Aerodynamic Shape Optimization for Unsteady Flows with Application to Laminar Flows

Kwesi P. Apponsah* and David W. Zingg †

University of Toronto, Institute for Aerospace Studies, Toronto, M3H 5T6, Ontario, Canada

An aerodynamic shape optimization framework for unsteady flow is applied to a range of two- and three-dimensional laminar flows. The shape optimization framework uses free-form deformation for geometry control with an underlying B-spline surface parameterization integrated with an efficient mesh deformation method. The mesh deformation is based on the linear elasticity method applied to a B-spline control volume parameterization of the mesh. A parallel implicit Newton-Krylov algorithm is used to solve the discretized flow equations and the discrete adjoint methodology is applied to both the flow and the mesh-movement algorithms to compute the gradient. For the two-dimensional studies, we consider three objectives based on the mean aerodynamic quantities: lift-constrained drag minimization, lift-to-drag ratio maximization, and lift maximization. For the drag minimization and lift-to-drag ratio maximization problems, the optimizer improved the performance of the baseline airfoil primarily by keeping the flow on the upper surface attached as long as possible and also pushing the camber towards the trailing edge to increase or maintain the lift coefficient. The optimizer improved the drag minimization objective by more than 20% and the lift-to-drag ratio maximization objective by about 50% for roughly the same initial drag. We also investigate the impact of design variable scaling on the convergence of the lift-maximization problem. For the three-dimensional studies, we consider a minimization of mean drag at a fixed mean lift, and we allow section shape, aerodynamic twist about the quarter-chord, and the chord length to vary along the span of the wing. The optimizer exploits all of the geometric freedom given to improve the design objective while satisfying the constraints imposed and produces some non-intuitive geometric changes, especially with respect to the wing planform.

I. Nomenclature

$\mathbf{B}$	=	mesh-deformation variables
C_l, C_L	=	lift coefficient
C_d, C_D	=	drag coefficient
c	=	chord
Δt	=	time step
$\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$	=	inviscid fluxes
$\mathbf{E_v}$, $\mathbf{F_v}$, $\mathbf{G_v}$	=	viscous fluxes
$\mathcal{F}$	=	mesh-deformation and flow residuals
$\mathcal{J}$	=	objective function
$\mathcal{L}$	=	Lagrangian function
m	=	number of mesh-deformation increments
$\mathcal{M}$	=	mesh-deformation residual
N	=	number of time steps
$\mathbf{Q}$	=	conserved flow variables
$\mathcal{R}$	=	flow residual
$\tilde{\mathcal{R}}$	=	temporal component of flow residual
$\hat{\mathcal{R}}$	=	spatial component of flow residual
$\mathbf{V}$	=	mesh-deformation and conserved flow variables

*PhD Candidate, AIAA Student Member (kwesi.apponsah@mail.utoronto.ca)

†University of Toronto Distinguished Professor of Computational Aerodynamics and Sustainable Aviation, Director, Centre for Research inSustainable Aviation, Director, Centre for Computational Science and Engineering, AIAA Associate Fellow (dwz@utias.utoronto.ca)

$\mathbf{X}$	=	design variables
x, y, z	=	Cartesian coordinates
ξ, η, ζ	=	curvilinear coordinates

II. Introduction

The field of computational fluid dynamics has matured over the last three decades due to the availability of powerful computers and efficient algorithms. Despite this maturity, computing unsteady flows still remains a challenge due to the long turnaround time for such problems. Driving an optimization algorithm with such unsteady flow simulations can also be a very expensive exercise. Nevertheless, many aerospace phenomena such as transonic buffet, aero-elastic flutter, vortex shedding associated with low Reynolds number flows, and aircraft noise all require unsteady flow simulations. Design problems involving turbine blades, helicopter blades and active flow control devices also require unsteady flow analysis. Although unsteady flow simulations remain expensive on current state-of-the-art computers, it is important to advance the computational tools suitable for analyzing and designing aerodynamic shapes for such flow applications since many aerospace problems depend on such advances.

The use of steady flow solutions to drive design optimization problems for aerodynamic applications is fairly well established [1–4]. In contrast, unsteady flow design optimization has not been explored to the same extent due largely to the high cost associated with simulating such flows. The earliest work in this area was undertaken by Yee *et al.* [5], who performed aerodynamic shape optimization of rotor airfoils using unsteady viscous flow simulations. They constructed a response surface model from unsteady viscous flow simulations and used the model to drive the optimization problem. He *et al.* [6] used the two-dimensional Navier Stokes equations to drive an optimization problem for an unsteady flow past a cylinder at a Reynolds number of 200 and 1000. They obtained the optimal frequency and amplitude required to rotate the cylinder in order to reduce drag and no geometric changes were allowed. Beyond these initial studies, many studies [7–17] have formulated the adjoint methodology [2, 18] for unsteady flows either in the time domain or the frequency domain, and have applied these formulations to various aerodynamic design problems. Some of the problems tackled include optimizing pitching airfoils, minimizing aircraft far-field noise, designing airfoils for low Reynolds number applications, designing a tilt rotor, as well as optimizing the lift-to-drag ratio of a fighter jet in the presence of aero-elastic effects. As computing power continue to improve, solving these expensive aerodynamic shape optimization problems will become commonplace and there is the need to provide a suite of benchmark cases to allow researchers to evaluate their algorithms. In our recent work [19], we presented and discussed some of our results towards achieving this objective. We focused primarily on minimizing the mean drag coefficient at a fixed lift for two- and three-dimensional geometries for laminar flow applications and also performed some convexity studies for the two-dimensional cases.

Our objective is to present a robust methodology for optimizing aerodynamic shapes under deterministic unsteady flow conditions, to demonstrate and characterize its performance, and to provide some insights into the solution of such unsteady optimization problems. To achieve this goal we investigate a number of example applications to allow us to study the effectiveness of the algorithm and to examine the shape changes qualitatively. First, we consider two-dimensional laminar flow optimization problems with different objective functions, starting from different initial geometries, and study the impact of design variable scaling [20] on lift maximization. Finally, we perform a lift-constrained drag minimization of a rectangular wing for laminar flow applications.

III. Optimization Problem and Methodology

In this section we cast the optimization problem in a general form and then discuss the individual components of the methodology used to solve it. This general form allows one to formulate the design problem in a compact form regardless of the underlying state equations that drive the optimization process.

A. Optimization Problem

The problem is defined as a constrained optimization problem of the form

$$\min_{\mathbf{X}} \quad \mathcal{J}(\mathbf{X}, \mathbf{V}) \quad \text{subject to} \quad \begin{cases} \mathcal{F}(\mathbf{X}, \mathbf{V}) = \mathbf{0} & \\ c_i(\mathbf{X}, \mathbf{V}) = 0 & i \in \mathcal{E} \\ c_i(\mathbf{X}, \mathbf{V}) \leq 0 & i \in \mathcal{I} \end{cases} \tag{1}$$

where the independent variables $\mathbf{X}$ and $\mathbf{V}$ represent the design variables and state variables for the underlying physical system, respectively, and $\mathcal{J}$ represents the objective function to be minimized. For unsteady flows, the objective function can be a time-averaged functional such as lift, drag, lift-to-drag ratio, or the functional at some time step n. The variable, $\mathcal{F}$, is the discretized partial differential equation (PDE) that governs the state of the physical system that drives the design problem. For the problems we consider this represents the *discretized flow* and *mesh-movement equations*. The flow state variable, $\mathbf{Q}$, and mesh-movement state variable, $\mathbf{B}$, are lumped into a single state variable, $\mathbf{V}$. The functions $c_i, i \in \mathcal{E}$ are *equality constraints*, while $c_i, i \in \mathcal{I}$ are *inequality constraints* (these constraints are smooth real-valued linear/nonlinear functions). We use the SNOPT optimizer [21] to solve the optimization problem. Since SNOPT is a gradient-based optimizer, we must provide the gradients for the objective and constraint functions.

To compute the gradients of the objective and constraints that depend on the state variable $\mathbf{V}$ we use the discrete adjoint approach. Since the PDE equality constraint, $\mathcal{F}$, is zero for any value of $\mathbf{X}$, the optimization problem in equation (1) can be reformulated as a Lagrangian problem of the form

$$\min_{\mathbf{X}} \quad \mathcal{L}(\mathbf{X}, \mathbf{V}, \mathbf{\Lambda}) = \mathcal{J}(\mathbf{X}, \mathbf{V}) + \mathbf{\Lambda}^T \mathcal{F}(\mathbf{X}, \mathbf{V}) \tag{2}$$

$$\text{subject to} \quad \begin{cases} c_i(\mathbf{X}, \mathbf{V}) = 0 & i \in \mathcal{E} \\ c_i(\mathbf{X}, \mathbf{V}) \leq 0 & i \in \mathcal{I} \end{cases}$$

where $\mathbf{\Lambda}$ is the state adjoint vector (Lagrange multiplier). Since the state variables in $\mathbf{V}$ depend implicitly on the design variables in $\mathbf{X}$ through the PDE constraint, for any valid $\mathbf{X}$, there is a unique solution for $\mathbf{V}(\mathbf{X})$. Therefore, once the discrete PDE constraint $\mathcal{F} = \mathbf{0}$ is solved, the optimizer updates only the design variables, $\mathbf{X}$, at each design iteration.

The gradient, $\mathcal{G}$, of the Lagrangian in equation (2) with respect to the design variables is given as

$$\mathcal{G} = \left(\frac{\partial \mathcal{J}}{\partial \mathbf{X}} + \mathbf{\Lambda}^T \frac{\partial \mathcal{F}}{\partial \mathbf{X}} \right) + \left(\frac{\partial \mathcal{J}}{\partial \mathbf{V}} + \mathbf{\Lambda}^T \frac{\partial \mathcal{F}}{\partial \mathbf{V}} \right) \frac{d\mathbf{V}}{d\mathbf{X}} \tag{3}$$

To derive a formulation consistent with obtaining the first-order (necessary) optimality conditions for the optimization problem, the adjoint variables are chosen such that

$$\left[\frac{\partial \mathcal{F}}{\partial \mathbf{V}} \right]^T \mathbf{\Lambda} + \left[\frac{\partial \mathcal{J}}{\partial \mathbf{V}} \right]^T = \mathbf{0} \tag{4}$$

and in the process we avoid computing the sensitivities of the state variables with respect to the design variables (i.e. $d\mathbf{V}/d\mathbf{X}$). Equation (4) is referred to as the adjoint problem. Once the governing equations and adjoint equations have been solved, the gradient of the optimization problem can be computed from equation (3) using

$$\mathcal{G} = \frac{\partial \mathcal{J}}{\partial \mathbf{X}} + \mathbf{\Lambda}^T \frac{\partial \mathcal{F}}{\partial \mathbf{X}} \tag{5}$$

We discuss the details of the gradient computation and adjoint problem in the subsequent subsections.

B. Geometry Control and Mesh-Movement

We use B-splines to parameterize all of the geometries in this work. This allows us to retain an analytical representation of the geometries, thereby making it easier to compute sensitivities with respect to the surface. The B-spline surface representation is embedded in a Free-Form (FFD) deformation volume to allow us to control the geometry during optimization. The FFD control is combined with an axial deformation curve to allow for planform changes (i.e. sweep, dihedral etc.) [22]. The FFD geometry control gives us flexibility in the choice of the number of geometric design variables since it is independent of the parameterization [23].

We have a coupled mesh-movement algorithm that allows us to update the grid when the geometry changes. This algorithm formulates a state equation based on linear elasticity theory to move the grid after a geometry is updated [24]. The stiffness matrix for the linear elasticity problem is defined such that smaller mesh elements have higher stiffness and vice-versa. An orthogonality measure is used to apply additional stiffness to skewed elements to help retain their quality throughout the optimization. To ensure that the small-strain assumption used for the linear elasticity problem remains valid, the mesh-movement is broken into increments when large geometric changes are expected. B-splines are used to parameterize the grid for the purposes of moving the mesh. This leads to about an one order magnitude reduction in the size of the mesh-movement linear problem compared to the flow problem. The state equation and variables for the linear problem at each mesh-movement increment are $\mathcal{M}_j$ and $\mathbf{B}_j$ respectively.

C. Flow Problem

The flow solver is based primarily on work in [25–27]. A short description of the flow solver and the solution strategy for the unsteady flow problem is presented below.

1. Governing Equations

The Navier-Stokes equations in generalized curvilinear coordinates (i.e. $(x, y, z) \rightarrow (\xi, \eta, \zeta)$) are given by

$$\partial_t \hat{\mathbf{Q}} + \partial_\xi \hat{\mathbf{E}} + \partial_\eta \hat{\mathbf{F}} + \partial_\zeta \hat{\mathbf{G}} = Re^{-1} \left(\partial_\xi \hat{\mathbf{E}}_\mathbf{v} + \partial_\eta \hat{\mathbf{F}}_\mathbf{v} + \partial_\zeta \hat{\mathbf{G}}_\mathbf{v} \right) \tag{6}$$

where

$$\hat{\mathbf{Q}} = J^{-1}\mathbf{Q}$$

$$\hat{\mathbf{E}} = J^{-1}\left(\xi_x \mathbf{E} + \xi_y \mathbf{F} + \xi_z \mathbf{G}\right), \quad \hat{\mathbf{F}} = J^{-1}\left(\eta_x \mathbf{E} + \eta_y \mathbf{F} + \eta_z \mathbf{G}\right), \quad \hat{\mathbf{G}} = J^{-1}\left(\zeta_x \mathbf{E} + \zeta_y \mathbf{F} + \zeta_z \mathbf{G}\right),$$

$$\hat{\mathbf{E}}_\mathbf{v} = J^{-1}\left(\xi_x \mathbf{E}_\mathbf{v} + \xi_y \mathbf{F}_\mathbf{v} + \xi_z \mathbf{G}_\mathbf{v}\right), \quad \hat{\mathbf{F}}_\mathbf{v} = J^{-1}\left(\eta_x \mathbf{E}_\mathbf{v} + \eta_y \mathbf{F}_\mathbf{v} + \eta_z \mathbf{G}_\mathbf{v}\right), \quad \hat{\mathbf{G}}_\mathbf{v} = J^{-1}\left(\zeta_x \mathbf{E}_\mathbf{v} + \zeta_y \mathbf{F}_\mathbf{v} + \zeta_z \mathbf{G}_\mathbf{v}\right)$$

The vector $\mathbf{Q}$ represents the conserved variables, and J is the metric Jacobian resulting from the coordinate transformation from Cartesian to curvilinear coordinates. The vectors $\mathbf{E}$, $\mathbf{F}$ and $\mathbf{G}$ contain the inviscid fluxes, while the vectors $\mathbf{E}_\mathbf{v}$, $\mathbf{F}_\mathbf{v}$ and $\mathbf{G}_\mathbf{v}$ contain the viscous fluxes. The term ∂_x is a shorthand for $\frac{\partial}{\partial x}$, ξ_x is a shorthand for $\frac{\partial \xi}{\partial x}$, and so on. The reader can look up [28] for more details on how the coordinate transformation is applied to the governing equations.

2. Discretization

We discretize the Navier-Stokes equations on structured multi-block grids using second-order summation-by-parts (SBP) operators. At block interfaces and boundaries, simultaneous approximation terms (SATs) are used to enforce continuity and boundary conditions in an accurate and stable manner. Further details on this discretization strategy can be found in [25, 26, 29]

The semi-discrete form of the governing equations is given as

$$\frac{d\hat{\mathbf{Q}}}{dt} + \hat{\mathcal{R}}(\hat{\mathbf{Q}}) = \mathbf{0} \tag{7}$$

where the vector $\hat{\mathcal{R}}$ contains the discretized spatial terms. The semi-discrete equation (7) is recast into a fully-discrete equation of the form

$$\tilde{\mathcal{R}}(\hat{\mathbf{Q}}) + \hat{\mathcal{R}}(\hat{\mathbf{Q}}) = \mathbf{0} \tag{8}$$

where $\tilde{\mathcal{R}}$ represents the contribution to the discrete residual associated with the time-marching method.

For time integration, the second-order backward formula (BDF2) and the explicit singly-diagonal implicit Runge-Kutta (ESDIRK) schemes are considered. At each time step n, the flow residual $\mathcal{R}_n$ for a linear multi-step method such as BDF2 can be expressed as

$$\mathcal{R}_n = \tilde{\mathcal{R}}_n + \hat{\mathcal{R}}_n \quad \text{for } n = 1, \dots, N \tag{9}$$

where N is the number of time steps and

$$\tilde{\mathcal{R}}_n = \frac{3\hat{\mathbf{Q}}_n - 4\hat{\mathbf{Q}}_{n-1} + \hat{\mathbf{Q}}_{n-2}}{2\Delta t}$$

If we consider a multi-stage time marching method like ESDIRK4, the discrete residual is given as

$$\mathcal{R}_j^{(n)} = \frac{1}{a_{jj}}\tilde{\mathcal{R}}_j^{(n)} + \hat{\mathcal{R}}_j^{(n)} + \sum_{k=1}^{j-1} \frac{a_{jk}}{a_{jj}} \hat{\mathcal{R}}_k^{(n)} \quad \text{for } n = 1, \dots, N \quad j = 2, \dots, 6 \tag{10}$$

where

$$\tilde{\mathcal{R}}_j^{(n)} = \frac{\hat{\mathbf{Q}}_j^{(n)} - \hat{\mathbf{Q}}^{(n-1)}}{\Delta t}$$

and a_{jj} and a_{jk} are the coefficients associated with the ESDIRK scheme. The explicit first stage is specified as $\hat{\mathbf{Q}}_1^{(n)} = \hat{\mathbf{Q}}^{(n-1)}$, and the solution at time step n is given as $\hat{\mathbf{Q}}^{(n)} = \hat{\mathbf{Q}}_6^{(n)}$. The BDF2 time-marching method is used for all the results presented in this paper.

3. Solution to the Discrete Problem

The resulting system of nonlinear equations $\mathcal{R}_n(\hat{\mathbf{Q}}_1, \ldots, \hat{\mathbf{Q}}_{n-1}, \hat{\mathbf{Q}}_n) = \mathbf{0}$ at each time step n or stage j is solved using an inexact-Newton method. The linear problem from the linearized Newton problem at each time step or stage is solved using a Krylov subspace method. A distributed Schur-complement technique [30] is used to precondition the distributed linear system, and the preconditioner is obtained from an incomplete Lower-Upper (ILU) factorization with some fill. Since the Schur-complement preconditioning changes from iteration to iteration, a flexible variant of the generalized minimal residual algorithm (FMGRES) [31, 32] is used to solve the distributed linear system.

D. Gradient Computation

Let the state vector, $\mathcal{F}$, which contains the mesh-movement and flow equations be

$$\mathcal{F} = [\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m, \mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_N]^T \tag{11}$$

and the state variable vector, $\mathbf{V}$, which contains the mesh-movement control points and flow conserved variables be

$$\mathbf{V} = [\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_m, \mathbf{Q}_0, \mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_N]^T \tag{12}$$

The adjoint variable vector, $\mathbf{\Lambda}$, which contains the mesh-movement and flow adjoints is also given as

$$\mathbf{\Lambda} = \left[\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_m, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_N\right]^T \tag{13}$$

The mesh-movement equations, state variables and adjoint variables are $\mathcal{M}$, $\mathbf{B}$ and $\boldsymbol{\lambda}$, respectively, and the flow equations, state variables and adjoint variables are defined as $\mathcal{R}$, $\mathbf{Q}$ and $\boldsymbol{\psi}$, respectively. Here, m is the number of mesh movement increments and the N is the number of flow time steps. For a linear multi-step method such as BDF2 we can rewrite equation (2) with its constituent state equations as

$$\mathcal{L} = \sum_{n=1}^{N} \left[\omega_n \mathcal{J}_n + \boldsymbol{\psi}_n^T \mathcal{R}_n\right] + \sum_{j=1}^{m} \boldsymbol{\lambda}_j^T \mathcal{M}_j \tag{14}$$

where ω_n is the integration weight assigned to the objective function contribution (i.e. $\mathcal{J}_n$) from time step n. Further, we can write the gradient equation (5) as

$$\mathcal{G} = \sum_{n=1}^{N} \left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{X}} + \boldsymbol{\psi}_n^T \frac{\partial \mathcal{R}_n}{\partial \mathbf{X}}\right] + \sum_{j=1}^{m} \boldsymbol{\lambda}_j^T \frac{\partial \mathcal{M}_j}{\partial \mathbf{X}} \tag{15}$$

If one considers a multi-stage method such as ESDIRK4, equation (2) can be reformulated as

$$\mathcal{L} = \sum_{n=1}^{N} \left[\omega_n \mathcal{J}_n + \sum_{k=2}^{6} \left(\boldsymbol{\psi}_k^{(n)}\right)^T \mathcal{R}_k^{(n)}\right] + \sum_{j=1}^{m} \boldsymbol{\lambda}_j^T \mathcal{M}_j \tag{16}$$

and the gradient equation (5) as

$$\mathcal{G} = \sum_{n=1}^{N} \left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{X}} + \sum_{k=2}^{6} \left(\boldsymbol{\psi}_k^{(n)}\right)^T \frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{X}}\right] + \sum_{j=1}^{m} \boldsymbol{\lambda}_j^T \frac{\partial \mathcal{M}_j}{\partial \mathbf{X}} \tag{17}$$

E. Adjoint Problem

The adjoint variables present in the gradient computation are computed as follows for the BDF2 time marching method

$$\left[\frac{\partial \mathcal{R}_n}{\partial \mathbf{Q}_n}\right]^T \boldsymbol{\psi}_n = -\left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{Q}_n}\right]^T \quad \text{for } n = N \tag{18}$$

$$\left[\frac{\partial \mathcal{R}_n}{\partial \mathbf{Q}_n}\right]^T \boldsymbol{\psi}_n = -\left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{Q}_n} + \sum_{k=1}^{p} \boldsymbol{\psi}_{n+k}^T \frac{\partial \mathcal{R}_{n+k}}{\partial \mathbf{Q}_n}\right]^T \quad \text{for } n = N-1, \ldots, 1\ ,\ p = \min\left[N-n, r\right] \tag{19}$$

$$\left[\frac{\partial \mathcal{M}_j}{\partial \mathbf{B}_j}\right]^T \boldsymbol{\lambda}_j = -\underbrace{\left[\sum_{n=1}^{N}\left(\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{B}_j} + \boldsymbol{\psi}_n^T \frac{\partial \mathcal{R}_n}{\partial \mathbf{B}_j}\right)\right]^T}_{\text{last mesh-deformation increment RHS}} \quad \text{for } j = m \tag{20}$$

$$\left[\frac{\partial \mathcal{M}_j}{\partial \mathbf{B}_j}\right]^T \boldsymbol{\lambda}_j = -\left[\boldsymbol{\lambda}_{j+1}^T \frac{\partial \mathcal{M}_{j+1}}{\partial \mathbf{B}_j}\right]^T \quad \text{for } j = m-1, \ldots, 1 \tag{21}$$

where r is the number of time levels of previous solutions required for the time marching method. For BDF2, r is equal to 2 (r can be varied to generalize the above expressions for any linear multi-step time marching method).

For a multi-stage time marching method such as the ESDIRK4, the adjoint variables are computed by solving the following equations

$$\left[\frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \boldsymbol{\psi}_k^{(n)} = -\left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{Q}_k^{(n)}}\right]^T \quad \text{for } n = N, \quad k = 6 \tag{22}$$

$$\left[\frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \boldsymbol{\psi}_k^{(n)} = -\left[\sum_{i=k+1}^{6} \left(\boldsymbol{\psi}_i^{(n)}\right)^T \frac{\partial \mathcal{R}_i^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \quad \text{for } n = N, \quad k = 5, \ldots, 2 \tag{23}$$

$$\left[\frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \boldsymbol{\psi}_k^{(n)} = -\left[\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{Q}_k^{(n)}} + \sum_{i=2}^{6} \left(\boldsymbol{\psi}_i^{(n+1)}\right)^T \frac{\partial \mathcal{R}_i^{(n+1)}}{\partial \mathbf{Q}_k^{(n)}} - \boldsymbol{\psi}_1^{(n+1)}\right]^T \quad \text{for } n = N-1, \ldots, 1, \quad k = 6 \tag{24}$$

$$\left[\frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \boldsymbol{\psi}_k^{(n)} = -\left[\sum_{i=k+1}^{6} \left(\boldsymbol{\psi}_i^{(n)}\right)^T \frac{\partial \mathcal{R}_i^{(n)}}{\partial \mathbf{Q}_k^{(n)}}\right]^T \quad \text{for } n = N-1, \ldots, 1, \quad k = 5, \ldots, 2 \tag{25}$$

$$\boldsymbol{\psi}_1^{(n)} = -\left[\sum_{i=2}^{6} \left(\boldsymbol{\psi}_i^{(n)}\right)^T \frac{\partial \mathcal{R}_i^{(n)}}{\partial \mathbf{Q}_1^{(n)}}\right]^T \tag{26}$$

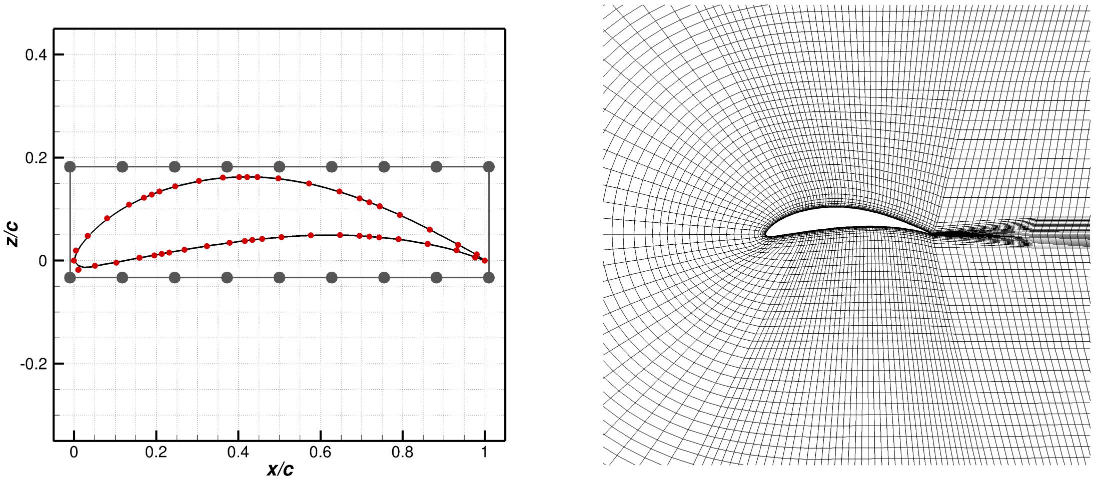
$$\left[\frac{\partial \mathcal{M}_j}{\partial \mathbf{B}_j}\right]^T \boldsymbol{\lambda}_j = -\underbrace{\left[\sum_{n=1}^{N}\left(\omega_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{B}_j} + \sum_{k=2}^{6} \left(\boldsymbol{\psi}_k^{(n)}\right)^T \frac{\partial \mathcal{R}_k^{(n)}}{\partial \mathbf{B}_j}\right)\right]^T}_{\text{last mesh-deformation increment RHS}} \quad \text{for } j = m \tag{27}$$

$$\left[\frac{\partial \mathcal{M}_j}{\partial \mathbf{B}_j}\right]^T \boldsymbol{\lambda}_j = -\left[\frac{\partial \mathcal{M}_{j+1}}{\partial \mathbf{B}_j} \boldsymbol{\lambda}_{j+1}\right]^T \quad \text{for } j = m-1, \ldots, 1 \tag{28}$$

We refer the reader to [19] for the full derivation of the unsteady flow adjoint and mesh-movement adjoint equations.

IV. Two-Dimensional Optimization Problems

In this section we investigate two-dimensional geometry optimization for laminar flow applications. For all of the cases, we consider a flow at Mach 0.2 and a Reynolds number of 800. We use the Chuch Hollinger 10 smoothed (CH10SM) airfoil as the baseline geometry and the NACA0012 airfoil as an initial geometry for convexity studies. We also enforce a minimum area constraint of 0.08 and constrain the thickness to decrease by not more than 25% of the initial value for all of the cases. The control window for the optimization problems is set to a nondimensional time interval of $t = [46, 76]$ with a time step of $\Delta t = 0.01$. The time step intervals $[1, 1000]$, $[1001 - 2000]$, and $[2001 - 4000]$ where computed with a time step of $\Delta t = 0.014$, $\Delta t = 0.012$, and $\Delta t = 0.01$ respectively. These time-step windows where considered to be part of the adjusting phase of the flow. The computational mesh is divided into 120 blocks, and each block has 15 grid points in the both the streamwise and normal directions. Sixty mesh nodes are placed on both the upper and lower surfaces of the mesh, and 120 mesh points are placed in the wake region. The off-wall spacing was set to 1.94×10^{-4} chord lengths and the far-field boundary is placed 4 chord lengths in the normal direction and 8 chord lengths in the streamwise direction. The size of the grid and the far-field distance introduce significant errors, hence, we under-predict the aerodynamic coefficients by about 10% compared to the grid converged values. Our goal is to study the effectiveness of our methodology and examine the shape changes qualitatively; therefore, these



(a) Parameterized Airfoil

(b) CFD Grid

Fig. 1 CH10SM airfoil

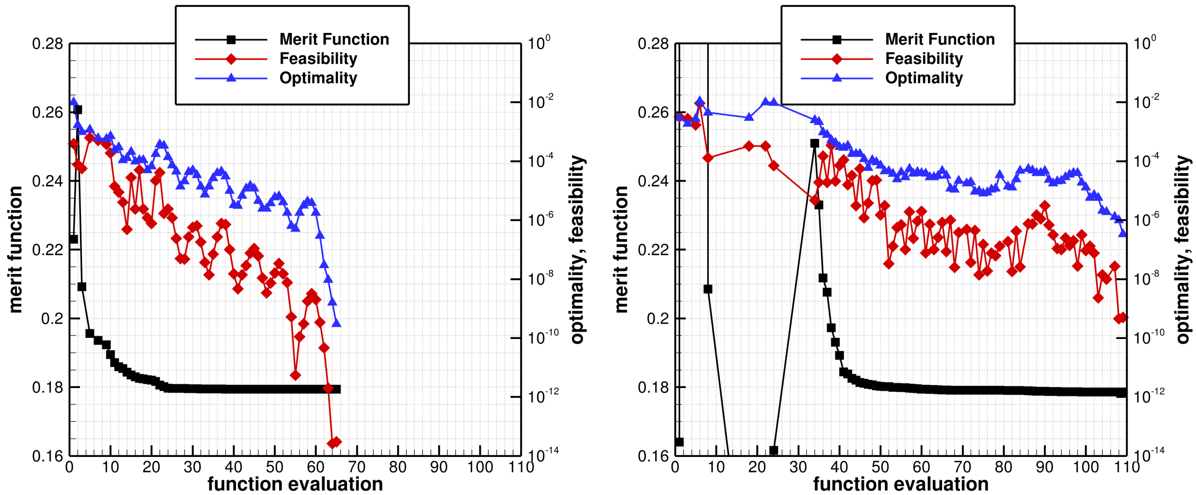
choices are sufficient and enable suitable turnaround times for our study. Figure 1 shows the CH10SM airfoil and the flow grid. The red-circles in Figure 1a represent the B-spline control points, and the gray circles and connecting lines show the FFD box for controlling the geometry. The geometry is parameterized with 5 B-spline control points along the streamwise direction of each of the 8 blocks wrapped around the geometry. To control the geometry, the parameterized geometry is embedded in a FFD box with 18 control points, leading to 18 geometric design variables in addition to the angle of attack design variable.

A. Mean Drag Coefficient Minimization with a Mean Lift Constraint

Here the objective is to minimize the mean drag coefficient at a fixed mean lift coefficient of 0.75. For the initial CH10SM airfoil, the lift target is achieved at an angle of attack of 8.6°. The flow separates around the mid-chord of the CH10SM airfoil and vortices are shed in the wake region, leading to a periodic flow behaviour. For the NACA0012 airfoil the lift target is achieved at an angle of attack of 18.07° and the flow behaviour is periodic as well due to vortex shedding. Both the CH10SM and NACA0012 optimization problems were started at an initial angle of attack of 8°.

Figure 2 shows the optimization histories for the CH10SM and NACA0012 cases. Throughout this paper, *merit function* is defined as a composite function of the objective function and the nonlinear constraints, and *optimality* is a measure of the norm of the gradient of the merit function. *Feasibility* is used to measure the violation in the non-linear constraint, which is the mean-lift for this study. For the CH10SM airfoil case, we obtained about 99% of all the improvement in the objective after 30 function evaluations and optimality was reduced by over 6 orders of magnitude. Feasibility was also reduced to machine zero. For the NACA0012 case, optimality was reduced by about 4 orders of magnitude and feasibility was reduced by about 7 orders of magnitude. Almost all of the objective improvement was realized after about 70 function evaluations. The NACA0012 initial problem had a higher violation in the lift constraint so the optimizer required more function evaluations to converge. Further, the area constraint is active and is the same value (i.e. 0.080) for both cases.

We show the optimized geometries and the history of the aerodynamic coefficients in Figures 3 and 4 respectively. The optimizer primarily moves the camber towards the trailing edge to achieve the lift target and flattens the upper surface of the airfoil to delay flow separation (see Figure 5). This reduces the size of the separated wake, thereby reducing drag. The two different initial geometries produce nearly identical airfoils. The resulting geometries also compare well to the benchmark case (i.e. referred to as the *B-spline* case throughout the paper) in [19] where the B-spline surface control points were used to control the geometry rather than the FFD control points. The differences in the flap-like feature at the trailing edge arise because the B-spline case has more geometric freedom in that region. The B-spline case has 4 control points between 80% and 100% of the chord compared to 1 control point for the FFD geometry control. This difference in design variables can be seen in Figure 1a, where the red circles represent the



(a) CH10SM airfoil

(b) NACA0012 airfoil

Fig. 2 Mean coefficient of drag minimization - optimization history

	CH10SM	CH10SM OPT(B-spline)	CH10SM OPT	NACA0012 OPT
α	8.6°	7.68°	8.67°	9.38°
C_L	0.7500	0.7500	0.7500	0.7500
C_D	0.2288	0.1845	0.1794	0.1786
ΔC_D	-	19.36%	21.59%	21.94%
C_L/C_D	3.278	4.006	4.181	4.199

Table 1 Mean coefficient of drag minimization summary

B-spline control points and the grey circles represent the the FFD control points. We also observed that the FFD offset at the trailing edge tends to influence the constraint that fixes the trailing edge during optimizations. When large geometric changes occur relative to the initial geometry, the trailing edge tends to move slightly. As a result, the symmetric NACA0012 case ends up with a slight negative twist and the optimizer increases the angle of attack slightly compared to the CH10SM airfoil (see Table 1). Further investigation into the sizing and the location of the FFD sections relative to the initial geometry may be required to understand how these parameters impact the optimization results. In terms of convexity, our view is that this optimization problem is likely to be convex and the differences we observe can be attributed mainly to the geometry control and how the fixed trailing edge is handled. In Figures 4a and 4b we observe a significant drop in the amplitude of the oscillating aerodynamic quantities. We attribute this reduction to a combination of factors. First, the wake of the optimized airfoil is narrower and the vortices shed are weaker compared to the initial CH10SM airfoil (see Figure 6). In addition, the flap-like feature also seems to delay the interaction between the lower and upper surface wakes. The delayed interaction pushes the shedding vortices further downstream, thereby minimizing the induced force on the geometry. As shown in Table 1, a drag reduction (ΔC_D) of more than 20% was observed for both initial geometries, which is comparable to the B-spline case investigated in [19].

B. Mean Lift-to-Drag Ratio Maximization

For this case we maximize the mean lift-to-drag ratio. Here as well we consider the CH10SM and the NACA0012 airfoils as the initial geometries. We start both problems at an initial angle of attack of 8°. Figure 7 shows the optimization histories for the CH10SM and the NACA0012 airfoils. Similar to the drag minimization problem, the optimizer flattens the upper surface (see Figure 8a) which keeps the flow attached to compensate for the drag penalty it incurs by increasing the lift. The flap-like feature at the trailing edge essentially controls the camber required to increase

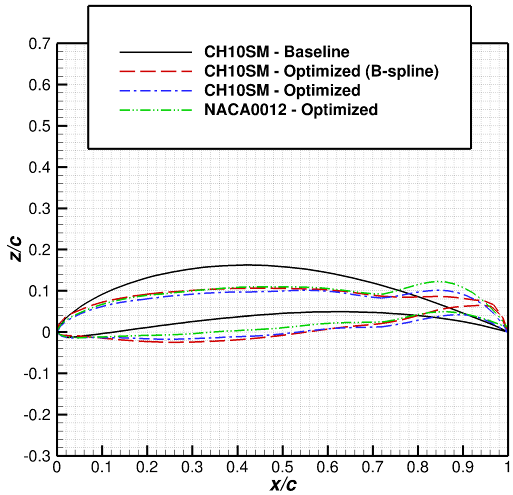
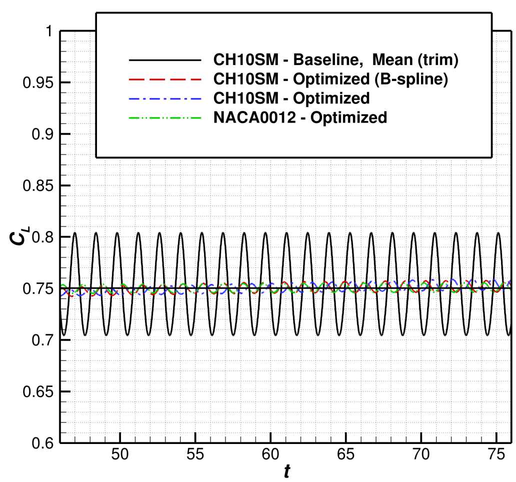
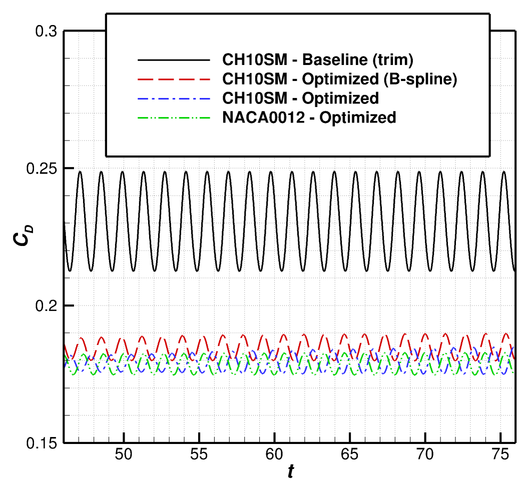


Fig. 3 Mean coefficient of lift-constrained drag minimization optimized geometries

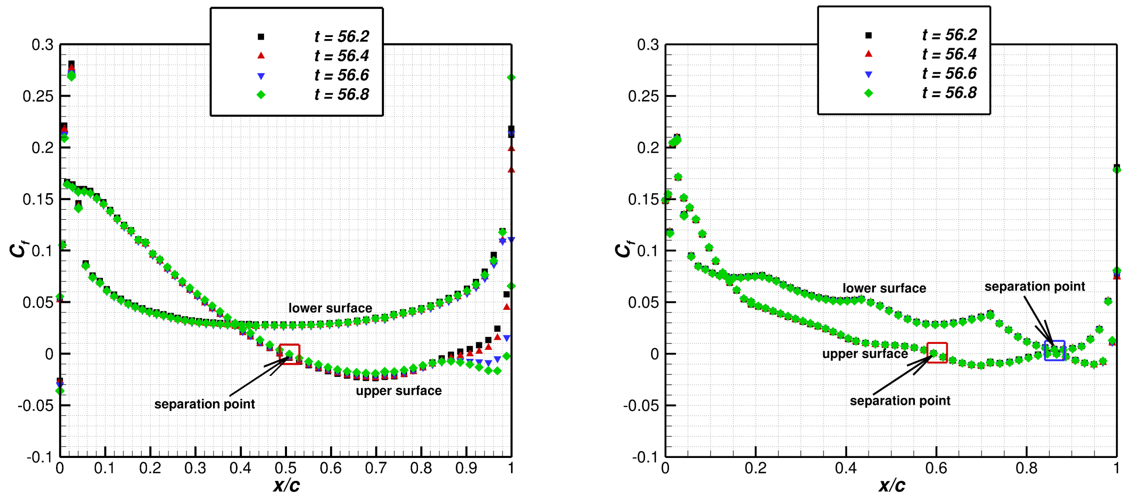


(a) Coefficient of lift



(b) Coefficient of drag

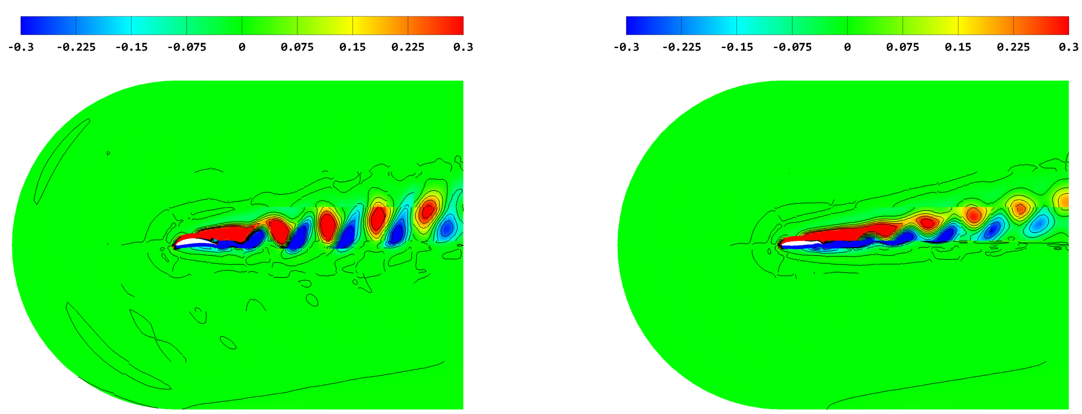
Fig. 4 Time history of aerodynamic coefficients



(a) Coefficient of friction for a half-cycle - baseline

(b) Coefficient of friction for a half-cycle - optimized

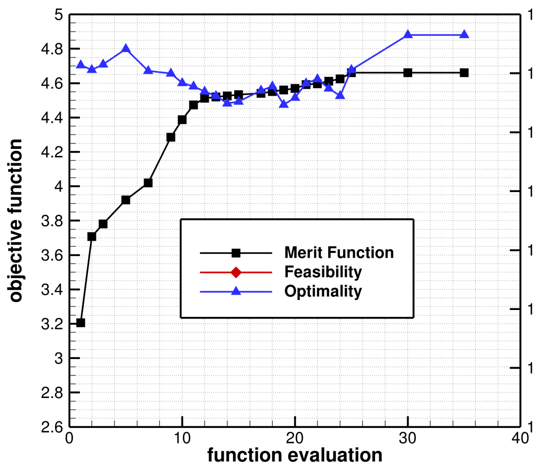
Fig. 5 Coefficient of friction and separation points for a half cycle



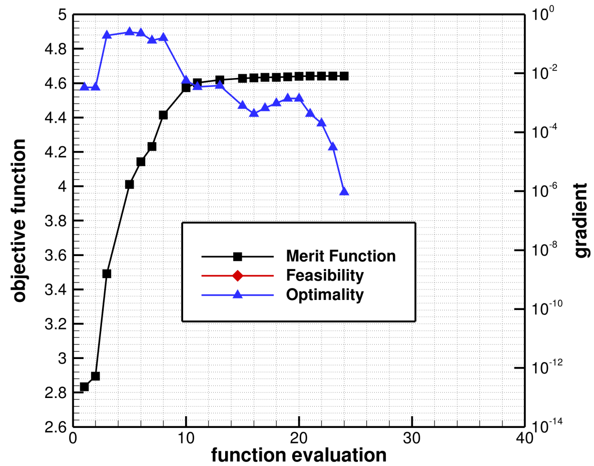
(a) CH10SM baseline at $C_L = 0.75$

(b) CH10SM optimized

Fig. 6 Y-vorticity contours at the final step

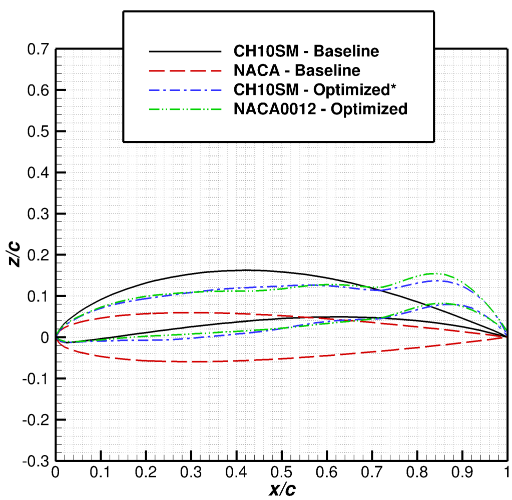


(a) CH10SM airfoil

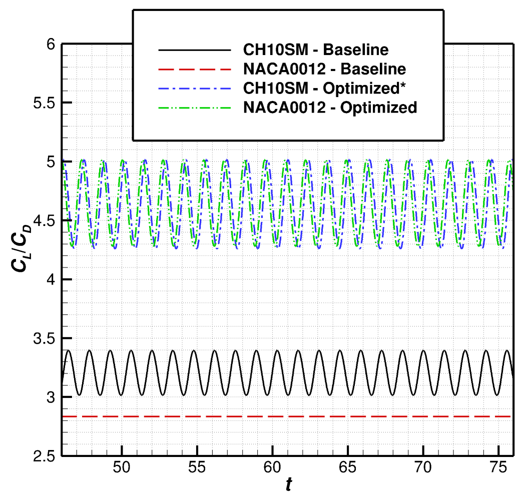


(b) NACA0012 airfoil

Fig. 7 Mean lift-to-drag ratio maximization results - optimization history



(a) Optimized geometries



(b) Lift-to-drag ratio time history

Fig. 8 Mean lift-to-drag ratio maximization results

the lift. Compared to the initial CH10SM airfoil, the optimizer achieves an increase in lift of over 40% with roughly the same drag for an increase in lift-to-drag ratio of about 49% as seen in Figure 8b and Table 2. While the final result for the CH10SM baseline is similar to that of the NACA0012 baseline, the optimizer stalled for the CH10SM case and could not reduce the norm of the gradient by more than an order of magnitude. The asterisk by the CH10SM label in 8a indicates that the case is not fully converged. Nevertheless, the geometry and its performance is similar to that resulting from the optimization with the NACA0012 as the initial airfoil. Similar to the NACA0012 drag minimization case, the trailing edge for the NACA0012 moves up slightly for the same reasons as discussed earlier. This leads to an increase in the angle of attack for the final geometry. In terms of convexity, our observations indicate this problem is likely to be convex or uni-modal, but we believe further investigation is needed to confirm this with a high degree of certainty. Finally, the area constraint is also active for all of the results.

Case	CH10SM - Baseline	NACA0012 - Baseline	CH10SM	NACA 0012
α	8°	8°	10.79°	12.05°
C_L	0.7150	0.4650	1.063	1.057
C_D	0.2309	0.1640	0.230	0.228
C_L/C_D	3.097	2.814	4.618	4.631

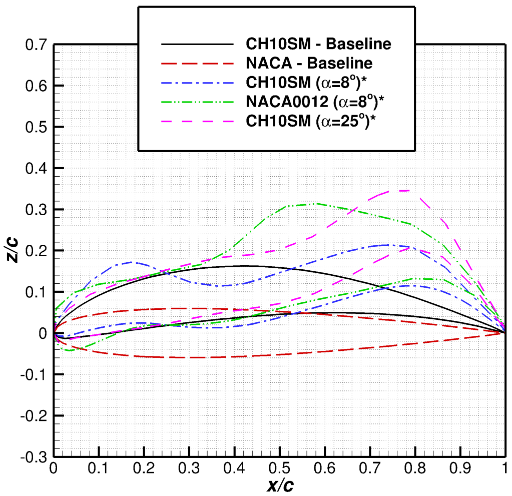
Table 2 Mean lift-to-drag ratio maximization summary

C. Mean Lift Coefficient Maximization

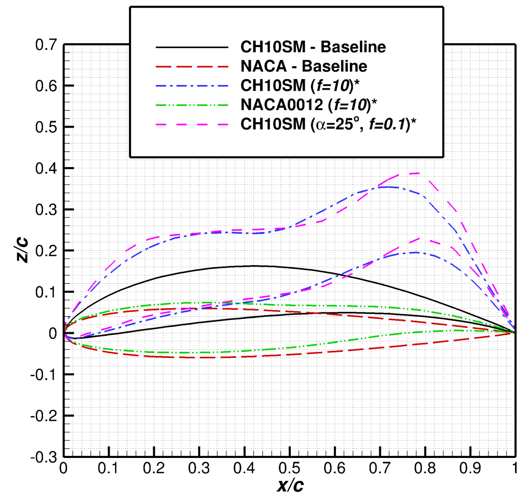
Here we consider maximizing the mean lift of the two initial airfoils again at a Reynolds number of 800 and Mach 0.2. The thickness and area constraints are the same as the values used for the previous cases (i.e. $t \geq 0.75t_0$ and $A \geq 0.08$). For both initial airfoils, we consider a base case with an initial angle of attack of 8°. We also consider the case where the angle of attack design variable is scaled by a constant factor. We show the final geometries for these cases in Figure 9, with convergence histories displayed in Figure 10 and details provided in Table 3. The optimizer increases the camber of the initial geometries significantly in order to maximize the mean lift. The camber is also pushed towards the trailing edge region, leading to a flap-like feature similar to the previous cases.

Initially, the CH10SM base cases stalled and the optimizer was unable to increase the lift as expected. The mesh-movement algorithm failed at times due to large geometric changes. In addition, the optimizer did not increase the angle of attack (see Table 3) as expected, so we reran the case at an initial angle of attack of 25°. By increasing the initial angle of attack, the optimizer was able to increase the lift to the levels seen in the NACA0012 case. For the NACA0012 base case, the optimizer reduced the norm of the gradient by 3 orders of magnitude (i.e. from O^{-3} to O^{-6}), which is sufficient to indicate that a local minimum has been found.

To investigate why the optimizer struggled to improve the lift-maximization problems, we looked at how the norm of the gradient with respect to the geometric design variables compares to the gradient with respect to the angle of attack in the course of an optimization. From Figure 10, the initial gradient of the objective to the geometric design variables ($\mathcal{G}_g$) is much larger than the objective sensitivity to angle of attack ($\mathcal{G}_a$) for the CH10SM case. Therefore, the optimizer applies large geometric changes in the initial iterations which leads to mesh-movement problems for the CH10SM cases. Eventually, the failed mesh movements lead to convergence issues. Typically, one would expect the optimizer to prioritize increasing angle of attack in the initial stages to maximize lift, so we scaled the angle of attack inversely by a constant factor (i.e. f). Table 4 shows the improvement we get from scaling the angle of attack design variable. We include a case where angle of attack is scaled inversely by a factor 0.1 to reinforce the idea that when the initial problem is less sensitive to the angle of attack design variable, the optimizer relies primarily on large geometric changes to maximize lift. In Figure 11 we show that scaling the angle of attack design variable inversely by a factor of 10 makes the initial problem more sensitive to changes in angle of attack. This allows the optimizer to prioritize increasing the the angle of attack design variable over the geometric design variables in the initial stages. When the constant scaling ceases to be effective, the optimizer resorts to large geometric changes to maximize the lift which leads to convergence issues later on due to mesh-movement failures. Further investigation into design variable scaling and an automated approach for backtracking and modifying the number of mesh-movement increments and other parameters such as the Poisson ratio used for the linear elasticity problem could help reduce mesh-movement failures during optimizations where large geometric changes are expected. Comparing the results obtained for the two different initial airfoils shown in Table 4 and in Figure 11, it appears that two different local minima are being found. The airfoils are very different as are the angles of attack, with the optimization initiated with the NACA0012 airfoil producing a higher lift coefficient and much higher drag. In addition, the optimization problem initiated with the NACA0012 airfoil without scaling the angle of attack design variable also produces a different local minimum. These different local minima suggest that this optimization problem is multi-modal, or non-convex. The different local minima do not result solely from the different initial geometry. For example, the results initiated with the NACA0012 shown in Tables 3 and 4 converge to two different local minima as a result of the different design variable scaling.

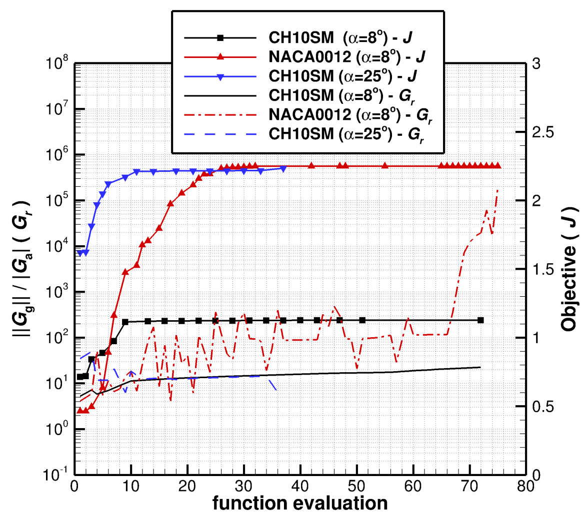


(a) Results without design variable scaling

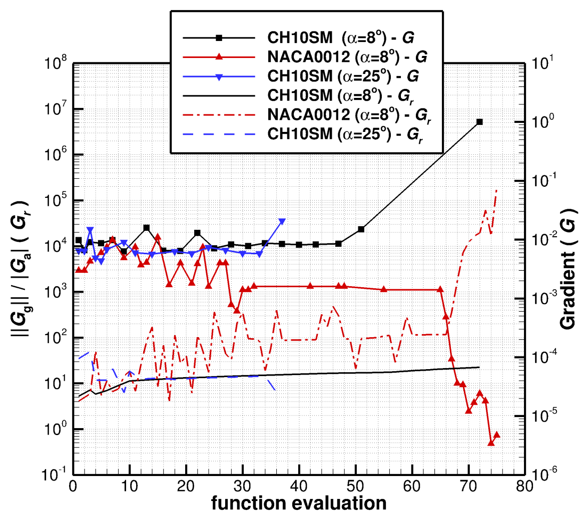


(b) Results with scaling of the angle of attack design variable

Fig. 9 Mean lift maximization results



(a) Objective function

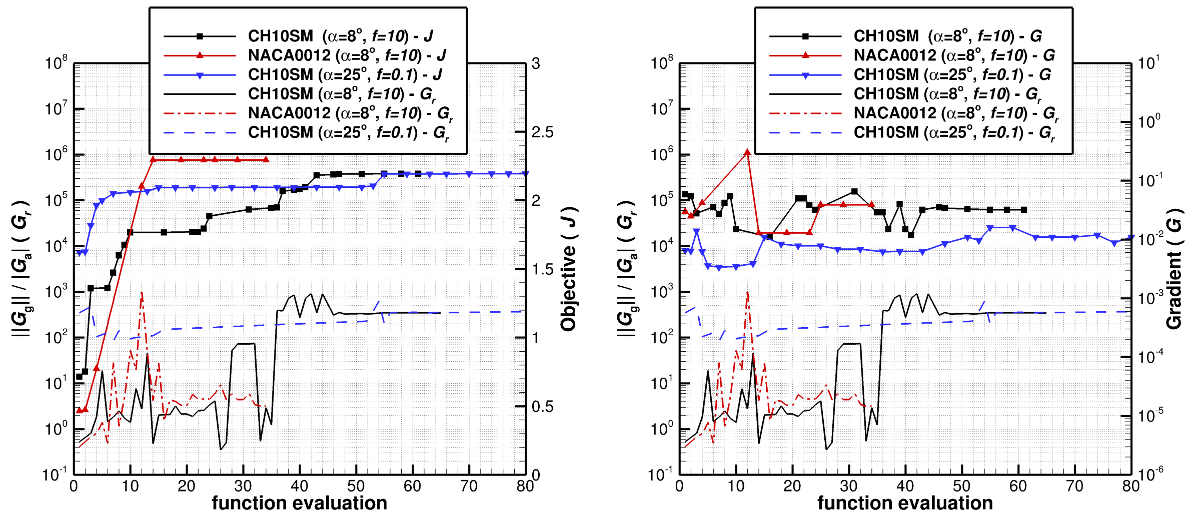


(b) Gradient

Fig. 10 Base cases with no design variable scaling - gradients comparison $\mathcal{G}_r = \frac{||\mathcal{G}_g||_2}{|\mathcal{G}_a|}$

Case	CH10SM ($\alpha = 8°$)	NACA 0012 ($\alpha = 8°$)	CH10SM($\alpha = 25°$)
α	9.98°	30.35°	27.161°
C_L	1.1270	2.2502	2.2337
C_D	0.3110	1.0718	0.8474
C_L/C_D	3.624	2.099	2.525
Area, A	0.09608	0.1453	0.1146
iterations, k	72	75	37

Table 3 Mean lift maximization summary - no design variable scaling



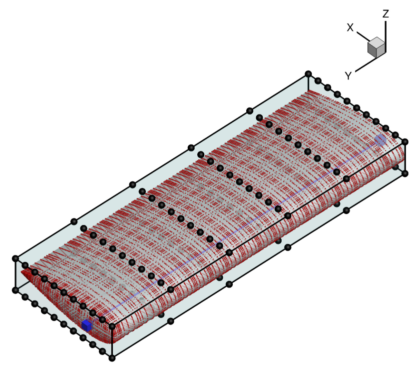
(a) Objective function

(b) Gradient

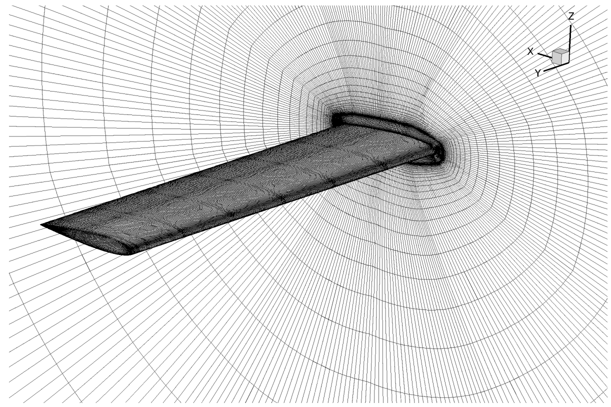
Fig. 11 Cases with scaling the angle of attack design variable - gradients comparison $\mathcal{G}_r = \frac{||\mathcal{G}_g||_2}{|\mathcal{G}_a|}$

Case	CH10SM ($\alpha = 8°$, $f = 10$)	NACA 0012 ($\alpha = 8°$, $f = 10$)	CH10SM ($\alpha = 25°$, $f = 0.1$)
α	26.54°	50°	24.9606°
C_L	2.1944	2.3825	2.2075
C_D	0.9338	2.7409	0.9375
C_L/C_D	2.350	0.869	2.355
Area, A	0.1289	0.08126	0.1359
iterations, k	61	21	131

Table 4 Mean lift maximization summary - scaled cases



(a) Parameterized wing with FFD volume



(b) CFD grid

Fig. 12 NACA 0012 rectangular wing optimization

V. Three-Dimensional Optimization Problems

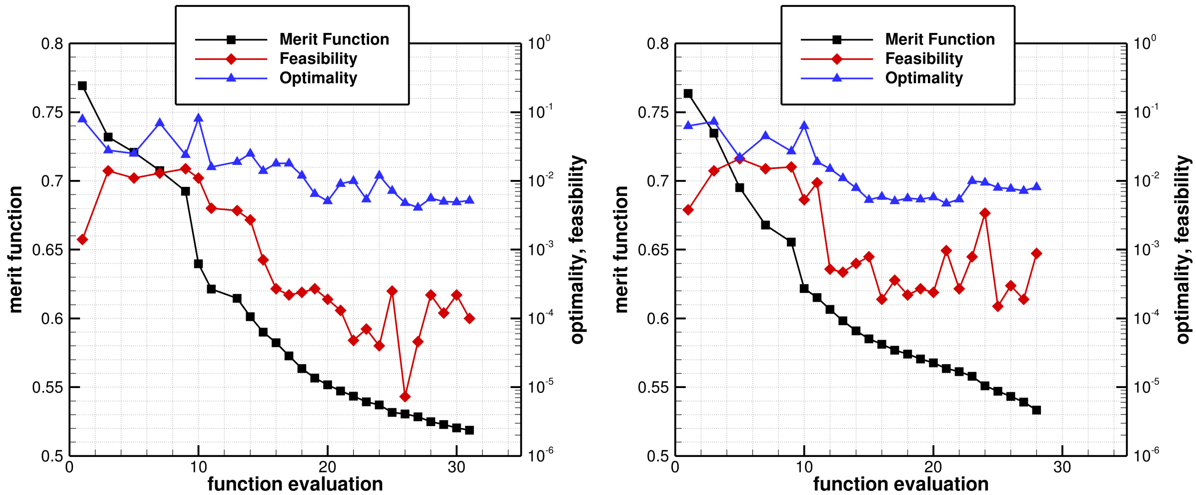
We use a NACA0012 rectangular wing with an aspect ratio of 8 as the baseline geometry. The objective is to minimize the mean drag at a mean lift (i.e. computed as $C_L S$, where S is the area of the half-wing) of 1.65 in chord lengths squared. We use the same flow conditions as the airfoil design studies (i.e. Re = 800, M = 0.2). Figure 12 shows the parameterized baseline geometry and the CFD grid. The CFD grid has approximately 7.1 million nodes divided over 1152 blocks with $11 \times 11 \times 51$ nodes in each block. The off-wall spacing is 2.77×10^{-5} based on the initial chord length. The grid wraps a hemisphere around the geometry and the far-field boundary is placed 40 chord lengths radially from the geometry. The flow problem was time-marched for 7000 time steps with the last 3000 time steps used as the control window. The first 4000 time steps are computed using a time step of $\Delta t = 0.004$, and a time step of $\Delta t = 0.002$ is used for the control window (i.e. $t = [16, 22]$). Varying the time step at the adjusting phase of the flow allows us to improve the turnaround time for these problems.

In the first study, the optimizer is allowed to modify the section chord to introduce taper in addition to the freedom to modify the section shape and aerodynamic twist about the quarter-chord line. This case is similar to the study in [19], where only section shape and aerodynamic twist were allowed to vary. The current study will be referred to as the *tapered case* throughout the paper and the study in [19] will be referred to as the *simple case*. In a second study, we extended the tapered case further by imposing a projected area constraint equal to the projected area of the initial wing (i.e. $A_{\text{prj}} = 4.0$). We refer to this study as the *projected area case* throughout the paper.

In terms of constraints, the section thickness, t_c, is constrained to not decrease by more than 25% of the initial value, t_{c0}. The chord is allowed to vary between $0.5c_0$ and $2c_0$, where c_0 is the initial chord length of the section. At the wing tip, the chord is allowed to vary between $0.5c_0$ and c_0. The aerodynamic twist about the quarter chord is allowed to vary between $\pm 15°$. The geometry is controlled with 6 FFD sections along the span, leading to 144 and 143 geometric design variables for the first and second study respectively in addition to the angle of attack. For the second study the twist freedom at the root of the wing is removed hence the difference in the geometric design variables.

We show the optimization histories of the two cases in Figure 13. For the tapered case, the optimizer was terminated after 31 function evaluations (i.e. 25 optimization iterations). Despite the partial convergence, the optimality and feasibility were reduced by an order of magnitude, and the objective was reduced by more than 32%. In Figure 14a, we show the evolution of the planform of the wing indicating the changes in taper as the optimizer proceeds. Figure 15a also shows the thickness and twist variations at selected sections of the final geometry. We observed that the sections are similar to the two-dimensional cases, in that, the optimizer introduces a flap-like feature to allow it to achieve the lift requirement, while a reduction in the wetted area in combination with the small wing tip allows it to reduce the drag.

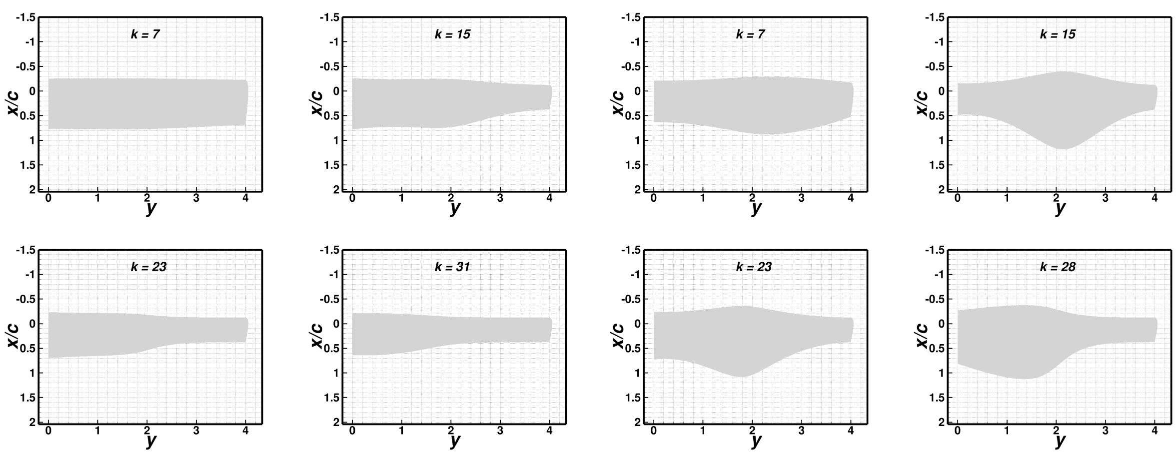
In Figure 13b we show the optimization history of the projected area case. At the time of writing the optimizer had completed 28 function evaluations (i.e. 23 optimization iterations) and had not been terminated yet. The reduction in the optimality is close to an order of magnitude and feasibility has dropped by an order of magnitude. The objective has been reduced by more 30% and the projected area constraint is active. The evolution of the planform (see Figure 14a) shows the optimizer exploits the taper freedom given and increases the chord from the root to the mid section in order to satisfy the projected area constraint. Here as well, the optimizer adds a flap-like feature throughout the sections, as



(a) Tapered case

(b) Projected area case

Fig. 13 Optimization convergence history



(a) Tapered case

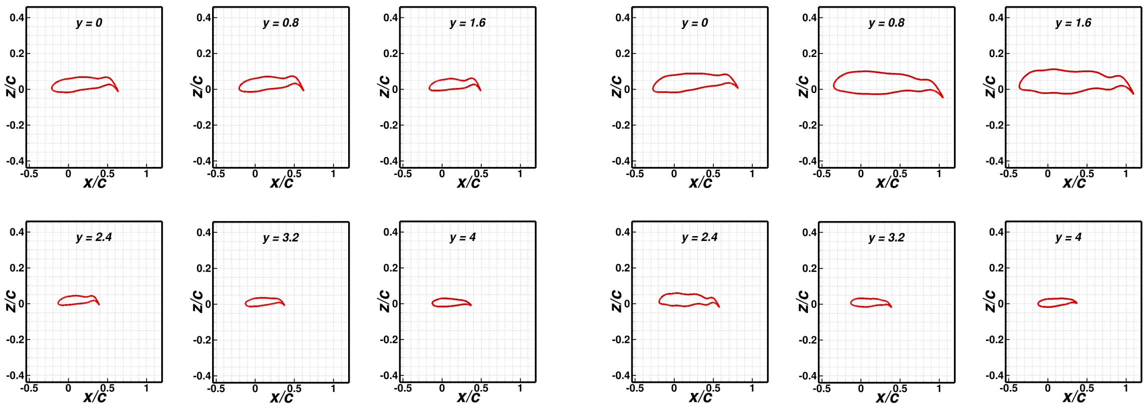
(b) Projected area case

Fig. 14 Planform view versus function evaluation

shown in Figure 15b, to help maintain the required lift. Compared to the simple case, the projected area case has a lower drag at the current optimization iteration. The tapered case will be restarted to see if we can obtain a deeper convergence and further improvement in the objective. For both the tapered and projected area optimization cases, we see a reduction in the amplitude of the oscillations of the objective function in the control window (see Figure 16), similar to the simple case and the two-dimensional lift-constrained drag minimization problem. Although the control window should be increased to reflect the modified behaviour of the optimized geometry with the projected area constraint, the present problem formulation is sufficient to illustrate the performance of the algorithm and the trends in the geometric changes that lead to drag reduction for these flow conditions.

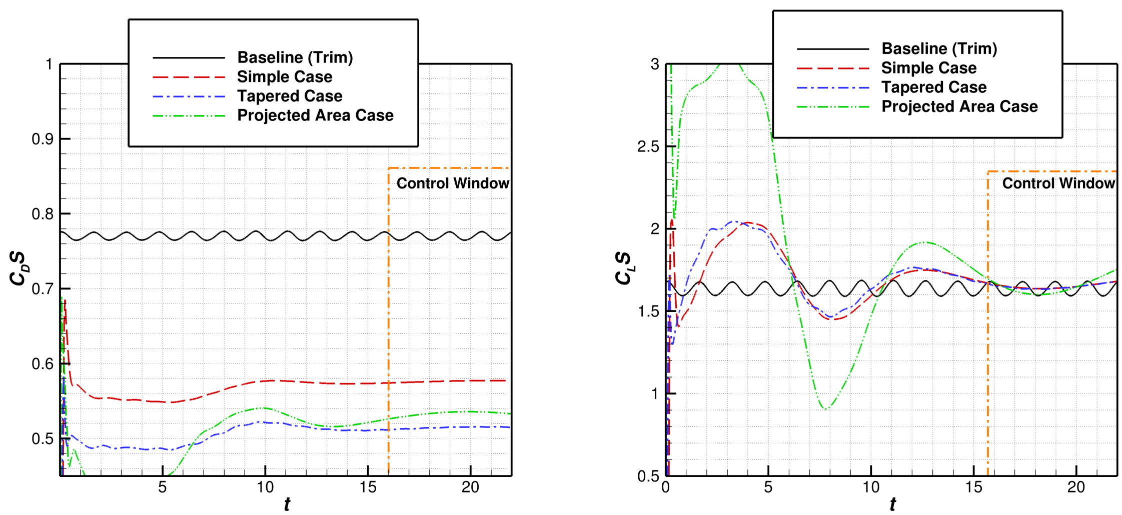
VI. Discussions and Conclusions

We have presented a methodology for solving aerodynamic shape optimization problems for unsteady flows and applied it to a range of design problems for two- and three-dimensional laminar flows. For the two-dimensional lift-constrained drag minimization and lift-to-drag ratio maximization problems, the optimizer flattens the upper surface of the airfoil to keep the flow attached as long as possible which helps reduce the drag associated with flow separation.



(a) Tapered case (b) Projected area case

Fig. 15 Optimized wing sections



(a) Drag history (objective) (b) Lift history

Fig. 16 Aerodynamic quantities for optimized geometries

Case	Baseline	Simple	Tapered	Projected Area
α	12°	9.19°	8.70°	3.08°
$C_L S$	1.65	1.6495	1.6489	1.6463
$C_D S$	0.770	0.5761	0.5184	0.5330
$\Delta C_D S$	-	25.18%	32.68%	30.78%
C_L/C_D	2.143	2.863	3.181	3.089
A_{prj}	4.0	4.0	2.56	3.999

Table 5 NACA 0012 rectangular wing optimization summary

To increase lift or meet the lift target the optimizer pushes the airfoil camber towards the trailing edge, which leads to the formation of a flap-like feature. The flap-like feature also plays a minor role in reducing the amplitude of the oscillations in the aerodynamic quantities for the lift-constrained drag minimization problem. The optimized airfoils obtained in the lift-constrained drag minimization study are nearly identical to the results in [19], although there are differences in the geometry control system, the number of design variables, and the bounds on the geometric constraints. Qualitatively, the lift-constrained drag minimization case compares well with results obtained in [14] with the NACA0012 at a Reynolds number 1000. The study showed a drag reduction of about 18%, and the amplitude of the oscillations in the aerodynamic quantities was reduced compared to the baseline. The are some key differences in the problem definition, in particular the lift-constraint, the angle of attack design variable, the thickness constraints and their use of Non-Uniform B-spline (NURBS) curves to parameterize the airfoil.

For the lift-maximization problem, applying some form of scaling to the design variables and having a robust mesh-movement algorithm is necessary for improving convergence and minimizing mesh movement failures. We also observed that for the lift-constrained drag minimization and the lift-to-drag ratio maximization problems, the optimized geometries are nearly identical when the optimization was started with two different initial airfoils. Therefore, these two optimization problems are likely to be convex or uni-modal. For the lift-maximization problem we observed multiple optimized airfoils based on the initial airfoil, initial angle of attack, and the design variable scaling; therefore, this optimization problem is likely to be non-convex or multi-modal.

For the three-dimensional problems, the optimizer exploits all of the geometric freedom given to reduce the drag while satisfying the constraints imposed. Similar to the two-dimensional problems, we observed a flap-like feature throughout the sections, which helps to maintain the lift target with the optimizer loading the wing highly towards the root to minimize the induced drag. The oscillations in the amplitude of the aerodynamic quantities are also reduced for the three-dimensional cases and we observed a drag reduction of more than 32% and 30% for the tapered and projected area cases respectively. Further, adding geometric freedom and constraints incrementally has allowed us to test that the various components of our methodology behave as expected. For the three-dimensional problems, a flow solve costs on the average $\sim$ 13.5s per time step and the flow adjoint costs on the average $\sim$ 4.5s per time step using 1152 Intel Skylake processors. The processors are distributed over 29 compute nodes, and they are clocked at a speed of 2.4GHz. Each compute node has 40 cores with 202GB of RAM and an EDR InfiniBand interconnect, which can transfer data at speeds ranging from 100Gb/s to 300Gb/s. For 7000 time steps, the cost per function evaluation is about $\sim$ 44 hours, this includes one complete flow solve and two flow adjoint evaluations (i.e. one for the objective function and another for the lift constraint). For each function evaluation, the cost of the mesh-movement problem is about 40s, and the mesh adjoint costs about 80s using PETSc[33–35]. Using 30 function evaluations as the baseline, an optimization can be completed in about 60 to 120 days when the time between restarts due to waiting in the HPC queue is accounted for. Further, the hard drive requirement is in the order of 2TB for each three-dimensional optimization study. Although these studies are computationally expensive, we hope that our work will encourage other researchers to study similar unsteady flow optimization problems as computing power increases and becomes cheaper. As part of our future work, we hope to investigate some practical problems such as optimizing wings in the presence of transonic buffet to improve off-design performance.

We conclude by making the following key observations;

- the optimization problems and results we have presented can be used to test and characterize other aerodynamic shape optimization methodologies for unsteady flows
- the geometry control and the efficient mesh-movement approach, which has been previously validated for steady flows, is also effective for unsteady flows with substantial geometric changes
- the cost of the unsteady flow adjoint problem is about a third of the cost of the unsteady flow problem for the cases we considered
- for fixed wing applications, the cost of the mesh movement and mesh adjoint problems are insignificant compared to the flow and flow adjoint problems since the geometry is updated once per function evaluation
- some interesting geometric features have been identified that improve various objective functions and reduce unsteadiness, which may have applications where Reynolds numbers are very low

Acknowledgements

We are grateful for the financial support from the Natural Sciences and Engineering Research Council (NSERC), Centre for Aerial Robotics Research and Education (CARRE) and the Dr. Lorne Heuckroth Scholarship towards this work. All computations were performed on the Niagara supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation, the Government of Ontario, Ontario Research Fund - Research Excellence, and the University of Toronto.

References

[1] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.

[2] Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1, 1998, pp. 213–237.

[3] LeDoux, S. T., Vassberg, J. C., Young, D. P., Fugal, S., Kamenetskiy, D., Huffman, W. P., Melvin, R. G., and Smith, M. F., "Study Based on the AIAA Aerodynamic Design Optimization Discussion Group Test Cases," *AIAA Journal*, Vol. 53, No. 7, 2015, pp. 1910–1935.

[4] Reist, T. A., Koo, D., Zingg, D. W., Bochud, P., Castonguay, P., and Deblond, D., "Cross-Validation of High-Fidelity Aerodynamic Shape Optimization Methodologies for Aircraft Wing-Body Optimization," *AIAA Journal*, Vol. 58, No. 6, 2020.

[5] Yee, K., Kim, K., and Lee, D., "Aerodynamic Shape Optimization of Rotor Airfoils Undergoing Unsteady Motion," AIAA Paper 99-3107, 1999.

[6] He, J., Glowinski, R., Meltcalfe, R., Norlander, A., and Periaux, J., "Active Control and Drag Optimization for Flow Past a Circular Cylinder," *Journal of Computational Physics*, Vol. 163, 2000, pp. 83–117.

[7] Nadarajah, S., and Jameson, A., "Optimal Control of Unsteady Flows using a Time Accurate Method," AIAA Paper 2002-5436, 2002.

[8] Nadarajah, S., and Jameson, A., "Optimum Shape Design for Unsteady Flows with a Time Accurate Continuous and Discrete Adjoint Method," *AIAA Journal*, Vol. 45, No. 7, 2007, pp. 1478–1491.

[9] Nadarajah, S., and Jameson, A., "Optimum Shape Design for Unsteady Three-Dimensional Viscous Flows using a Non-Linear Frequency Domain Method," *Journal of Aircraft*, Vol. 44, No. 5, 2007, pp. 1513–1527.

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

[11] Nielsen, E. J., Diskin, B., and Yamaleev, N. K., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, Vol. 48, No. 6, 2010, pp. 1195–1206.

[12] Rumpfkeil, M., and D. W. 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.

[13] Rumpfkeil, M., and Zingg, D. W., "A Hybrid Algorithm for Far-Field Noise Minimization," *Computers and Fluids*, Vol. 39, No. 9, 2010, pp. 1516–1528.

[14] Srinath, D. N., and Mittal, S., "Optimal aerodynamic design of airfoils in unsteady viscous flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, 2010, pp. 1976, 1991.

[15] Economon, T. D., Palacios, F., and Alonso, J. J., "Unsteady Continuous Adjoint Approach for Aerodynamic Design on Dynamic Meshes," *AIAA Journal*, Vol. 53, No. 9, 2015, pp. 2437, 2453.

[16] Vishnampet, R., Bodony, D. J., and Freund, J. B., "A practical discrete-adjoint method for high-fidelity tubulence simulations," *Journal of Computational Physics*, Vol. 285, 2015, pp. 173, 192.

[17] Nimmagadda, S., Economon, T. D., Illario da Silva, C. R., Alonso, J. J., Zhou, B. Y., and Albring, T., "Low-cost unsteady discrete adjoints for aeroacoustic optimization using temporal and spatial coarsening," AIAA Paper 2018-1911, 2018.

[18] Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal Fluid Mechanics*, Vol. 64, 1974, pp. 97–110.

[19] Apponsah, K. P., and Zingg, D. W., "Aerodynamic Shape Optimization for Unsteady Flows: Some Benchmark Problems," AIAA Paper 2020-0541, 2020.

[20] Zingg, D. W., Leung, T. M., Diosady, L., Truong, A., Elias, S., and Nemec, M., "Improvements to a Newton-Krylov Adjoint Algorithm for Aerodynamic Optimization," AIAA Paper 2005-4857, 2005.

[21] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for Large-Scale Constrained Optimization," *SIAM Journal of Optimization*, Vol. 12, 2002, pp. 979–1006.

[22] Gagnon, H., and Zingg, D. W., "Two-Level Free-Form and Axial Deformation for Exploratory Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 53, No. 7, 2015, pp. 2015–2026.

[23] Lee, C., Koo, D., and Zingg, D. W., "Comparison of B-spline Surface and Free-form Deformation Control Methods for Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 55, No. 1, 2017, pp. 228–240.

[24] Hicken, J. E., and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 401–413.

[25] Hicken, J. E., and Zingg, D. W., "A Parallel Newton-Krylov Solver for the Euler Equations Discretized Using Simultaneous Approximation Terms," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2273–2786.

[26] Osusky, M., and Zingg, D. W., "Parallel Newton-Krylov-Schur Solver for Navier-Stokes Equations Discretized Using Summation-By-Parts Operators," *AIAA Journal*, Vol. 51, No. 12, 2013.

[27] Boom, P. D., and Zingg, D. W., "Time-Accurate Flow Simulations Using an Efficient Newton-Krylov-Schur Approach with High-Order Temporal and Spatial Discretization," AIAA Paper 2013-0383, 2013.

[28] Pulliam, T. H., and Zingg, D. W., *Fundamental Algorithms in Computational Fluid Dynamics*, Springer-Verlag, 2014.

[29] Del Rey Fernández, D. C., Hicken, J., and Zingg, D. W., "Review of Summation-By-Parts Operators with Simultaneous Approximation Terms for the Numerical Solution of Partial Differential Equations," *Computers and Fluids*, Vol. 95, 2014, pp. 171–196.

[30] Saad, Y., and Sosonika, M., "Distributed Schur complement techniques for general sparse linear systems," *SIAM Journal on Scientific Computing*, Vol. 21, 1999, pp. 1337–1357.

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

[32] Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific and Statistical Computing*, Vol. 14, 1993, pp. 461–469.

[33] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[34] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Web page," `https://www.mcs.anl.gov/petsc`, 2021. URL `https://www.mcs.anl.gov/petsc`.

[35] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.15, Argonne National Laboratory, 2021. URL `https://www.mcs.anl.gov/petsc`.