be explicitly dependent on the mesh geometry, such as in the case of the periodic gust-response problem discussed in Sec. IV. Since the evaluation of the volume and surface integrals of the discretized governing equations (5) requires the reference-to-physical transformation denoted by the coordinate mapping modal coefficients [in Eq. (8)], the discretized unsteady residual is shown as a function of $\tilde{\mathbf{x}}$ rather than $\mathbf{x}$, with the understanding that the mesh configuration (and thus $\tilde{\mathbf{x}}$) is held fixed throughout the flow time-integration marching and only changes between design iterations.

5. *Objective Functional*

Consider a time-dependent scalar-valued functional L, which refers to the objective functional for an optimization problem and is related to the time-dependent flowfield variables, such as time-integrated drag coefficients or target unsteady pressure distributions, etc. A general formulation for the objective functional is expressed as

$$L = L(\tilde{\mathbf{x}}(\mathbf{D}), \tilde{\mathbf{u}}(\tilde{\mathbf{x}}(\mathbf{D}))) \tag{18}$$

where $\tilde{\mathbf{u}}$ denotes the computed unsteady flow solution and $\mathbf{D}$ represents a set of design variables that produce changes in the surface mesh, as described previously. This functional expression is written in terms of the coordinate modal coefficients rather than grid node coordinates for the reason described above. It is also shown that a variation in any component of $\mathbf{D}$ not only produces changes in the element geometric mappings, but also produces changes in the time-dependent flowfield variables, which together contribute to a resulting variation in the objective functional.

B. Adjoint-Based Sensitivity Calculation

The derivation of the unsteady discrete adjoint technique for sensitivity analysis starts with the forward linearization problem by taking the derivatives of Eq. (18) with respect to the design variables via the chain rule, which leads to the following expression for the sensitivity derivatives:

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}} + \frac{\partial L}{\partial \tilde{\mathbf{u}}}\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{\mathbf{x}}}\right)\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{x}_s}\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right) \tag{19}$$

We next examine each term in the above equation, where $\partial \mathbf{x}_s/\partial \mathbf{D}$ and $\partial \mathbf{x}_q/\partial \mathbf{D}$ refer to the sensitivity of surface mesh points and surface quadrature points with respect to the design variables, which are obtained by linearizing the definition of the bump functions as they appear in Eqs. (14) and (15), respectively, and $\partial \mathbf{x}/\partial \mathbf{x}_s$ represents the sensitivity of the entire set of mesh points with respect to surface node displacements, which is evaluated based on the mesh deformation Eq. (16) as

$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}_s} = [K]^{-1} \tag{20}$$

where $[K]^{-1}$ represents the inverse of the mesh stiffness matrix. We note that the sensitivities $\partial \mathbf{x}/\partial \mathbf{x}_s$ rely purely on the grid node information; thus, the stiffness matrix has the same formulation as the one derived in finite volume methods [42]. Because of the fact that the geometric mapping coefficients $\tilde{\mathbf{x}}$ are determined by both mesh points $\mathbf{x}$ and extra surface quadrature points $\mathbf{x}_q$, $\partial \tilde{\mathbf{x}}/\partial \mathbf{x}$ and $\partial \tilde{\mathbf{x}}/\partial \mathbf{x}_q$ provide the sensitivity of the coordinate mapping coefficients with respect to the entire set of mesh point displacements and surface quadrature point displacements, respectively. Moreover, the formulations in fact correspond to the relative entries of the inverse projection mapping matrix, $\boldsymbol{\Phi}^{-1}$ defined in Eq. (9) in an elementwise manner. It is also noted that $\partial \tilde{\mathbf{x}}/\partial \mathbf{x}_q$ has nonzero components only for physical boundary elements, where high-order curved elements are allowed to occur. The flow sensitivities due to perturbations of the geometric mapping coefficients $\tilde{\mathbf{x}}$ (arising from the mesh deformation) are denoted by $\partial \tilde{\mathbf{u}}/\partial \tilde{\mathbf{x}}$ and can be formulated by linearizing Eq. (17) with respect to the design variables via the chain rule, as

$$\begin{gathered}\left(\left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}^0}\right]\frac{\partial \tilde{\mathbf{u}}^0}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}_e}{\partial \mathbf{u}^b}\right]\frac{\partial \mathbf{u}^b}{\partial \tilde{\mathbf{x}}} + \frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{x}}}\right)\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{D}} = 0 \quad \text{or} \\ \frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{\mathbf{x}}} = -\left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]^{-1}\frac{\partial \bar{\mathbf{R}}_e}{\partial \tilde{\mathbf{x}}}\end{gathered} \tag{21}$$

where

$$\frac{\partial \bar{\mathbf{R}}_e}{\partial \tilde{\mathbf{x}}} = \frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}^0}\right]\frac{\partial \tilde{\mathbf{u}}^0}{\partial \tilde{\mathbf{x}}} + \left[\frac{\partial \mathbf{R}_e}{\partial \mathbf{u}^b}\right]\frac{\partial \mathbf{u}^b}{\partial \tilde{\mathbf{x}}} \tag{22}$$

Note that the sensitivity of the unsteady flow residual with respect to the flow initial condition has contributions only from $[\partial \mathbf{R}_e^1/\partial \tilde{\mathbf{u}}^0]$ and $[\partial \mathbf{R}_e^2/\partial \tilde{\mathbf{u}}^0]$ in the BDF2 scheme and from $[\partial \mathbf{R}_e^{(s),1}/\partial \tilde{\mathbf{u}}^0]$ for the intermediate stages of the first time step in the IRK4 scheme. However, the last two terms in Eq. (22) vanish if uniform initial and freestream boundary conditions are given. Substituting Eqs. (20) and (21) into Eq. (19) yields the expression for the gradient sensitivities, shown as

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}} - \frac{\partial L}{\partial \tilde{\mathbf{u}}}\left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]^{-1}\frac{\partial \bar{\mathbf{R}}_e}{\partial \tilde{\mathbf{x}}}\right)\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}[K]^{-1}\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right) \tag{23}$$

The discrete adjoint problem is then formulated by transposing both sides of Eq. (23), which yields

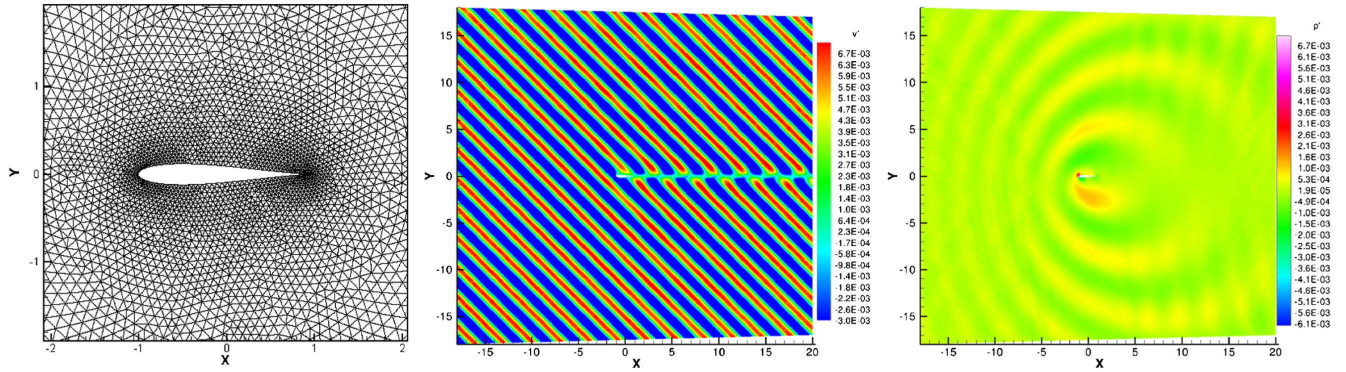
$$\begin{aligned}\left(\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}\right)^T = {}& \left[\left(\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}\right)^T [K]^{-T}\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}\right)^T + \left(\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)^T\left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\right)^T\right]\left[\left(\frac{\partial L}{\partial \tilde{\mathbf{x}}}\right)^T \right. \\ & \left. - \left(\frac{\partial \bar{\mathbf{R}}_e}{\partial \tilde{\mathbf{x}}}\right)^T\left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]^{-T}\left(\frac{\partial L}{\partial \tilde{\mathbf{u}}}\right)^T\right]\end{aligned} \tag{24}$$

where $[\cdot]^{-T}$ denotes the inverse of the transposed matrix. Equation (24) corresponds to the formulation for evaluating the discrete adjoint-based sensitivity derivatives in the context of discontinuous Galerkin discretizations. Some notable differences between the above formulation for DG methods and the corresponding adjoint formulation [16] in finite volume methods can be observed: First, both surface grid points and surface quadrature points are deformed simultaneously based on the bump functions; the consequent sensitivity terms relevant to the design variables in this formulation include both $(\partial \mathbf{x}_s/\partial \mathbf{D})^T$ and $(\partial \mathbf{x}_q/\partial \mathbf{D})^T$. Second, the sensitivities of the coordinate mapping, $(\partial \tilde{\mathbf{x}}/\partial \mathbf{x})^T$ and $(\partial \tilde{\mathbf{x}}/\partial \mathbf{x}_q)^T$, are required to be evaluated accordingly. As discussed previously, these terms represent the effect of the entire set of mesh points and the surface quadrature points on the geometric mapping coefficients. Third, the sensitivities denoted in the second bracket of the right-hand side of the equation, such as $[\partial \mathbf{R}_e/\partial \tilde{\mathbf{x}}]^T$ or $(\partial L/\partial \tilde{\mathbf{x}})^T$, must be evaluated with respect to the corresponding modal coefficients of geometric mappings in the DG discretizations, as opposed to the direct use of grid coordinates in finite volume methods.

We note that the term $[\partial \mathbf{R}_e/\partial \tilde{\mathbf{u}}]$ in Eq. (24) corresponds to the Jacobian matrix of the full unsteady residual. Since a direct solve for the inverse of the full Jacobian matrix or its transpose can be very expensive, the unsteady flow-adjoint variables, $\boldsymbol{\lambda}_u$, are introduced by replacing the last two terms, satisfying

$$\left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]^{-T}\left(\frac{\partial L}{\partial \tilde{\mathbf{u}}}\right)^T = \boldsymbol{\lambda}_u \quad \text{or} \quad \left[\frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{u}}}\right]^T \boldsymbol{\lambda}_u = \left(\frac{\partial L}{\partial \tilde{\mathbf{u}}}\right)^T \tag{25}$$

Thus, the transpose of the Jacobian of the discrete unsteady flow equations is used in the definition of the flow-adjoint variables, evaluated using the computed unsteady flow states, $\{\tilde{\mathbf{u}}^n\}$. Therefore, the unsteady flow-adjoint solution corresponds to a series of linear problems spanning the entire time domain, expressed as $\{\boldsymbol{\lambda}_u^n\}$. Details about the solution procedure of the unsteady flow-adjoint problems for the implicit temporal schemes employed in this work are discussed in the next section. Substituting the unsteady flow-adjoint variables $\boldsymbol{\lambda}_u$ into Eq. (24) yields



a) Enlarged view of the unstructured triangular mesh

b) Y-component velocity disturbance

c) Pressure disturbance

Fig. 4 Computational mesh (with 13,739 unstructured triangular elements) and instantaneous solution contours for a two-dimensional gust over a symmetric Joukowski airfoil with a reduced frequency of $k_1 = 2$, using a fourth-order spatial DG discretization and the implicit BDF2 temporal scheme.

$$\left(\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}\right)^T = \left(\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}\right)^T [K]^{-T} \left(\frac{\partial \bar{L}}{\partial \mathbf{x}}\right)^T + \left(\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)^T \left(\frac{\partial \bar{L}}{\partial \mathbf{x}_q}\right)^T \tag{26}$$

where

$$\left(\frac{\partial \bar{L}}{\partial \mathbf{x}}\right)^T = \left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}}\right)^T \left(\frac{\partial \bar{L}}{\partial \tilde{\mathbf{x}}}\right)^T \tag{27}$$

$$\left(\frac{\partial \bar{L}}{\partial \mathbf{x}_q}\right)^T = \left(\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}_q}\right)^T \left(\frac{\partial \bar{L}}{\partial \tilde{\mathbf{x}}}\right)^T \tag{28}$$

$$\left(\frac{\partial \bar{L}}{\partial \tilde{\mathbf{x}}}\right)^T = \left(\frac{\partial L}{\partial \tilde{\mathbf{x}}}\right)^T - \left(\frac{\partial \bar{\mathbf{R}}_e}{\partial \tilde{\mathbf{x}}}\right)^T \boldsymbol{\lambda}_u \tag{29}$$

where $(\partial \bar{L}/\partial \mathbf{x})^T$ denotes objective sensitivities with respect to the mesh deformation and $(\partial \bar{L}/\partial \mathbf{x}_q)^T$ denotes objective sensitivities with respect to the surface quadrature point deformation. Note that the second term in Eq. (29) implies the summation of the multiplication spanning the entire time domain.

Returning to Eq. (26), a similar approach is taken to avoid a direct solve for the inverse of the transposed mesh stiffness matrix $[K]^{-T}$ by introducing the mesh-adjoint variables $\boldsymbol{\lambda}_x$, satisfying

$$[K]^{-T} \left(\frac{\partial \bar{L}}{\partial \mathbf{x}}\right)^T = \boldsymbol{\lambda}_x \quad \text{or} \quad [K]^T \boldsymbol{\lambda}_x = \left(\frac{\partial \bar{L}}{\partial \mathbf{x}}\right)^T \tag{30}$$

The mesh-adjoint vector is a size of $d \times N_n$ (d and N_n refer to the spatial dimension and the total number of mesh points, respectively), and it corresponds to the solution of a single linear problem, which is distinct from the series of time-dependent flow-adjoint solutions discussed previously. Substituting Eq. (30) into Eq. (26), the final expression for the sensitivity derivatives is then shown as

$$\left(\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}}\right)^T = \left(\frac{\partial \mathbf{x}_s}{\partial \mathbf{D}}\right)^T \boldsymbol{\lambda}_x + \left(\frac{\partial \mathbf{x}_q}{\partial \mathbf{D}}\right)^T \left(\frac{\partial \bar{L}}{\partial \mathbf{x}_q}\right)^T \tag{31}$$

or

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{D}} = \boldsymbol{\lambda}_x^T \frac{\partial \mathbf{x}_s}{\partial \mathbf{D}} + \frac{\partial \bar{L}}{\partial \mathbf{x}_q} \frac{\partial \mathbf{x}_q}{\partial \mathbf{D}} \tag{32}$$

Since the terms relevant to the design variables [i.e., $(\partial \mathbf{x}_s/\partial \mathbf{D})^T$ and $(\partial \mathbf{x}_q/\partial \mathbf{D})^T$] are evaluated at the last step in this formulation, the evaluation of the adjoint-based sensitivity derivatives is essentially independent of the number of design variables, which makes the adjoint method well suited for cases with large numbers of design variables.

C. Unsteady Flow-Adjoint Formulation

In this section, we focus on the derivation of the unsteady flow-adjoint formulation denoted in Eq. (25) for the fourth-order implicit Runge–Kutta scheme, and we refer readers to [40,43] for a general backward-difference temporal scheme.

Because of the specific ESDIRK class of RK schemes employed in the current work, the flow solution at the first stage of each time step is in fact equal to the solution at the final stage of the previous time step, denoted as $\tilde{\mathbf{u}}^{(1),n} = \tilde{\mathbf{u}}^{(6),n-1} = \tilde{\mathbf{u}}^{n-1}$, where the first bracketed superscript represents an intermediate-stage index and the second denotes the time-step index. Then the vectors of the unsteady flow solution and the full unsteady residual for the IRK4 scheme denoted in Eq. (12) can be expressed, respectively, as

$$\tilde{\mathbf{u}} = \{\tilde{\mathbf{u}}^{(s),n}, s = 2, 3, \ldots, 6;\ n = 1, 2, \ldots, N\}$$

and

$$\mathbf{R}_e = \{\mathbf{R}_e^{(s),n}, s = 2, 3, \ldots, 6;\ n = 1, 2, \ldots, N\}$$

where N denotes the total number of time steps. In this manner, we write the unsteady flow-adjoint formulation denoted in Eq. (25) for a simplified time domain that involves only two steps ($N = 2$) in Eq. (33), with the understanding that the complete matrix for arbitrary time domains ($N > 2$) follows a similar form:

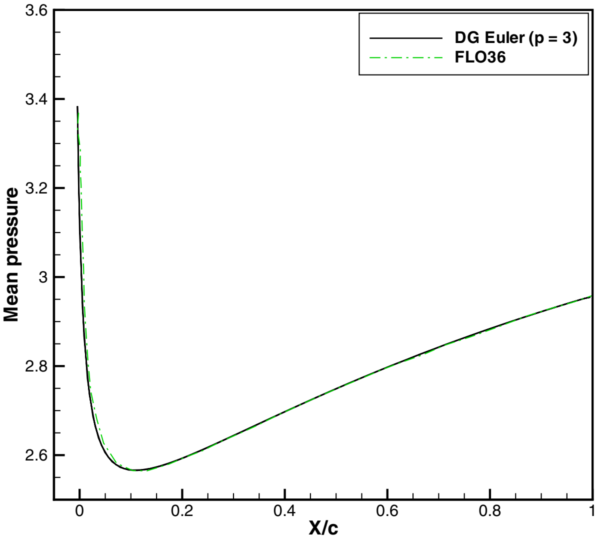


Fig. 5 Comparison of DG-Euler solution with potential flow solution from FLO36 [47] for the mean pressure distribution on the symmetric Joukowski airfoil in the gust-response test case.

$$
\begin{bmatrix}
\left[\frac{\partial \mathbf{R}_e^{(2),1}}{\partial \tilde{\mathbf{u}}^{(2),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(3),1}}{\partial \tilde{\mathbf{u}}^{(2),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(4),1}}{\partial \tilde{\mathbf{u}}^{(2),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),1}}{\partial \tilde{\mathbf{u}}^{(2),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),1}}{\partial \tilde{\mathbf{u}}^{(2),1}}\right]^T & 0 & 0 & 0 & 0 & 0 \\
0 & \left[\frac{\partial \mathbf{R}_e^{(3),1}}{\partial \tilde{\mathbf{u}}^{(3),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(4),1}}{\partial \tilde{\mathbf{u}}^{(3),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),1}}{\partial \tilde{\mathbf{u}}^{(3),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),1}}{\partial \tilde{\mathbf{u}}^{(3),1}}\right]^T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(4),1}}{\partial \tilde{\mathbf{u}}^{(4),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),1}}{\partial \tilde{\mathbf{u}}^{(4),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),1}}{\partial \tilde{\mathbf{u}}^{(4),1}}\right]^T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(5),1}}{\partial \tilde{\mathbf{u}}^{(5),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),1}}{\partial \tilde{\mathbf{u}}^{(5),1}}\right]^T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(6),1}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(2),2}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(3),2}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(4),2}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),2}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(6),1}}\right]^T \\
0 & 0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(2),2}}{\partial \tilde{\mathbf{u}}^{(2),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(3),2}}{\partial \tilde{\mathbf{u}}^{(2),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(4),2}}{\partial \tilde{\mathbf{u}}^{(2),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),2}}{\partial \tilde{\mathbf{u}}^{(2),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(2),2}}\right]^T \\
0 & 0 & 0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(3),2}}{\partial \tilde{\mathbf{u}}^{(3),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(4),2}}{\partial \tilde{\mathbf{u}}^{(3),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),2}}{\partial \tilde{\mathbf{u}}^{(3),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(3),2}}\right]^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(4),2}}{\partial \tilde{\mathbf{u}}^{(4),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(5),2}}{\partial \tilde{\mathbf{u}}^{(4),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(4),2}}\right]^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(5),2}}{\partial \tilde{\mathbf{u}}^{(5),2}}\right]^T & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(5),2}}\right]^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left[\frac{\partial \mathbf{R}_e^{(6),2}}{\partial \tilde{\mathbf{u}}^{(6),2}}\right]^T
\end{bmatrix}_{\tilde{\mathbf{u}}}
\begin{bmatrix}
\boldsymbol{\lambda}_u^{(2),1} \\ \boldsymbol{\lambda}_u^{(3),1} \\ \boldsymbol{\lambda}_u^{(4),1} \\ \boldsymbol{\lambda}_u^{(5),1} \\ \boldsymbol{\lambda}_u^{(6),1} \\ \boldsymbol{\lambda}_u^{(2),2} \\ \boldsymbol{\lambda}_u^{(3),2} \\ \boldsymbol{\lambda}_u^{(4),2} \\ \boldsymbol{\lambda}_u^{(5),2} \\ \boldsymbol{\lambda}_u^{(6),2}
\end{bmatrix}
=
\begin{bmatrix}
\frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(2),1}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(3),1}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(4),1}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(5),1}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(6),1}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(2),2}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(3),2}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(4),2}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(5),2}} \\ \frac{\partial L^T}{\partial \tilde{\mathbf{u}}^{(6),2}}
\end{bmatrix}_{\tilde{\mathbf{u}}}
\tag{33}
$$

Note that the calculation of the unsteady adjoint solution in the IRK4 scheme involves multistage computations at each implicit time step; moreover, since the lower triangular part of the transpose of the full unsteady Jacobian matrix, $[\partial \mathbf{R}_e/\partial \tilde{\mathbf{u}}]_{\tilde{\mathbf{u}}}^T$, contains only zero entries, the computation of the adjoint solution at an intermediate stage s of the time step n requires the adjoint solution at later stages, which leads to a backward time integration. In particular, the adjoint problem at the final time step N is first solved by

$$
\left[\frac{\partial \mathbf{R}_e^{(6),N}}{\partial \tilde{\mathbf{u}}^{(6),N}}\right]^T \boldsymbol{\lambda}_u^{(6),N} = \frac{\partial L}{\partial \tilde{\mathbf{u}}^{(6),N}}^T \tag{34}
$$

Then the adjoint solution at other intermediate stages in the final time step can be evaluated following the backward-recursion relation as

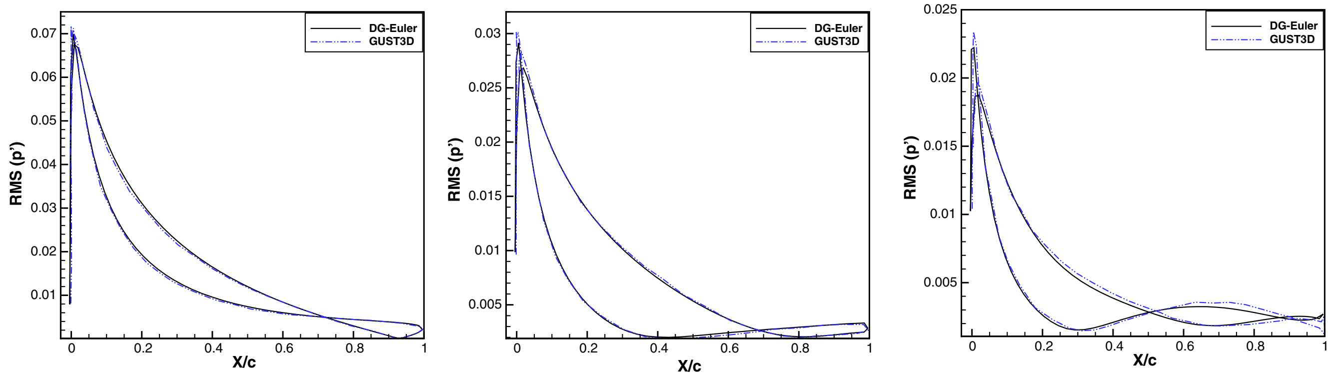
$$
\left[\frac{\partial \mathbf{R}_e^{(s),N}}{\partial \tilde{\mathbf{u}}^{(s),N}}\right]^T \boldsymbol{\lambda}_u^{(s),N} = \frac{\partial L}{\partial \tilde{\mathbf{u}}^{(s),N}}^T - \sum_{m=s+1}^{6} \left[\frac{\partial \mathbf{R}_e^{(m),N}}{\partial \tilde{\mathbf{u}}^{(s),N}}\right]^T \boldsymbol{\lambda}_u^{(m),N} \quad (s = 5, 4, \cdots, 2) \tag{35}
$$

For earlier time steps $n = N-1, N-2, \ldots, 1$, the multistage adjoint solution in the IRK4 scheme is computed based on the following formulations:

$$
\left[\frac{\partial \mathbf{R}_e^{(6),n}}{\partial \tilde{\mathbf{u}}^{(6),n}}\right]^T \boldsymbol{\lambda}_u^{(6),n} = \frac{\partial L}{\partial \tilde{\mathbf{u}}^{(6),n}}^T - \sum_{m=2}^{6} \left[\frac{\partial \mathbf{R}_e^{(m),n+1}}{\partial \tilde{\mathbf{u}}^{(6),n}}\right]^T \boldsymbol{\lambda}_u^{(m),n+1} \tag{36}
$$

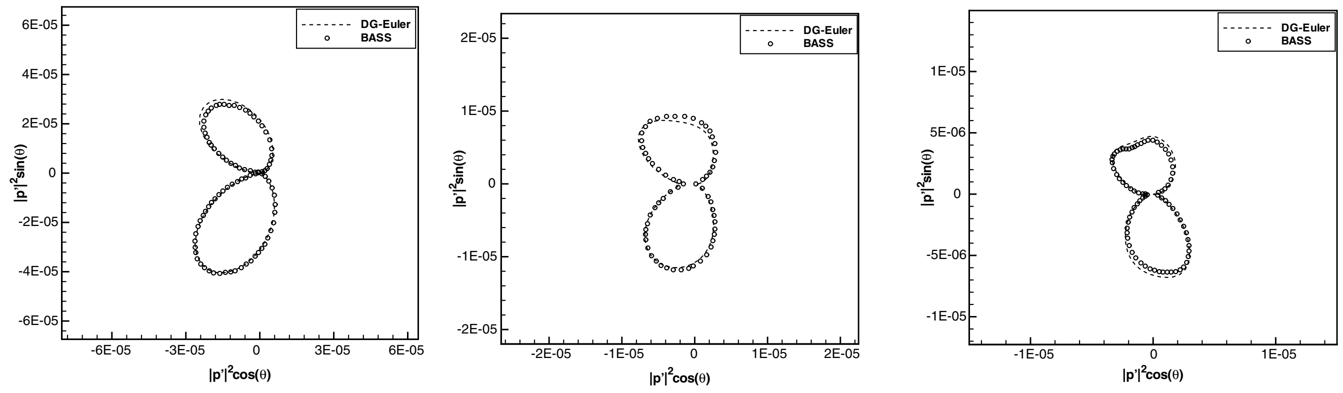
$$
\left[\frac{\partial \mathbf{R}_e^{(s),n}}{\partial \tilde{\mathbf{u}}^{(s),n}}\right]^T \boldsymbol{\lambda}_u^{(s),n} = \frac{\partial L}{\partial \tilde{\mathbf{u}}^{(s),n}}^T - \sum_{m=s+1}^{6} \left[\frac{\partial \mathbf{R}_e^{(m),n}}{\partial \tilde{\mathbf{u}}^{(s),n}}\right]^T \boldsymbol{\lambda}_u^{(m),n} \quad (s = 5, 4, \ldots, 2) \tag{37}
$$

where $(\partial L/\partial \tilde{\mathbf{u}}^{(s),n})^T$ shows the dependence of the selected objective functional on an intermediate-stage flow solution ($s < 6$), which turns out to be zero if the objective is a function of the solutions at only the discrete time-step locations. Note that the right-hand side of the adjoint formulation in Eq. (34) for evaluating the adjoint problem at the final time step ($s = 6$, $n = N$) is different from that in Eq. (36) for the evaluation at an intermediate time step ($s = 6$, $n \neq N$). In addition, the flow Jacobian matrices [40] denoted in Eqs. (34–37) can



a) $k_1 = 0.1$ b) $k_1 = 1$ c) $k_1 = 2$

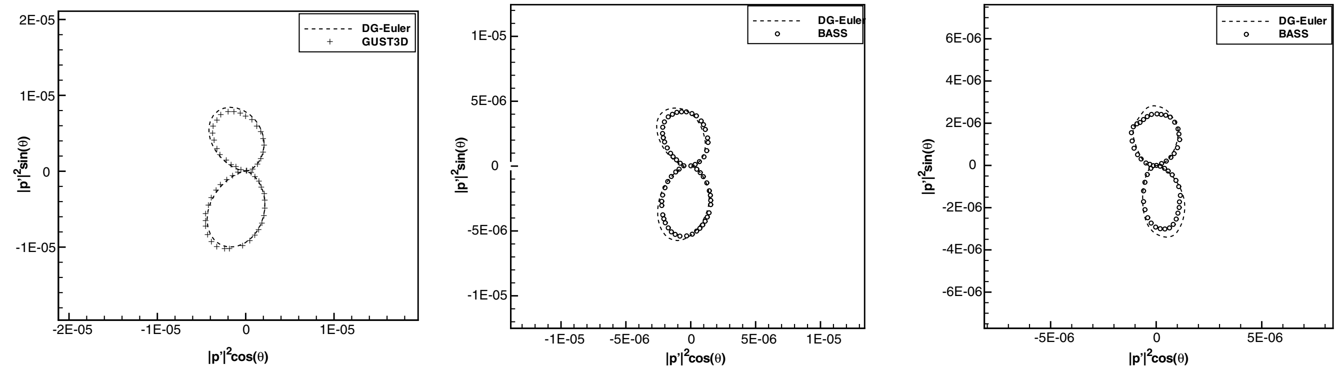
Fig. 6 Comparison of DG-Euler solution with linearized Euler solution GUST3D [30] for the rms pressure disturbance distribution on the symmetric Joukowski airfoil surface in the gust-response test case.



a) $\boldsymbol{R = 1c, k_1 = 0.1}$

b) $\boldsymbol{R = 1c, k_1 = 1}$

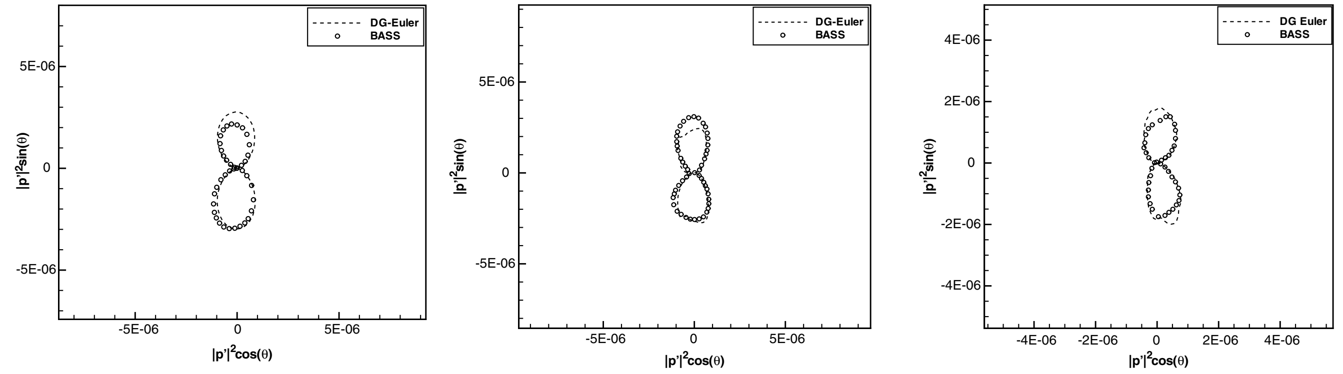
c) $\boldsymbol{R = 1c, k_1 = 2}$



d) $\boldsymbol{R = 2c, k_1 = 0.1}$

e) $\boldsymbol{R = 2c, k_1 = 1}$

f) $\boldsymbol{R = 2c, k_1 = 2}$



g) $\boldsymbol{R = 4c, k_1 = 0.1}$

h) $\boldsymbol{R = 4c, k_1 = 1}$

i) $\boldsymbol{R = 4c, k_1 = 2}$

Fig. 7 Comparison of DG-Euler solution with high-order nonlinear BASS [33] solution or linearized Euler solution GUST3D [30] for the acoustic intensity distribution at locations of $\boldsymbol{R = 1c}$, $\boldsymbol{R = 2c}$ and $\boldsymbol{R = 4c}$ in the gust-response test case with reduced frequencies of $\boldsymbol{k_1 = 0.1}$, $\boldsymbol{k_1 = 1}$ and $\boldsymbol{k_1 = 2}$.

be evaluated by linearizing Eq. (12) with respect to the appropriate flow state, and the solution strategy for solving each of the linear problems uses the p-multigrid approach [40] to accelerate the solution convergence. Given the flow-adjoint solution, the vector denoted by the multiplication $[\partial\bar{\mathbf{R}}_e/\partial\tilde{\mathbf{x}}]^T\boldsymbol{\lambda}_u$ in Eq. (29) is then calculated by the summation

$$\sum_{n=1}^{N}\sum_{s=2}^{6}\left[\frac{\partial\bar{\mathbf{R}}_e^{(s),n}}{\partial\tilde{\mathbf{x}}}\right]^T_{\tilde{\mathbf{u}}}\boldsymbol{\lambda}_u^{(s),n}$$

D. Shape-Optimization Procedure

Once the objective sensitivities have been evaluated, they are used to drive a design-optimization process to seek a minimum in a specified objective functional. It is well known that the efficiency of an optimization process relates closely to the particular optimization algorithm. While a gradient-based steepest descent method is simple to implement, this approach tends to deliver slow convergence when a large number of design variables are employed. The current work employs the PORT trust-region optimization strategy [44], which chooses a region around the current input variables at each iteration to obtain a new set of inputs. In addition, PORT requires the initial bounds to be specified for the set of design variables at the first design iteration, and the bounds are generally selected to be reasonable for the analysis solver and to avoid generation of collapsed shapes.

For a typical design-optimization iteration, five sequential steps are required:

1) Solve the unsteady flow equations using either the BDF2 scheme or the IRK4 scheme.

2) Solve the unsteady flow-adjoint variables and then the mesh-adjoint variables denoted in Eqs. (25) and (30), respectively.

3) Evaluate gradients or objective functional sensitivities $\mathrm{d}L/\mathrm{d}\mathbf{D}$ described in the previous section [cf. Eq. (31) or Eq. (32)].

4) Compute a new set of design variables, $\mathbf{D}^{\text{new}}$, using the PORT optimization algorithm based on the computed sensitivity derivatives.

5) Deform the geometry shape including additional surface points for curved elements using the new design variables as the magnitudes of the Hicks–Henne bump function and then deform the interior mesh based on the mesh motion equations. Geometric mapping coefficients are then reevaluated based on the newly deformed mesh.

This procedure is repeated until the objective functional is sufficiently minimized.

IV. Numerical Results

In this section, we first investigate the applicability of the current nonlinear implicit DG solver for a single-airfoil aeroacoustics gust-response problem [30] via comparisons of the unsteady surface pressure disturbances as well as acoustic intensity in the model problem with those obtained from NASA BASS and GUST3D computational aeroacoustics codes [30,33,45,46]. Next two numerical examples of unsteady airfoil shape optimization are presented to demonstrate the performance of the adjoint-based shape-optimization strategy, where the first aims to change the shape of the original NACA0012 airfoil to match a time-dependent surface pressure profile produced by a two-dimensional vortical gust impinging on a RAE-2822 airfoil, and the objective of the second is to minimize the acoustic noise generated by subsonic flow over a NACA0012 airfoil with a $0.03c$ thick blunt trailing edge.

A. Analysis Study for Single-Airfoil Gust-Response Problem

Consider a two-dimensional vortical gust that convects past a 12%-thick 0 deg angle-of-attack Joukowski airfoil with the gust velocity distribution in the upstream given by

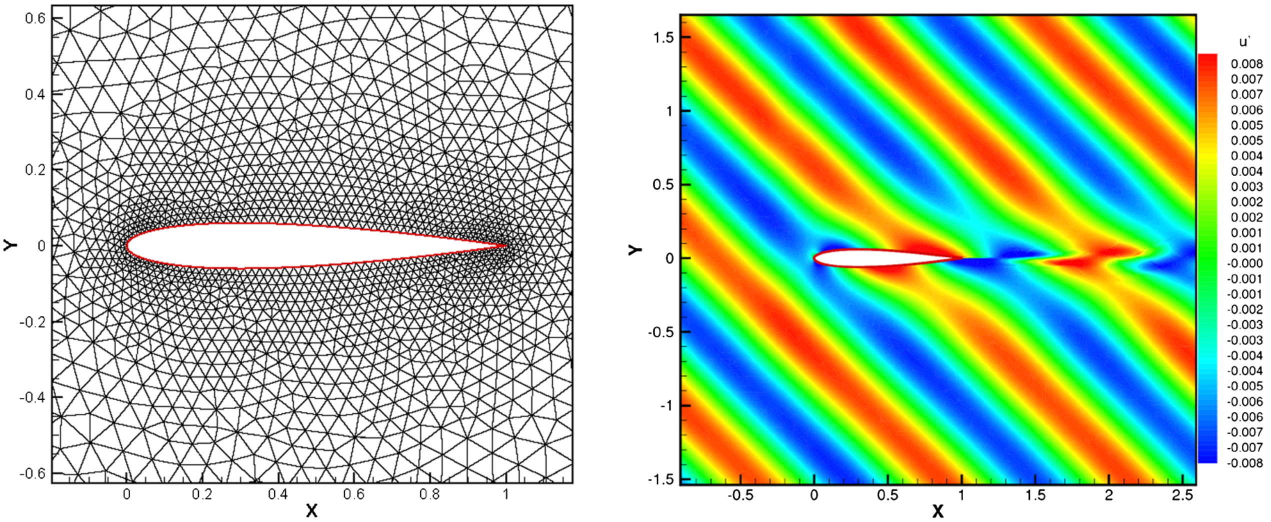
$$u_g = -(\varepsilon k_2 M_\infty a_\infty / \sqrt{k_1^2 + k_2^2}) \cos(\mathbf{k} \cdot \mathbf{x} - \omega t) \tag{38}$$

$$v_g = (\varepsilon k_1 M_\infty a_\infty / \sqrt{k_1^2 + k_2^2}) \cos(\mathbf{k} \cdot \mathbf{x} - \omega t) \tag{39}$$

where $\mathbf{k} = (k_1, k_2)$ denotes the wave number vector that specifies the direction of the gust propagation; ε is a small parameter satisfying $\varepsilon \ll 1$ and is set to 0.02; M_∞ and a_∞ denote the Mach number and speed of sound in the freestream, respectively; ω represents the gust angular frequency, which is equal to $2k_1 M_\infty a_\infty / c$, where c denotes the airfoil chord length; and k_1 is set equal to k_2 and denotes the reduced frequency. The mean flow parameters at infinity are specified as $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (\gamma, M_\infty, 0, 1)$, where $M_\infty = 0.5$.

A fourth-order-accurate (i.e., $p = 3$) DG discretization and the implicit BDF2 scheme with a time-step size of 0.001 are employed for the respective spatial and temporal discretizations. The HLLC approximate Riemann solver [34] is used for the flux function at all interior edges/boundaries and the particular wall boundary treatment described in Sec. II is employed at all wall/surface boundaries. The unsteady flowfield simulation starts with a steady-state solution in the absence of the gust and then the time-dependent gust problem is solved until the flow becomes periodic. The rms pressure disturbance $\sqrt{\overline{(p')^2}}$ on the airfoil surface and the acoustic intensity $\overline{(p')^2}$ at circular locations of $R = 1c$, $2c$, and $4c$ are calculated for three reduced frequencies consisting of $k_1 = 0.1$, 1, and 2, where the respective time steps to complete one period of the gust problem are 125,667, 12,566, and 6283 based on the fixed time-step size chosen in this case. Current results are compared with the GUST3D linearized Euler solution and the high-order nonlinear BASS solution documented in [30,33,46]. Before proceeding to the solution comparisons, it is worthwhile to emphasizing that the gust simulation in fact constitutes a challenge for obtaining a high-accuracy solution. The incoming gust, even with a small amplitude of disturbances, shows a strong interaction with the mean flow [30]. In addition, higher-order-accurate discretization methods or fine meshes (in sense of both space and time) are generally required even near the far-field boundaries to ensure that the small-amplitude gust can propagate accurately from the upstream to the airfoil. The near-field simulation, however, still encounters effects from the reflected acoustic waves generated in the finite outflow boundaries, leading to somewhat reduced accuracy in the overall gust simulation.

Figure 4a shows the computational mesh used in the gust problem, which contains 13,739 unstructured triangular elements. Figures 4b and 4c illustrate representative flowfield results for the instantaneous y component velocity disturbance and the instantaneous pressure disturbance in the case of a two-dimensional gust impinging on the symmetric Joukowski airfoil at a reduced frequency of $k_1 = 2$. It is seen that the periodic gust starts to disturb the flowfield from upstream and convects downstream. Figure 4c exhibits the qualitative features of the acoustic field noise, generated by the interaction of the gust with the Joukowski airfoil. The mean pressure distribution on the airfoil surface is plotted in Fig. 5 and is compared with the FLO36 potential flow solution [47] digitized from [46], in which good agreement can be observed. The corresponding rms pressure disturbances on the airfoil surface are shown in Fig. 6 for various reduced frequencies and are compared with the linearized



a) Original mesh and airfoil geometry (NACA0012) **b) Instantaneous x-component velocity**

Fig. 8 Plots of a) priginal computational mesh (with 4657 unstructured triangular elements) and the NACA0012 airfoil geometry and b) contours of the instantaneous x component velocity for a two-dimensional vorticity gust over the original NACA0012 airfoil with a reduced frequency of $k = 5$ at final time $T = 15$, using a fourth-order discontinuous Galerkin discretization and the implicit IRK4 scheme.

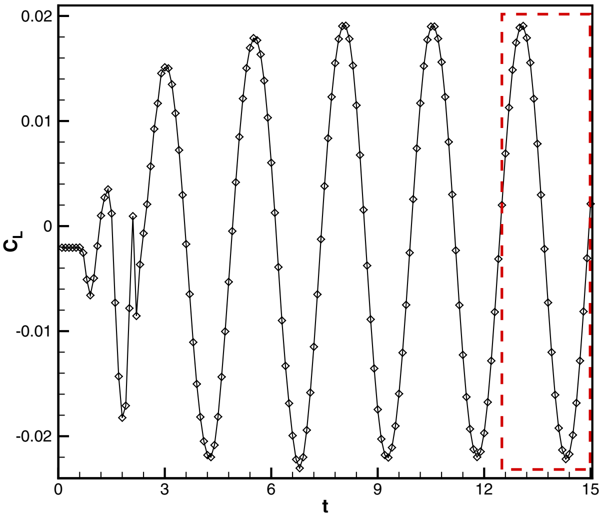


Fig. 9 Profile of time-dependent lift coefficient for the original NACA0012 airfoil in the inverse shape-optimization test case, where the dashed-line box indicates the time interval in which the objective functional is computed.

Euler solution GUST3D [30], where the agreement of the current DG-Euler solution with the GUST3D benchmark solution is very good. In Fig. 7, a series of computed results on acoustic intensity at different far-field locations away from the center point of the airfoil are compared with the high-order nonlinear BASS solution documented in [33,46], in which a prefactored six-order compact scheme and a fourth-order 5-6 low-dispersion and low-dissipation Runge–Kutta scheme [48] were used for the respective spatial and temporal discretizations. The corresponding result for the case of $R = 2c$ and $k_1 = 0.1$ is compared to the GUST3D solution, since the BASS solution was not available. Although a much finer C-grid (containing over 94,000 mesh points and more than twice the degrees of freedom in the presented scheme) was used in the BASS simulations, one can see that the current DG-Euler solver performs consistently well in the acoustic intensity comparison. Good agreement is obtained particularly for cases at these near-field locations (e.g., Figs. 7a–7f). Some discrepancies can be observed for cases at the circular locations of $R = 4c$, more evidently in the high-wave-number $k = 2$ case. The degradation in accuracy is likely caused by boundary reflections as the acoustic waves exit the finite computational domain. Current research is continuing on the development of a nonreflecting (or absorbing) boundary condition method to more effectively eliminate all possible reflection waves from outer boundaries.

B. Unsteady Shape Optimization for the Single-Airfoil Gust-Response Problem

The first design problem used to demonstrate the performance of the adjoint-based shape optimization considers the single-airfoil vorticity gust-response problem discussed previously, where the root-mean-square difference between pressure distributions on the target RAE-2822 airfoil configuration with the original NACA0012 airfoil configuration is selected as the objective functional, given by

$$L = \sqrt{\frac{\sum_{n=n_s}^{N} \sum_{j=1}^{N_s} \sum_{q=1}^{N_q} (p_{q,j}^n - (p_{q,j}^n)^*)^2}{N^* \cdot N_s \cdot N_q}} \tag{40}$$

where $p_{q,j}^n$ represents the pressure value obtained from the current airfoil configuration at the nth time step for quadrature point q at the airfoil surface edge j, and $(p_{q,j}^n)^*$ represents the pressure value for the target RAE-2822 airfoil configuration at the same location and time step. The respective n_s, N and N^* refer to the starting and ending time-step indices and the number of time steps in the calculation of the objective functional. N_s and N_q denote the total number of surface edges and quadrature points per edge, respectively. Since the target pressure distribution is a function of airfoil geometry curvature, the pressure values initially obtained from the target airfoil must be interpolated to obtain a set of pressure targets that correspond to the locations of the original airfoil in the functional definition for each time step. Furthermore, the objective functional of the rms difference is theoretically capable of reaching a smaller value as the interpolation error is decreased. In this test case, a cubic interpolation function is used to obtain the target pressure values according to the x coordinates of the quadrature points for the original NACA0012 computational mesh; thus, the displacements of surface grid points as well as additional surface quadrature points are only allowed in the y-coordinate direction for this interpolation approach. The design variables are set to be the magnitudes of the bump functions placed at surface grid points, ranging from 2 to 99% of the chord locations of the upper and lower airfoil surfaces. Once the objective functional and design variables are given, the sensitivity gradients can be evaluated using the discrete adjoint sensitivity technique described previously.

The initial geometry (NACA0012) and the computational mesh, which contains 4657 unstructured triangular elements and 117 design variables, are illustrated in Fig. 8a. A mean flow Mach number of 0.5 and a 0 deg angle of attack are specified for this design problem. The inflow vortical gust has a distribution as described

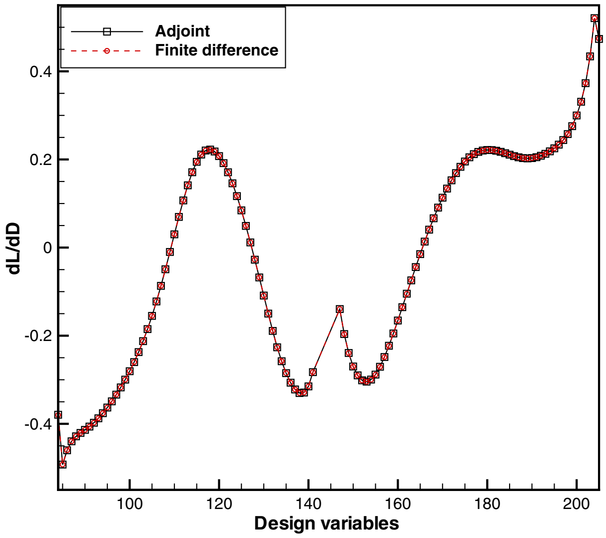


Fig. 10 Comparison of sensitivity derivatives using the unsteady discrete adjoint method and the finite difference method for the original mesh and airfoil geometry (NACA0012).

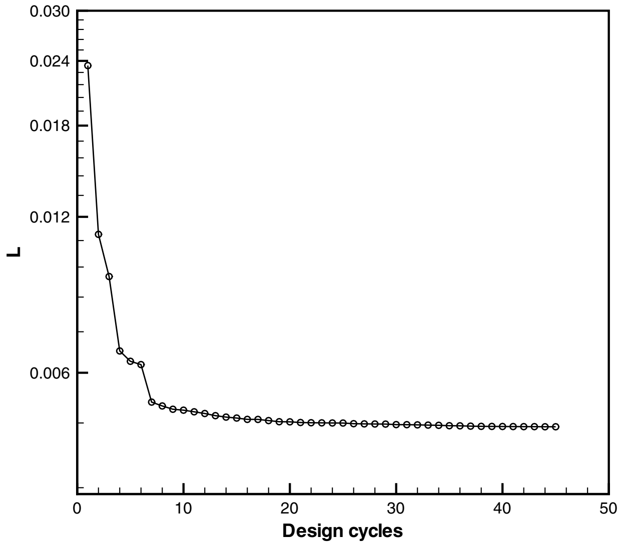
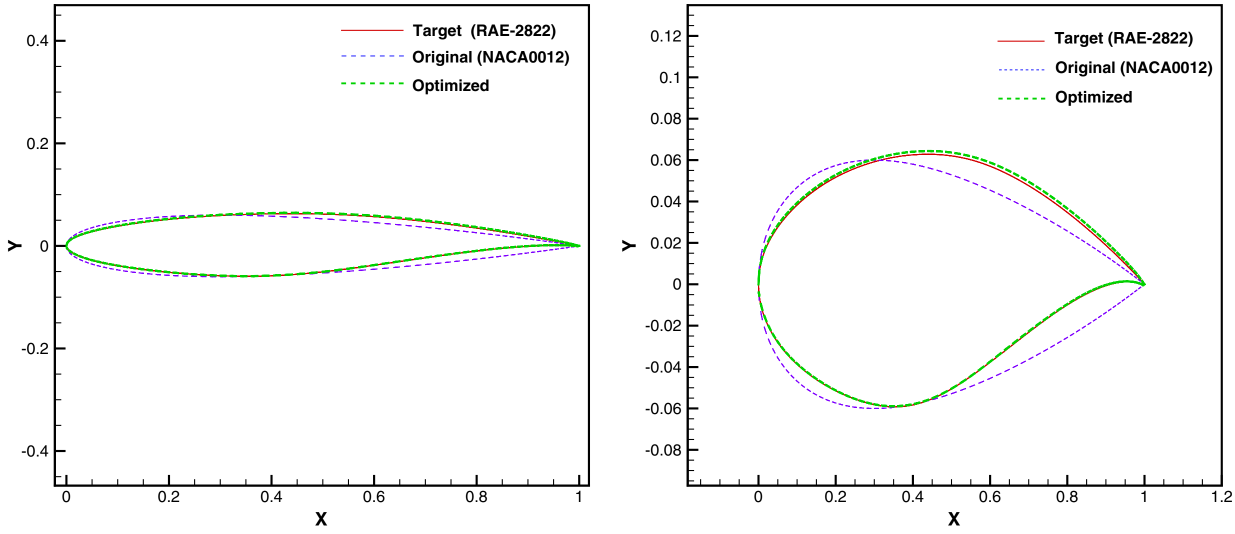


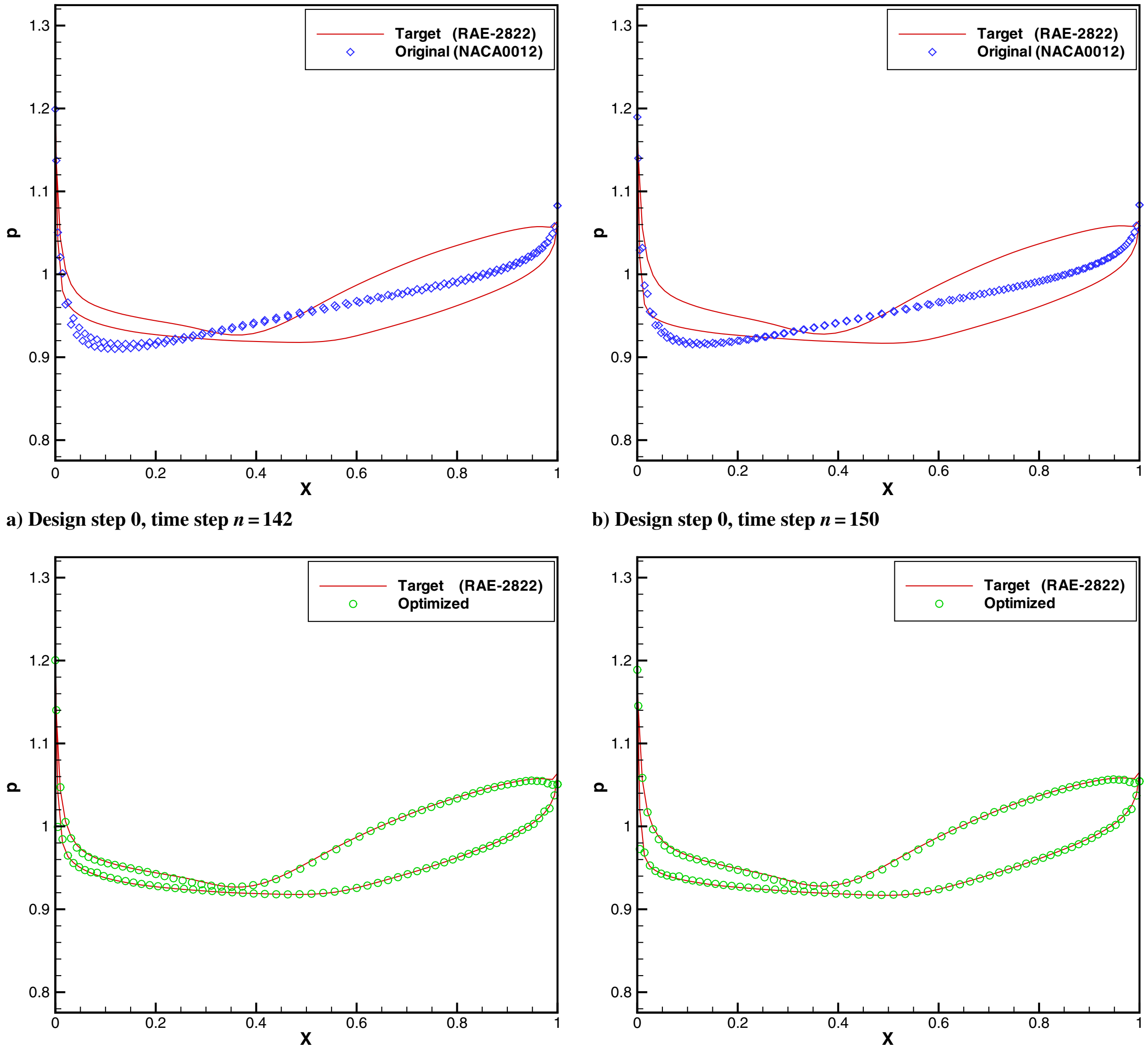
Fig. 11 Convergence of the objective functional in terms of the number of design iterations in the gust-response shape-optimization test case.



a) 1:1 scale

b) Exaggerated vertical scale

Fig. 12 Comparison of airfoil shapes among the target RAE-2822 airfoil, the original NACA0012 airfoil and the final optimized airfoil in the gust-response shape-optimization test case.



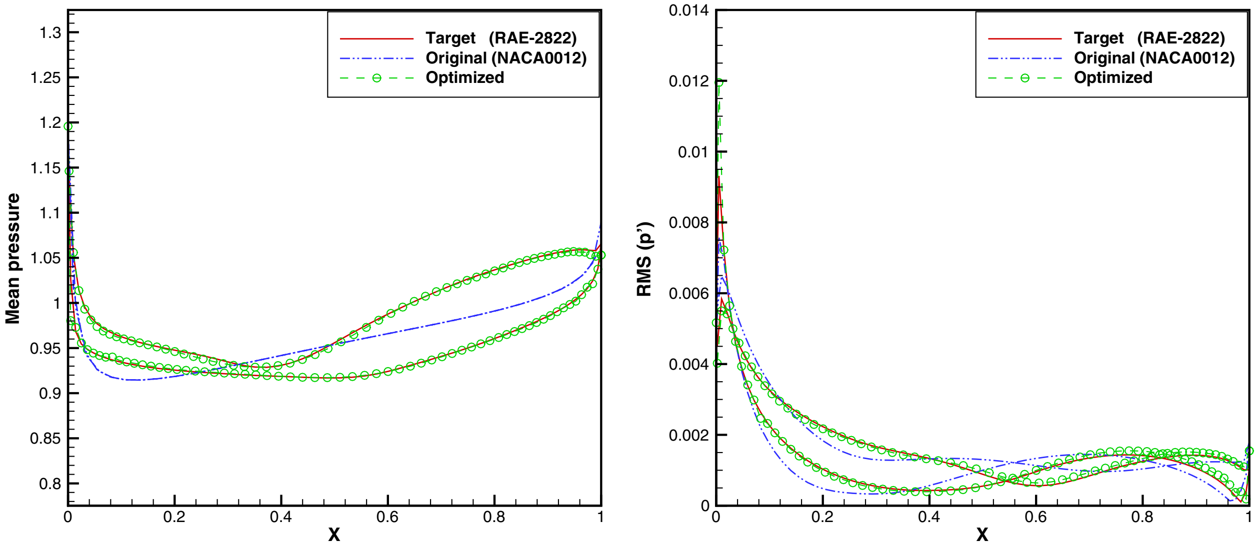
a) Design step 0, time step $n = 142$

b) Design step 0, time step $n = 150$

c) Design step 45, time step $n = 142$

d) Design step 45, time step $n = 150$

Fig. 13 Comparison of pressure distributions on airfoil surfaces at various time steps for the original, target and final optimized airfoils in the gust-response shape-optimization test case.



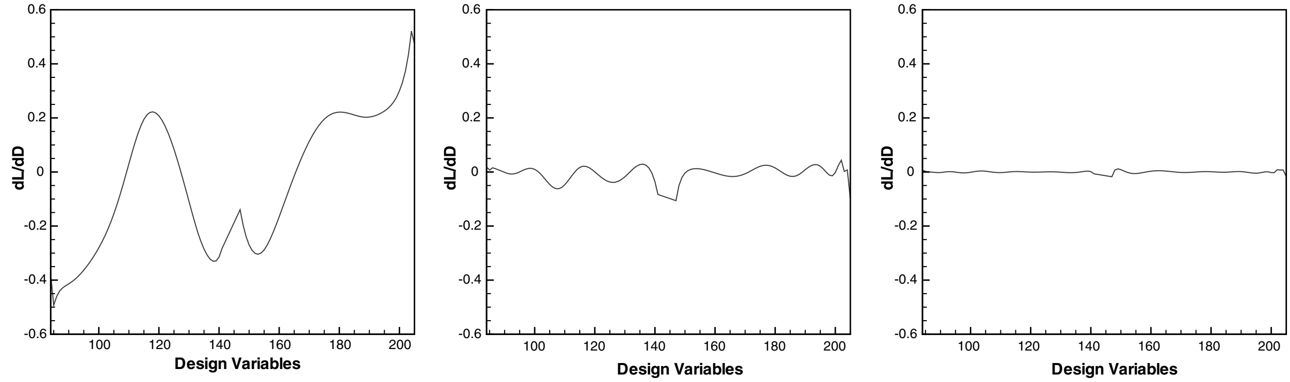
a) Mean pressure

b) RMS pressure disturbance

Fig. 14 Comparison of mean pressure distributions and rms pressure disturbances on the target (RAE-2822), the original (NACA0012) and the final optimized airfoil surfaces in the gust-response shape-optimization test case.

previously and we set the reduced frequency to be 5 in this case. A fourth-order (i.e., $p = 3$) spatial discontinuous Galerkin scheme and the IRK4 scheme are employed for the respective spatial and temporal discretizations. A steady-state solution for the original NACA0012 airfoil is solved in the absence of the gust and is used as the initial flow condition for the design-optimization problem. The time-step size and a total number of time steps employed are equal to 0.1 and 150, respectively. The flow and adjoint problems in the design procedure are solved using a p-multigrid approach [29] with 10 smoothing passes at each multigrid level. The convergence tolerance for the nonlinear flow problem and the linear adjoint problem is set to be 10^{-12}, which requires about 3–4 p-multigrid iterations at each stage of the IRK4 scheme in both problems. However, the linear nature of the adjoint problem often results in lower cost per iteration, compared with the flow problem. Specifically in this test case, the flow problem corresponds to 63% of the computational cost per design cycle, while 32% of the cost arises from solving the adjoint problem (and the rest of the linearization calculation only accounts for less than 5% of the total CPU cost). Figure 8b illustrates the instantaneous x component velocity disturbance at the final time $T = 15$ in the case of a vortical gust impinging on the original NACA0012 airfoil with a reduced frequency of $k_1 = 5$. Furthermore, in order to eliminate flow transient effects to the shape-optimization problem, only the last 25 time steps, which correspond to roughly one period of the periodic flow solution, are selected for the time interval in which the objective functional is computed. Figure 9 depicts the time-dependent lift-coefficient profile on the original airfoil configuration for the gust problem, where the dashed-line box indicates the actual objective time interval.

Before proceeding to the design-optimization procedure, the sensitivity vector computed using the proposed discrete adjoint procedure is verified by comparing with the finite differenced gradients, where the original airfoil geometry (NACA0012) and computational mesh are employed in this validation test. In the finite difference scheme, each component of the sensitivity vector (e.g., $\mathrm{d}L/\mathrm{d}D_i$) is obtained by perturbing the bump function placed at the designed surface point i, obtaining a new curved surface geometry and deformed computational mesh, and then recomputing the unsteady flow solution to obtain the change of the objective functional caused by the perturbation of D_i. This procedure is invoked for all design variables to obtain the entire sensitivity vector. Figure 10 illustrates comparison of the sensitivity values computed using the discrete adjoint approach (for the $p = 3$ DG discretization and the IRK4 temporal scheme) and the finite difference scheme using a small perturbation size of 10^{-6}. As the design variable index increases, the values shown in this figure are obtained from the original lower airfoil surface (starting from the trailing edge to the leading edge) and then from the upper airfoil surface (starting from the leading edge and ending at the trailing edge). It is shown that the

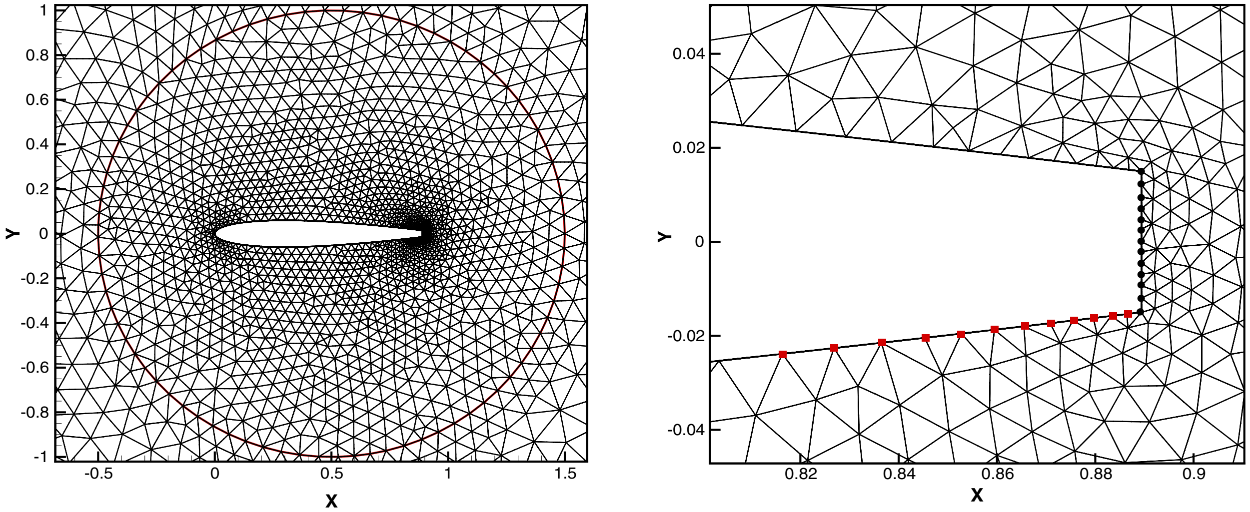


a) Design step 0

b) Design step 10

c) Design step 45

Fig. 15 Distribution of adjoint-based sensitivity derivatives at various design steps in the gust-response shape-optimization test case, where the gradient norms are 0.2516, 0.0264, and 0.0037, respectively.



a) Computational mesh **b) Enlarged view of the blunt trailing edge**

Fig. 16 Computational mesh and surface geometry of the initial NACA0012 airfoil with a $0.03c$ thick blunt trailing edge: a) circle indicates areas where pressure fluctuations are required to be minimized, and b) squares indicate locations of the design variables, while the dots indicate fixed grid points.

computed adjoint sensitivities provide an excellent match with the finite-differenced results; moreover, the average difference between the two approaches is within 0.03%.

Figure 11 plots the computed objective functional against the number of shape-optimization iterations, where it is shown that the objective functional decreases significantly in the first few design steps. The optimization was stopped after 45 design steps when the relative difference in the objective was less than 0.01%. However, an approximate one-order-of-magnitude reduction in the objective is still obtained at this point. Figure 12 compares the geometry shapes for the original, target and final optimized airfoils using a 1:1 scale as well as an exaggerated vertical scale to show details. It is evident that the target RAE-2822 airfoil shape differs considerably from the original symmetric NACA0012 airfoil, especially near the lower trailing edge. However, as illustrated in Figs. 12a and 12b, the consequent optimized shape matches the target considerably well and there are no visible discrepancies on the lower airfoil surfaces even in the exaggerated scale. Since the front and rear portions of the NACA0012 upper surface need to produce deformation in opposite directions to match the target shape, it is found that the convergence of the upper surface appears to be relatively slower than the lower surface during the optimization process; therefore, some minor difference between the upper airfoil surfaces can still be observed in Fig. 12b at this stage. In Fig. 13, a comparison of the time-dependent pressure profile on the original, target and optimized airfoils is provided for various time steps at the initial and final design steps. It is seen that the airfoil geometry has a strong effect on the airfoil unsteady pressure in the gust-response problem, however, the target unsteady profile is captured very accurately by the final optimized airfoil. Figure 14 exhibits comparison of the mean pressure and rms pressure disturbance distributions on these airfoils obtained from a one-period time integration. It is shown that the optimized airfoil consistently provides good agreement on the target mean pressure and rms pressure fluctuation profiles, while only a slight overprediction in the rms pressure disturbance is observed near the upper leading edge. Distributions of the adjoint-based sensitivity derivatives at various design steps are depicted in Fig. 15, in which the discontinuity in the sensitivity distributions at design variable indices 141 and 147 implies the transition from the lower airfoil surface to the upper surface. The peaks in the initial sensitivity distribution occur near the NACA0012 trailing edge, which indicates that a shape change in this region can mostly affect the reduction of the objective functional. The norm of the sensitivity derivatives is decreased by about 1 order of magnitude within only 10 design iterations due to the rapid objective convergence in the first few design steps, and a two-order-of-magnitude reduction in gradient norm is approximately obtained in 45 design iterations. As seen in Fig. 15c, the sensitivities are driven to be quite small, compared with the initial design stage. However, the small peaks occurring close to the upper leading and trailing edges (i.e., design indices 147 and 205) indicate that some displacements in surface shape would continue to be produced in these regions, which may further help to decrease the remaining discrepancies between the target and optimized upper surfaces observed previously in the shape comparison.

C. Airfoil Shape Design for Blunt-Trailing-Edge Acoustic Radiation

The second test case considers pressure fluctuation (noise) minimization for a blunt-trailing-edge flow [49]. Figure 16 illustrates the original computational mesh, which consists of 4942 elements and 100 surface points for a NACA0012 airfoil with a $0.03c$ ($H = 0.03$) thick blunt trailing edge and a chopped chord length of 0.8894. A freestream Mach number of 0.2 and a 0 deg angle of attack are specified in this case. Because of the sharp corners of the blunt trailing edge, vortices are generated, separate and are then convected downstream, producing pressure fluctuations in near and far fields from the airfoil body. In this case, we seek to minimize the pressure fluctuations (i.e., the difference between instantaneous pressure and mean pressure) at the one-chord length circular locations from the center point of the NACA0012 blunt-trailing-edge airfoil, as

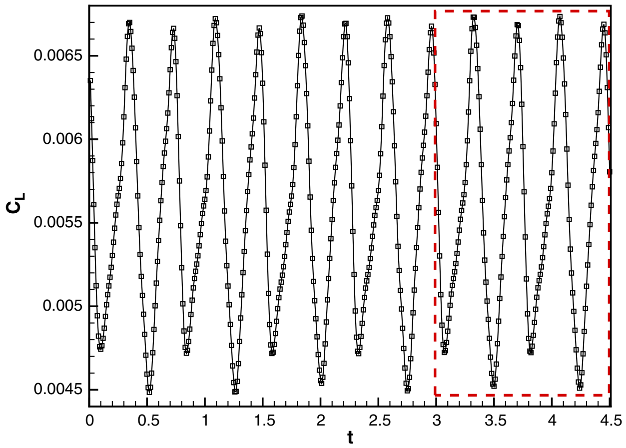
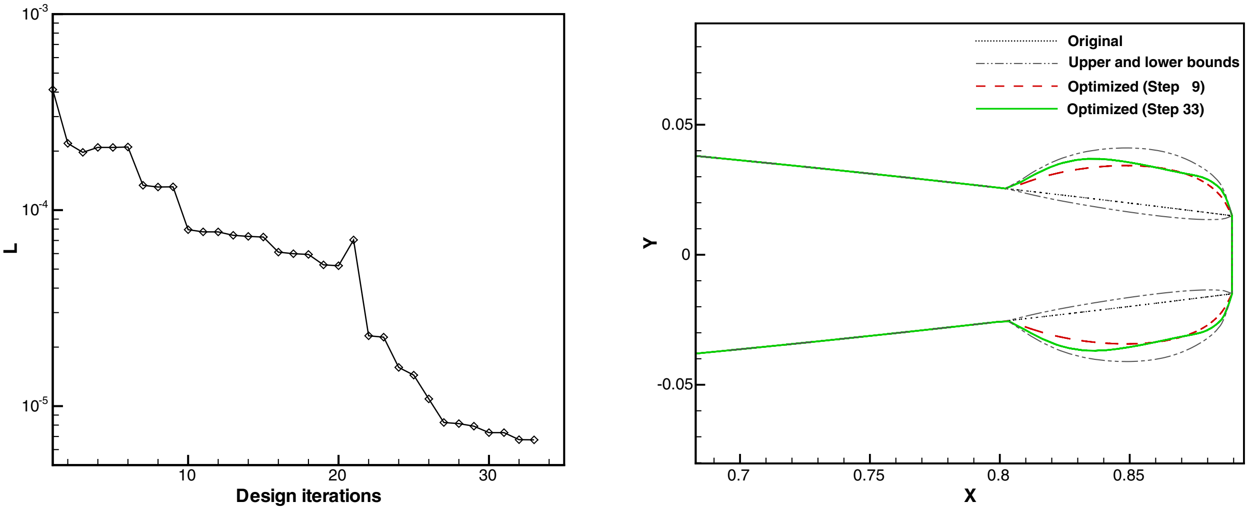


Fig. 17 Profile of time-dependent lift coefficient for subsonic flow over the initial NACA0012 airfoil with a $0.03c$ thick blunt trailing edge, where the dashed-line box indicates the objective time interval.



a) Objective convergence history

b) Design shapes

Fig. 18 Plots of a) convergence of the objective functional in terms of the number of design iterations for the blunt-trailing-edge flow and b) comparison of optimized shapes at various design steps with the original airfoil shape.

indicated by the bold circle in Fig. 16a. Therefore, the objective functional for this purpose is given by

$$L = \sqrt{\frac{\sum_{n=n_s}^{N} \sum_{q=1}^{N_p} (p_q^n - \bar{p}_q)^2}{N^* \cdot N_p}} \tag{41}$$

where N_p denotes the total number of grid points used to evaluate the noise generation at the one-chord length locations; n_s and N refer to the starting and ending time-step indices, respectively; p_q^n represents the unsteady pressure at grid point q and time step n; and $\bar{p}_q$ denotes the mean value of the unsteady pressure at the corresponding location, which is calculated as

$$\bar{p}_q = \left(\frac{1}{2} p_q^{n_s} + \sum_{n=n_s+1}^{N-1} p_q^n + \frac{1}{2} p_q^N \right) \Big/ T^* \tag{42}$$

where T^* denotes the time interval from time step n_s to time step N when pressure fluctuations are measured. In this test case, we first initialize the flow using the uniform flow condition in the far-field and then march the unsteady solution until it becomes periodic. Then a periodic flow solution is set as the initial condition to start the unsteady discrete adjoint-based design optimization. A time-step size of 0.01 and a total of 450 time steps are selected in the example, however, only the last 155 time steps, which correspond to approximately two periods of pressure fluctuations, are related to the time interval in which the pressure noise (i.e., the objective functional) is measured. This is because an adjustment time interval must be considered so that the pressure waves generated by the shedding vortices at the airfoil trailing edge could propagate through the one-chord circular locations specified. It is also noted that little or no shape change may be produced without a reasonable length of time-integration. Figure 17 depicts the time-dependent lift-coefficient profile for subsonic flow over the original NACA0012 blunt-trailing-edge airfoil and the dashed-line box in this figure indicates the actual time interval for the objective calculation.

A fourth-order (i.e., $p = 3$) discontinuous Galerkin scheme and the second-order BDF2 temporal scheme are employed for the respective spatial and temporal discretizations in this case. The Lax–Friedrichs approximate Riemann solver [35] is used for the flux function at interior edges/boundaries and the particular wall boundary treatment discussed in Sec. II is employed at wall/surface boundaries. The original symmetric airfoil geometry and the flow with 0 deg angle of attack lead to the generation of shedding vortices from the upper and lower trailing edges with a similar frequency. Thus, the present test case follows a consistent shape-deformation procedure throughout the design-optimization process in order to maintain a symmetric optimized airfoil for simplicity. This is achieved by setting the design variables only on the lower airfoil surface, as indicated by the square symbols in Fig. 16b, and the deformation of the mesh points as well as surface quadrature points on the upper airfoil surface is performed via the Hicks–Henne bump function and values of the design variables obtained from the lower airfoil configuration. In particular, 12 design variables are used and surface deformation is performed in the y-coordinate direction, while within 10% of the chord length from the trailing edge. Additionally, both flow and adjoint problems are solved using the p-multigrid approach, with a prescribed tolerance level of 10^{-10}. Similarly with the previous example, the computational cost in a typical design iteration mainly arises from the flow and adjoint solutions, however, the relative cost for the adjoint solver increases to 39% (as opposed to 32% in the previous example) and the flow solver corresponds to 57% of the total cost. The slight cost increase in the adjoint solver is due to the increased file input/output operations (for reading the flow states and writing the adjoint solution) as the number of time steps is increased.

The convergence history of the objective functional for the noise minimization design problem is presented in Fig. 18a. Because of the fact that the elements in regions near the upper and lower trailing edges tend to become very skewed if relatively large bounds are selected for the design variables, reasonable bounds must be specified initially to avoid failure of the flow solver or generation of collapsed surfaces. The optimization performance, however, is limited by the bounds, since the design variables still require more freedom in movements to further decrease the objective. Therefore,

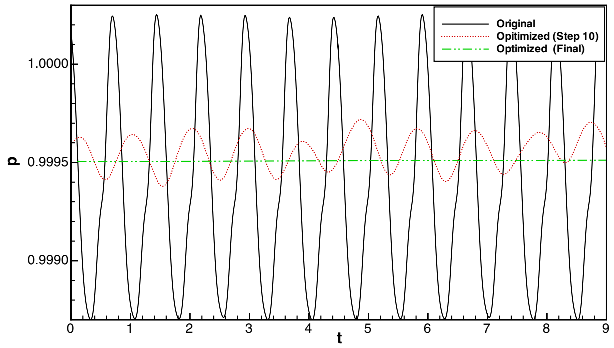
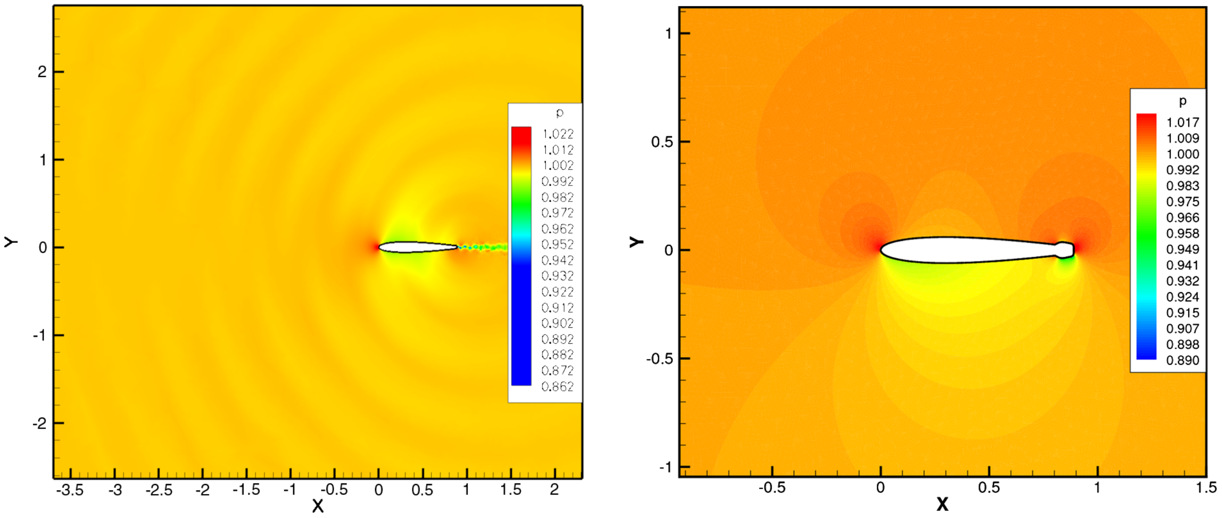


Fig. 19 Comparison of pressure signals at a location of one-chord length below the airfoil center point. (The pressure fluctuations produced by the optimized shapes are shifted close to the mean of the original pressure signals for comparison.)



a) Original airfoil **b) Final airfoil**

Fig. 20 Comparison of pressure solution contours for the original and final optimized airfoil geometries in the case of airfoil shape design for blunt-trailing-edge acoustic radiation.

in order to improve the quality of the deformed computational mesh, an edge-swapping procedure and a mesh-smoothing technique [12] are employed at the 10th design iteration, which enables the objective convergence to achieve about a one-order-of-magnitude reduction by the 20th design iteration. Additionally, while a periodic flow solution for the original airfoil geometry is used as an initial condition in the following design iterations to limit the overall computational effort, this flow solution cannot represent an accurate solution for the flowfield if there is any change in the computational mesh, hence an updated periodic solution is recomputed every 10 design iterations based on the corresponding working mesh and optimized geometry. As seen in Fig. 18a, this procedure introduces a spike at the 20th design iteration in the convergence history, but it contributes to a significant improvement in the objective convergence and a two-order-of-magnitude reduction in the objective is finally achieved in approximately 30 design steps, associated with a over two-order-of-magnitude reduction in gradient norms (reducing from 1.004 to 0.006). Figure 18b displays comparison of the optimized airfoil shapes at design iteration 9 and 33 (i.e., final) with the original airfoil geometry as well as the upper and lower bounds set in the optimizer. It is observed that the designed shapes attempt to increase the trailing-edge angles so as to eliminate the sharp corners of the original blunt trailing edge. Figure 19 illustrates the pressure fluctuations produced by the original airfoil and optimized airfoils at a location of $1c$ below the center point of the airfoil, where the pressure disturbance curves for the designed shapes are shifted close to the mean of the original pressure signature for comparison. One can clearly see that the generated noise by the original blunt-trailing-edge airfoil is reduced by about 70% after only 10 design steps; furthermore, the optimized shape at the final design step produces zero noise or pressure fluctuations at this location, since the corresponding flowfield finally reaches a steady-state condition. Figure 20 illustrates the pressure solution contours for the original and final optimized airfoil geometries, where it is shown that a steady-state solution is eventually achieved by the final designed airfoil shape. Finally, we point out that this test case is presented under the assumption of inviscid flow, since the flow and adjoint solvers are based on the Euler equations, and the viscous terms and the effects are not included. Therefore, the consequent shape of the optimized airfoil can vary when more realistic viscous flows or different flow conditions are considered.

V. Conclusions

This work presents a detailed derivation of an unsteady discrete adjoint formulation for high-order discontinuous Galerkin discretizations and investigates areas of unsteady aerodynamic shape optimization based on the developed discrete adjoint sensitivity techniques. The major computational cost at each design iteration arises from both the flow solution and the flow-adjoint solution, and the rest of the linearization calculation only accounts for about 5% of the computational effort. This implies that significant computational efficiency can be in fact gained using the developed adjoint techniques compared with a finite difference sensitivity calculation, where each design variable requires at least one additional flow analysis simulation. On the other hand, the flow-adjoint problem must be solved based on the same temporal scheme that is used in the flow marching scheme. The definition of the adjoint problem corresponds to a series of linear systems formulated by the transpose of the unsteady Jacobian matrix and is evaluated using the correct flow states. Therefore, in the adjoint-based sensitivity calculation, the unsteady flow solution to the primal problem is stored to disk (via direct access) so as to be accessible for the following adjoint solver. Furthermore, the evaluation of sensitivity derivatives requires linearization of geometric mapping coefficients with respect to both mesh points and extra surface quadrature points for high-order curved boundary elements in the discontinuous Galerkin discretizations. The present mesh deformation strategy involves a similar deformation method for the additional surface quadrature points as for the standard surface mesh points. This has been used successively herein for inviscid flow design problems, however, a more sophisticated method may be required for viscous flow design problems in which highly stretched and curved elements may be used in boundary-layer regions.

In addition, the present work also investigates capabilities of the current implicit DG solver for the aeroacoustics gust-response problem, in which accurate time-dependent flow solutions are required for both aerodynamics and aeroacoustics. Results show that the solutions of pressure disturbances on a symmetric Joukowski airfoil surface as well as acoustic intensity at the far-field locations provide good agreement with the results from other validated aeroacoustics codes. Future work will concentrate on the development of three-dimensional high-order-accurate DG aeroacoustics codes and on the extension of the current unsteady discrete adjoint strategy to more complex three-dimensional design problems.

Acknowledgments

The work of L. Wang and W. K. Anderson was supported by the Tennessee Higher Education Commission (THEC) Center of Excellence in Applied Computational Science and Engineering (CEACSE). D. Mavriplis was funded by NASA grant NX07AC31A and U.S. Air Force Office of Scientific Research contract FA9550-09-c-0021.

References

[1] Cockburn, B., and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, 1998, pp. 2440–2463. doi:10.1137/S0036142997316712

[2] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D., "p-Multigrid Solution of High-order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113. doi:10.1016/j.jcp.2005.01.005

[3] Luo, H., Baum, J. D., and Lohner, R., "A p-Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids," *Journal of Computational Physics*, Vol. 211, No. 2, 2006, pp. 767–783. doi:10.1016/j.jcp.2005.06.019

[4] Nastase, C. R., and Mavriplis, D. J., "High-Order Discontinuous Galerkin Methods Using an hp-Multigrid Approach," *Journal of Computational Physics*, Vol. 213, No. 1, 2006, pp. 330–357. doi:10.1016/j.jcp.2005.08.022

[5] Bassi, F., Crivellini, A., Pietro, D. A. D., and Rebay, S., "An Implicit High-Order Discontinuous Galerkin Method for Steady and Unsteady Incompressible Flows," *Computers and Fluids*, Vol. 36, No. 10, 2007, pp. 1529–1546. doi:10.1016/j.compfluid.2007.03.012

[6] Persson, P.-O., Bonet, J., and Peraire, J., "Discontinuous Galerkin Solution of the Navier-Stokes Equations on Deformable Domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, Nos. 17–20, 2009, pp. 1585–1595. doi:10.1016/j.cma.2009.01.012

[7] Estep, D., "A Posteriori Error Bounds and Global Error Control for Approximation of Ordinary Differential Equations," *SIAM Journal on Numerical Analysis*, Vol. 32, 1995, pp. 1–48. doi:10.1137/0732001

[8] Becker, R., and Rannacher, R., "An Optimal Control Approach to a Posteriori Error Estimation in Finite Element Methods," *Acta Numerica*, Vol. 10, 2001, pp. 1–102. doi:10.1017/S0962492901000010

[9] Houston, P., and Suli, E., "hp-Adaptive Discontinuous Galerkin Finite Element Methods for First-Order Hyperbolic Problems," *SIAM Journal on Scientific Computing*, Vol. 23, No. 4, 2001, pp. 1226–1252. doi:10.1137/S1064827500378799

[10] Hartmann, R., and Houston, P., "Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations," *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532. doi:10.1006/jcph.2002.7206

[11] Fidkowski, K. J., and Darmofal, D., "A Triangular Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 1653–1672. doi:10.1016/j.jcp.2007.02.007

[12] Wang, L., and Mavriplis, D. J., "Adjoint-Based h-p Adaptive Discontinuous Galerkin Methods for the 2D Compressible Euler Equations," *Journal of Computational Physics*, Vol. 228, No. 20, 2009, pp. 7643–7661. doi:10.1016/j.jcp.2009.07.012

[13] Mani, K., and Mavriplis, D. J., "Error Estimation and Adaptation for Functional Outputs in Time-Dependent Flow Problems," *Journal of Computational Physics*, Vol. 229, No. 2, 2010, pp. 415–440. doi:10.1016/j.jcp.2009.09.034

[14] Jameson, A., Alonso, J. J., Reuther, J. J., Martinelli, L., and Vassberg, J. C., "Aerodynamic Shape Optimization Techniques Based on Control Theory," AIAA Paper 1998–2538, 1998.

[15] Jameson, A., Martinelli, L., and Pierce, N., "Optimum aerodynamic design using the Navier-Stokes equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1, 1998, pp. 213–237. doi:10.1007/s001620050060

[16] Mavriplis, D. J., "A Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes," AIAA Paper 2006-0050, Jan. 2006.

[17] Mani, K., and Mavriplis, D. J., "An 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

[18] Elliott, J., and Peraire, J., "Practical 3D Aerodynamic Design and Optimization Using Unstructured Meshes," *AIAA Journal*, Vol. 35, 1997, pp. 1479–1485. doi:10.2514/2.271

[19] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 1997–0643, Jan. 1997.

[20] 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

[21] Collis, S. S., Chayour, K., and Heinkenschloss, M., "Optimal Transpiration Boundary Control for Aeroacoustics," *AIAA Journal*, Vol. 41, No. 7, 2003, pp. 1257–1270. doi:10.2514/2.2094

[22] Chen, G., and Collis, S. S., "Toward Optimal Control of Aeroacoustic Flows Using Discontinuous Galerkin Discretizations," AIAA Paper 2004-364, Jan. 2004.

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

[24] Rumpfkeil, M. P., and Zingg, D. W., "The Optimal Control of Unsteady Flows with a Discrete Adjoint Method," *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 5–22. doi:10.1007/s11081-008-9035-5

[25] Bassi, F., and Rebay, S., "High-Order-Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," *Journal of Computational Physics*, Vol. 138, 1997, pp. 251–285. doi:10.1006/jcph.1997.5454

[26] ven der Vegt, J., and ven der Ven, H., "Slip Flow Boundary Conditions in Discontinuous Galerkin Discretizations of the Euler Equations of Gas Dynamics," *Fifth World Congress on Computational Mechanics*, Vienna, Austria 7–12 July 2002, pp. 1–16.

[27] Cockburn, B., and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *SIAM Journal on Scientific Computing*, Vol. 16, 2001, pp. 173–261. doi:10.1023/A:1012873910884

[28] Kubatko, E. J., Dawson, C., and Westerink, J. J., "Time Step Restrictions for Runge-Kutta Discontinuous Galerkin Methods on Triangular Grids," *Journal of Computational Physics*, Vol. 227, No. 23, 2008, pp. 9697–9710. doi:10.1016/j.jcp.2008.07.026

[29] Wang, L., and Mavriplis, D. J., "Implicit Solution of the Unsteady Euler Equations for High-Order-Accurate Discontinuous Galerkin Discretizations," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 1994–2015. doi:10.1016/j.jcp.2007.03.002

[30] Scott, J. R., "Benchmark Solutions for Computational Aeroacoustics (CAA) Code Validation," NASA John H. Glenn Research Center at Lewis Field, TM-2004-213386, Cleveland, OH 2004.

[31] Scott, J. R., and Atassi, H. M., "A Finite Difference, Frequency-Domain Numerical Scheme for the Solution of the Gust Response Problem," *Journal of Computational Physics*, Vol. 119, No. 1, 1995, pp. 75–93. doi:10.1006/jcph.1995.1117

[32] Lockard, D. P., and Morris, P. J., "A Parallel Implementation of a Computational Aeroacoustic Algorithm for Airfoil Noise," *Journal of Computational Acoustics*, Vol. 5, No. 4, 1997, pp. 337–353. doi:10.1142/S0218396X97000198

[33] Hixon, R., Golubev, V., Mankbadi, R. R., Scott, J. R., Sawyer, S., and Nallasamy, M., "Application of a Nonlinear Computational Aeroacoustics Code to the Gust-Airfoil Problem," *AIAA Journal*, Vol. 44, No. 2, 2006, pp. 323–328. doi:10.2514/1.3478

[34] Batten, P., Clarke, N., Lambert, C., and Causon, D. M., "On the Choice of Wavespeeds for the HLLC Riemann Solver," *SIAM Journal on Scientific Computing*, Vol. 18, No. 6, 1997, pp. 1553–1570. doi:10.1137/S1064827593260140

[35] Shu, C.-W., "Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws," Inst. for Computer Applications in Science and Engineering Rept. 97-65, Hampton, VA, 1997.

[36] Lu, J. C.-C., "An A Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method," Ph.D. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, June 2005.

[37] Solin, P., Segeth, K., and Dolezel, I., *High-Order Finite Element Methods, Studies in Advanced Mathematics*, Chapman and Hall, Boca Raton, FL, 2003.

[38] Nastase, C. R., and Mavriplis, D. J., "A Parallel hp-Multigrid Solver for Three-Dimensional Discontinuous Galerkin Discretizations of the

Euler Equations," AIAA Paper 2007-0512, Jan. 2007.

[39] Dunavant, D. A., "High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 6, 1985, pp. 1129–1148.
doi:10.1002/nme.1620210612

[40] Wang, L., "Techniques for High-Order Adaptive Discontinuous Galerkin Discretizations in Fluid Dynamics," Ph.D. Dissertation, Department of Mechanical Engineering, Univ. of Wyoming, Laramie, WY, May 2009.

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

[42] Yang, Z., and Mavriplis, D. J., "Unstructured Dynamic Meshes with Higher-Order Time Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2005-1222, Jan. 2005.

[43] Mani, K., and Mavriplis, D. J., "Discrete Adjoint Based Time-Step Adaptation and Error Reduction in Unsteady Flow Problems," AIAA Paper 2007-3944, June 2007.

[44] Fox, P. A., Hall, A. D., and Schryer, N. L., "The PORT Mathematical Subroutine Library," *ACM Transactions on Mathematical Software*, Vol. 4, 1978, pp. 104–126.
doi:10.1145/355780.355783

[45] Hixon, R., Nallasamy, M., and Sawyer, S., "Parallelization Strategy for an Explicit Computational Aeroacoustics Code," AIAA Paper 2002-2583, July 2002.

[46] Golubev, V. V., Mankbadi, R. R., and Hixon, R., "Simulation of Airfoil Response to Impinging Gust Using High-Order Prefactored Compact Code," *Fourth Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, CP-2004-212954, NASA John H. Glenn Research Center at Lewis Field, Cleveland, OH, 2004.

[47] Jameson, A., and Caughey, D. A., "A Finite Volume Method for Transonic Potential Flow Calculations," AIAA Paper 1977-635, June 1977.

[48] Hixon, R., "Prefactored Small-Stencil Compact Schemes," *Journal of Computational Physics*, Vol. 165, 2000, pp. 522–541.
doi:10.1006/jcph.2000.6631

[49] Rumpfkeil, M. P., and Zingg, D. W., "A Hybrid Algorithm for Far-Field Noise Minimization," *Computers and Fluids*, Vol. 39, No. 9, 2010, pp. 1516–1528.
doi:10.1016/j.compfluid.2010.05.006.

Z. Wang
Associate Editor