

Adjoint-based Methodology for Time-Dependent Optimization

N. K. Yamaleev,^{*}B. Diskin,^{†‡}and E. J. Nielsen[§]

This paper presents a discrete adjoint method for a broad class of time-dependent optimization problems. The time-dependent adjoint equations are derived in terms of the discrete residual of an arbitrary finite volume scheme which approximates unsteady conservation law equations. Although only the 2-D unsteady Euler equations are considered in the present analysis, this time-dependent adjoint method is applicable to the 3-D unsteady Reynolds-averaged Navier-Stokes equations with minor modifications. The discrete adjoint operators involving the derivatives of the discrete residual and the cost functional with respect to the flow variables are computed using a complex-variable approach, which provides discrete consistency and drastically reduces the implementation and debugging cycle. The implementation of the time-dependent adjoint method is validated by comparing the sensitivity derivative with that obtained by forward mode differentiation. Our numerical results show that $O(10)$ optimization iterations of the steepest descent method are needed to reduce the objective functional by 3-6 orders of magnitude for test problems considered.

I. Introduction

Time-dependent optimization problems arise in many areas in science and engineering including various flow control applications such as controlling flow separation, airframe vibration, noise level, transition to turbulence, etc., as well as design optimization problems for essentially unsteady flows, including design and shape optimization of helicopter rotors, turbomachinery blades, aircraft wings, and other configurations. The overall complexity of these time-dependent optimization problems is much higher than that of steady-state aerodynamic optimization problems, which is one of the main reasons why this class of problems has not been used in real-life applications. Continuously expanding computer capabilities now allow more attention to be devoted to numerical solution of these class of time-dependent optimal control and design optimization problems. These problems can be considered as minimization of an appropriate cost functional (e.g., lift, drag, etc.). The resulting control laws or design variables are obtained by solving the corresponding time-dependent optimal control or design problems by using appropriate optimization algorithms.

Among various optimization techniques available in the literature (see, e.g., [1]), the adjoint method has recently grown in popularity, rapidly becoming one of the most widely used techniques for solving a variety of steady and unsteady optimization problems. The adjoint methodology is particularly attractive for optimal control/design problems, which include a large number of control variables, yet relatively few constraints. In contrast to a classical forward mode differentiation approach which requires solutions as twice as many flow problems as the number of the corresponding control variables, the adjoint methodology has the advantage of computing the cost functional gradients at a fixed expense independent of the number of control/design variables. This property of the gradient methods based on the adjoint formulation make them very well suited for steady aerodynamic design optimization problems.²⁻⁵ Although the adjoint-based methods have been successfully used for problems of optimal design within the steady-state aerodynamics, applications of the adjoint formulation to essentially time-dependent optimal control/design problems are still lacking. In,⁶ the 2-D continuous time-dependent adjoint incompressible Navier-Stokes equations and optimality conditions have been derived. This continuous adjoint-based method has been successfully used

^{*}Associate Professor, North Carolina A&T State University, Member AIAA. E-mail: nkyamale@ncat.edu

[†]Senior Research Scientist, National Institute of Aerospace, E-mail: bdiskin@nianet.org

[‡]Visiting Associate Professor, MAE, University of Virginia, Member AIAA.

[§]Research Scientist, NASA Langley Research Center, Senior Member AIAA, E-mail: Eric.J.Nielsen@nasa.gov

for solving the problem of boundary-layer instability suppression through wave cancellation. Nadarajah and Jameson⁷ derived and applied the time accurate continuous and discrete adjoint equations to the shape optimization of an oscillating airfoil in an 2-D inviscid transonic flow. In,⁸ a gradient method based on the discrete adjoint equations and the corresponding boundary conditions in the frequency domain has been developed. This approach significantly reduces the computational cost for shape optimization of a 3-D wing oscillating at a constant frequency. Note, however, that this technique is applicable only for periodic problems and its efficiency strongly depends on the number of harmonics in the time-dependent solution. Discrete adjoint-based methods operating directly in the time domain have been developed for the 2-D compressible Euler and Navier-Stokes equations in⁹ and,^{10,11} respectively.

The time-dependent adjoint-based methods mentioned above can be divided into two groups. The first one^{7-9,11} is developed for optimization problems with the design/control variables that are independent of time, while the second one^{6,10} involves the design/control variables that depend on time. In this paper, we develop a general discrete adjoint-based optimization methodology which is directly applicable to both classes of problems. This time-dependent optimization methodology can be directly applied to solving a very broad spectrum of time-dependent optimal control problems, where the control variables are in general time-dependent (e.g., the displacement of an actuator diaphragm or the velocity distribution at the actuator orifice, etc.) and design optimization problems where the design variable are in general do not depend on time (e.g., shape of a helicopter rotor or an aircraft wing, etc.).

The paper is organized as follows. In Section II, we present the continuous and discrete state equations. In Section III, the discrete time-dependent optimization problem is described. In Section IV, the general discrete time-dependent adjoint equations are derived. Section V discusses a technique for forming discrete adjoint operators by using complex variables. In Section VI, we present two test problems used for validating the developed time-dependent optimization methodology. We draw conclusions and present our plans for the future in Section VII.

II. Governing Equations

We consider the time-dependent, two-dimensional Euler equations describing the unsteady, inviscid compressible flow. The Euler equations written in the integral conservation law form are given by:

$$\frac{\partial V\mathbf{U}}{\partial t} + \oint_{\Gamma} \mathbf{F} \cdot \mathbf{n} d\Gamma = \mathbf{0} \quad (1)$$

where \mathbf{n} is the outward unit normal vector of the control volume with boundary Γ , V is the control volume, \mathbf{U} is the vector of conserved variables averaged over the control volume, and \mathbf{F} is the Cartesian inviscid flux vector.

The governing equations (1) are discretized by using a node-centered finite-volume scheme, where solution values are stored at the mesh nodes. The control volume around each grid node is constructed by connecting the centroids of the primal-mesh cells with midpoints of the surrounding edges. The discretized Euler equations including the boundary conditions can be written as follows:

$$\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}(\mathbf{Q}^n) = \mathbf{0}, \quad (2)$$

where $\mathbf{Q} = V\mathbf{U}$, and \mathbf{R} is the spatial undivided residual of the discretization which approximates the contour integral in Eq. (1). It should be noted that the above discrete formulation (2) is very general and can be applied to a broad class of time-dependent PDEs discretized using not only finite volume, but also finite difference and finite element schemes. The flux $\hat{\mathbf{F}}$ in the discretized integral is approximated using Roe's approximate Riemann solver

$$\hat{\mathbf{F}} = \frac{1}{2} \left[\hat{\mathbf{F}}_L + \hat{\mathbf{F}}_R - |\mathbf{A}| (\mathbf{U}_L - \mathbf{U}_R) \right], \quad (3)$$

where $\hat{\mathbf{F}}_L$ and $\hat{\mathbf{F}}_R$ are the "left" and "right" normal fluxes at the edge midpoint, \mathbf{U}_L and \mathbf{U}_R are the "left" and "right" reconstructed values of the solution vector at the edge midpoint, obtained from some polynomial approximation defined on each control volume, $|\mathbf{A}|$ is the Roe averaged matrix. In Eq. (2), the time derivative has been approximated using the implicit first-order backward-difference (BDF-1) formula. Note that second-order BDF formula as well as higher order implicit Runge-Kutta methods can also be used in the present formulation with minor modifications.

In the present paper, we consider only inviscid flow problems because our primary objective is to develop a time-dependent adjoint-based optimization methodology that is applicable to a broad spectrum of nonlinear state equations. Generalization of this methodology to the unsteady Reynolds-averaged Navier-Stokes (RANS) equations coupled with either one- or two-equation turbulence model is quite straightforward. In this case, only the flux residual \mathbf{R} should be changed, while the adjoint equations, which will be presented in Section IV, remain unchanged. Note, however, that for the RANS equations, questions related to robustness of the present adjoint-based methodology require special investigation, which are outside the scope of the present paper.

III. Discrete Time-Dependent Optimization Problem

We consider the following discrete time-dependent optimization problem:

$$\min_{\mathbf{D} \in \mathcal{D}_a} f(\mathbf{D}), \quad f(\mathbf{D}) = \sum_{n=1}^N f^n(\mathbf{D}) \Delta t, \quad f^n(\mathbf{D}) = F_{\text{obj}}^n(\mathbf{Q}^n(\mathbf{D})) + F_{\text{reg}}^n(\mathbf{U}, \mathbf{D}) \quad (4)$$

where \mathbf{D} is a vector of the control or design variables which in general depends on time, N is the total number of time steps over which the control \mathbf{D} is active, \mathbf{Q} is the solution of the unsteady, compressible Euler equations, F_{obj}^n is a part of the cost functional that represents the flow control objective, and F_{reg}^n is a regularization term, typically some weighted norm of the control variable. Note that this setting of the problem remains valid if \mathbf{D} does not depend on time. The above formulation (4) is very general and directly applicable for both time-dependent optimal control problems (e.g., active flow control via synthetic jet actuation) and aerodynamic design optimization for unsteady flows (e.g., design of a turbomachinery blade), and others. The set of admissible controls, \mathcal{D}_a , depends on specifics of the target physical system (e.g., how much suction and blowing can the actuators provide, or admissible length and thickness of a blade, etc.), but it should also ensure the existence of a solution of the optimization problem (4).

To reduce the complexity of the optimization problem, without loss of generality, we assume that the objective functional f is a scalar quantity. In the present analysis, we consider the following discrete convex functional:

$$F_{\text{obj}}^n = \beta_1 \left(C_L^n - C_{L_{\text{target}}}^n \right)^2 + \beta_2 \left(C_D^n - C_{D_{\text{target}}}^n \right)^2 \quad (5)$$

where $C_{L_{\text{target}}}^n$ and $C_{D_{\text{target}}}^n$ are given time-dependent target lift and drag coefficients, respectively, which are integrals of the normal and tangential components of the stress tensor over the controlled boundary surface.

The control variables \mathbf{D} have a precise physical meaning (e.g., the the Mach number or angle of attack as a function of time, etc.) and should remain bounded and be continuous in time. These physical constraints are incorporated into the optimization problem through the regularization/penalty term in the cost functional Eq. (4), which limits the size of the control. The regularization/penalty term F_{reg}^n is chosen as follows

$$F_{\text{reg}}^n = \frac{\alpha_1}{2} [\mathbf{D}^n]^T \mathbf{D}^n + \frac{\alpha_2}{2} \frac{1}{\Delta t^2} (\mathbf{D}^n - \mathbf{D}^{n-1})^T (\mathbf{D}^n - \mathbf{D}^{n-1}) \quad (6)$$

where α_1 and α_2 are nonnegative parameters that can be used to adjust the relative weights of the regularization terms appearing in the functional (6). The particular form of the penalty term (6) limits not only the magnitude of the control, but also the rate, at which the control changes, to provide the necessary smoothness of the control. The presence of the second term in the cost functional can also be interpreted as a constraint on the maximum kinetic energy generated by the control system, which is directly related to the energy consumption required for its operation.

It should be noted that the same penalty technique outlined above can be used to impose a more general nonlinear side constraints involving the state variable \mathbf{U} . If the optimization problem (4) is subject to the side constraint $\Phi(\mathbf{U}, \mathbf{D}) \leq 0$, then the following penalty term can be added to the objective functional to enforce this constraint:

$$F_{\text{reg}} = \alpha (\max[0, \Phi(\mathbf{U}, \mathbf{D})])^2, \quad (7)$$

where α is a positive user-defined parameter, and Φ is a continuously differentiable function of its arguments. Note that the above penalty term is continuously differentiable and active only when the constraint is violated, i.e., when $\Phi(\mathbf{U}, \mathbf{D}) > 0$. The above penalization guarantees that the constraint $\Phi(\mathbf{U}, \mathbf{D}) \leq 0$ is met if $\beta \rightarrow +\infty$. In practice, the parameter α can be increased during the iterative process to make sure that the side constraint is satisfied.

Currently, only matching objective functionals with the global extremum have been considered. Because, there are no spurious extrema in all test problems presented herein, the regularization term is set equal to zero. Though this simplified formulation works well for the Euler equations considered in this paper, for problems involving essentially nonlinear one- or two-equation turbulence models, the regularization term may play an important role and should be included into the optimization procedure.

IV. Time-dependent Adjoint Formulation

The discrete time-dependent optimization problem (4) is solved by using the method of Lagrange multipliers which is used to enforce the governing equations and the corresponding boundary conditions (2) as constraints. The discrete Lagrangian functional is defined as follows:

$$L(\mathbf{D}, \mathbf{Q}, \mathbf{\Lambda}) = \sum_{n=1}^N f^n \Delta t + \sum_{n=2}^N [\mathbf{\Lambda}^n]^T \left(\frac{\mathbf{Q}^n - \mathbf{Q}^{n-1}}{\Delta t} + \mathbf{R}^n \right) \Delta t + [\mathbf{\Lambda}^1]^T (\mathbf{Q}^1 - \mathbf{Q}^{\text{in}}), \quad (8)$$

where $\mathbf{\Lambda}$ is a vector of Lagrange multipliers or costate variables, $n = 1$ corresponds to the initial moment of time, \mathbf{Q}^{in} is the initial condition for the Euler equations, f^n is the objective functional given by Eq. (5), and $\mathbf{R}^n = \mathbf{R}(\mathbf{Q}^n, \mathbf{D})$ is the spatial undivided residual. Note that the first two terms in the Lagrangian are scaled by Δt , so that they approximate the corresponding time integrals in the continuous Lagrangian. Therefore, the discrete Lagrangian approaches its continuous counterpart as the number of time steps N increases. Furthermore, the scalar product of the costate vector and the vector of the governing equations in Eq. (8) can be interpreted as the integral over the computational domain, which again approximates the continuous Lagrangian.

The sensitivity derivative is obtained by differentiating the Lagrangian with respect to \mathbf{D} , which yields

$$\begin{aligned} \frac{dL}{d\mathbf{D}} &= \sum_{n=1}^N \left(\frac{\partial f^n}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{Q}^n}{\partial \mathbf{D}} \right]^T \frac{\partial f^n}{\partial \mathbf{Q}^n} \right) \Delta t + \left[\frac{\partial \mathbf{Q}^N}{\partial \mathbf{D}} \right]^T \left(\frac{\mathbf{\Lambda}^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N} \right]^T \mathbf{\Lambda}^N \right) \Delta t \\ &+ \sum_{n=2}^{N-1} \left[\frac{\partial \mathbf{Q}^n}{\partial \mathbf{D}} \right]^T \left(\frac{\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \mathbf{\Lambda}^n \right) \Delta t \\ &- \left[\frac{\partial \mathbf{Q}^1}{\partial \mathbf{D}} \right]^T \mathbf{\Lambda}^1 + \left(\left[\frac{\partial \mathbf{Q}^1}{\partial \mathbf{D}} \right]^T - \left[\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}} \right]^T \right) \mathbf{\Lambda}^1 + \sum_{n=2}^N \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} \right]^T \mathbf{\Lambda}^n \Delta t \end{aligned} \quad (9)$$

Regrouping the terms, Eq. (9) can be recast as follows:

$$\begin{aligned} \frac{dL}{d\mathbf{D}} &= \sum_{n=1}^N \frac{\partial f^n}{\partial \mathbf{D}} \Delta t + \left[\frac{\partial \mathbf{Q}^N}{\partial \mathbf{D}} \right]^T \left(\frac{\mathbf{\Lambda}^N}{\Delta t} + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N} \right]^T \mathbf{\Lambda}^N + \frac{\partial f^N}{\partial \mathbf{Q}^N} \right) \Delta t \\ &+ \sum_{n=2}^{N-1} \left[\frac{\partial \mathbf{Q}^n}{\partial \mathbf{D}} \right]^T \left(\frac{\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}}{\Delta t} + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \mathbf{\Lambda}^n + \frac{\partial f^n}{\partial \mathbf{Q}^n} \right) \Delta t \\ &+ \left[\frac{\partial \mathbf{Q}^1}{\partial \mathbf{D}} \right]^T \left(\frac{\mathbf{\Lambda}^1 - \mathbf{\Lambda}^2}{\Delta t} + \frac{\partial f^1}{\partial \mathbf{Q}^1} \right) \Delta t - \left[\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}} \right]^T \mathbf{\Lambda}^1 + \sum_{n=2}^N \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} \right]^T \mathbf{\Lambda}^n \Delta t \end{aligned} \quad (10)$$

For problems with a large number of control/design variables, it is desirable to avoid the calculation of $\partial \mathbf{Q} / \partial \mathbf{D}$ in the optimization procedure, because it requires multiple solves of the primary problem. Taking into account that so far, no constraints have been imposed on the Lagrange multipliers, the $\partial \mathbf{Q} / \partial \mathbf{D}$ term can be eliminated from Eq. (10) by setting the second, third, and fourth terms on the right hand side equal to zero, which results in the following adjoint equations for determining the Lagrange multipliers:

$$\frac{1}{\Delta t} \mathbf{\Lambda}^N + \left[\frac{\partial \mathbf{R}^N}{\partial \mathbf{Q}^N} \right]^T \mathbf{\Lambda}^N = - \frac{\partial f^N}{\partial \mathbf{Q}^N} \quad (11)$$

$$\frac{1}{\Delta t} (\mathbf{\Lambda}^n - \mathbf{\Lambda}^{n+1}) + \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}^n} \right]^T \mathbf{\Lambda}^n = - \frac{\partial f^n}{\partial \mathbf{Q}^n}, \quad 2 \leq n \leq N-1 \quad (12)$$

$$\frac{1}{\Delta t} (\mathbf{\Lambda}^1 - \mathbf{\Lambda}^2) = - \frac{\partial f^1}{\partial \mathbf{Q}^1}, \quad (13)$$

Equations (11) and (13) are initial and terminal conditions for the costate variables. Equations (12) represent a linear system of equations for the costate variables, which are solved backward in time. Once Eqs. (11-13)

have been solved, the vector of Lagrange multipliers Λ^n can be used to evaluate the last two terms in Eq. (10). As a result, the sensitivity derivative can be calculated as follows:

$$\frac{dL}{d\mathbf{D}} = \sum_{n=1}^N \frac{\partial f^n}{\partial \mathbf{D}} \Delta t + \sum_{n=2}^N \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}} \right]^T \Lambda^n \Delta t - \left(\frac{\partial \mathbf{Q}^{\text{in}}}{\partial \mathbf{D}} \right)^T \Lambda^1, \quad (14)$$

where $\frac{\partial f^n}{\partial \mathbf{D}}$ and $\frac{\partial \mathbf{R}^n}{\partial \mathbf{D}}$ are calculated by using \mathbf{Q}^n stored during the forward sweep in time.

The minimum of the functional is found by using the steepest descent method in which each step of the optimization cycle is taken in the negative gradient direction

$$D_m^{(k+1)} = D_m^{(k)} - \tau_m \frac{dL}{dD_m}, \quad (15)$$

where τ_m is the step size for the m -th component of the vector \mathbf{D} , k is the number of optimization cycle, D_m is the m -th component of the vector \mathbf{D} . The sensitivity derivative dL/dD in Eq. (15) is determined using Eq. (14) which requires the solution of the adjoint equations (11-13). During the solution of the adjoint equations that are integrated backward in time, the sensitivity derivative at each time step is computed and added to its value at the previous time step. At $n = 1$, the complete sensitivity derivative is available and used in Eq. (15) for calculating a new value of the control variable $\mathbf{D}^{(k+1)}$. Then the entire optimization cycle is repeated until $|L^{(k+1)} - L^{(k)}| < \epsilon$, where ϵ is a given tolerance. This optimization algorithm is quite sensitive to the step size τ and has been selected because of its simplicity. Other more efficient and robust gradient-based methods, such as conjugate gradient or quasi-Newton methods, can also be easily coupled with the time-adjoint formulation used in the present analysis.

As has been mentioned above, the selection of the optimization step size τ in Eq. (15) has a very strong effect on the number of optimization iterations required to reach an optimum solution. Large step sizes may result in instabilities in optimization iterations, whereas small step sizes provide stability, but drastically slow down the convergence. Therefore, the optimization step size in Eq. (15) is selected adaptively to maximize the convergence rate. In the present analysis, the following algorithm for choosing τ is used:

- 1) Each optimization cycle continues until a solution with a smaller integrated cost functional, $\sum_{n=1}^N f^n \Delta t$, has been found.
- 2) For a trial vector of τ , the new vector \mathbf{D} and the corresponding \mathbf{U} are computed on all time levels.
- 3) The n -th component of the vector τ , τ_n , is changed if one of the following two events occurs:
- 4) if the local cost functional increases, i.e., $f_{\text{trial}}^n > f^n$, then τ_n is decreased, $\tau_n = 0.5\tau_n$.
- 5) or if the local cost functional decreases slowly, $f^n > f_{\text{trial}}^n > 0.9f^n$,
then τ_n is changed depending on signs of $\frac{dL}{dD_n}$ at the current and the previous optimization cycles.
- 6) if signs are opposite, i.e., $\left(\frac{dL}{dD_n} \right)^{\text{current}} \left(\frac{dL}{dD_n} \right)^{\text{previous}} < 0$, then τ_n is decreased, $\tau_n = 0.5\tau_n$,
- 7) otherwise, the signs are the same and τ_n is increased, $\tau_n = 1.5\tau_n$.

Here, f_{trial}^n and f^n are the trial and current values of the objective functional at the time level n , and dL/dD_n is the sensitivity derivative with respect to the control variable D_n .

V. Forming Discrete Adjoint Operators by Using Complex Variables

As follows from Eqs. (11-13), the derivatives of \mathbf{R} and f with respect to \mathbf{Q} and \mathbf{D} are required to form the adjoint equations and the sensitivity derivative. It is very difficult and time-consuming to obtain these derivatives by manually differentiating a CFD solver, especially if complicated turbulence and physical models are involved. Furthermore, any changes in the discretization of the governing equations, boundary conditions, objective functional, or physical models require additional coding and debugging, thus making the software development cycle extremely lengthy. To overcome these difficulties, we use an approach based on complex variables, which has successfully been applied to solving design optimization problems in [12, 13]. The key idea of this technique is to approximate the required real-valued derivatives by using the following formula proposed by Lyness¹⁴:

$$\frac{\partial f}{\partial x} = \frac{\text{Im}[f(x + ih)]}{h} + O(h^2), \quad (16)$$

where $f(x)$ is a complex-valued function. In contrast to the finite-difference method, the above complex variable formula is robust for small h , while providing true second order accuracy. Another advantage of this approach is that no additional flow solves are required to evaluate this derivatives, because only the solution at the current time level \mathbf{Q}^n is needed to compute \mathbf{R} and f and their perturbed values. The complex variable approach drastically reduces the implementation cycle and provides adjoint-based optimization capabilities for realistic physical and turbulence models. Note, however, that this approach is not without penalties in the CPU time and memory as compared with the handcoded Jacobians implementation because complex arithmetic is used.

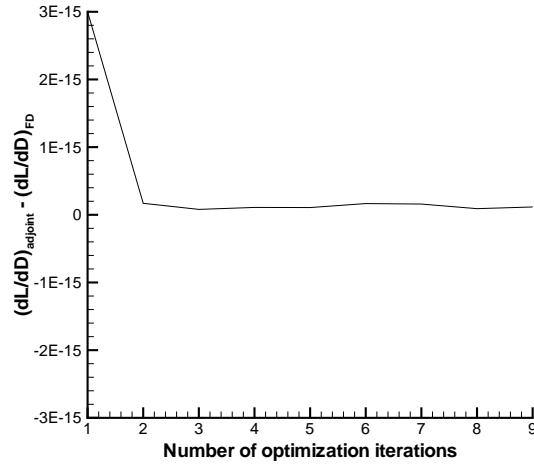


Figure 1. Comparison of the sensitivity derivatives computed using the finite difference and adjoint methods for the first test problem.

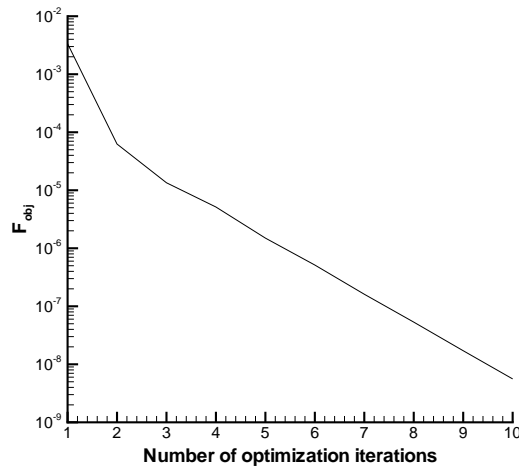


Figure 2. Convergence history of the objective functional F_{obj} .

VI. Numerical Results

In this section, we present computational results demonstrating how the adjoint-based method performs for two time-dependent optimization problems, involving flow matching functionals. The key distinction between these two problems is that for the first one, a design/control variable is independent of time, while for the second problem, control variables depend on time.

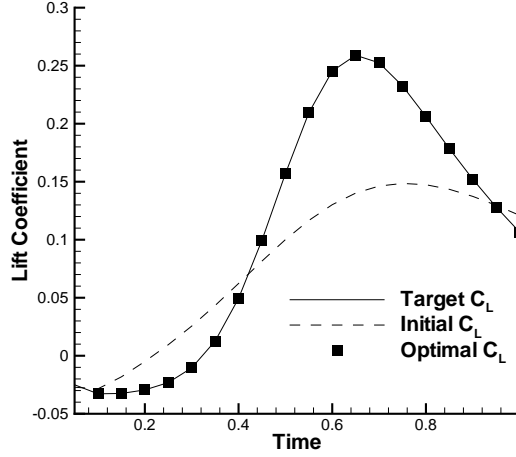


Figure 3. Comparison of the optimal lift coefficient computed using the adjoint-based method with its initial and target values for the first test problem.

The first problem is a minimization of a matching functional given by Eq. (5) with $\beta_2 = 0$ for the unsteady flow around a bump. The unsteadiness is introduced into the flow through the freestream Mach number which oscillates in time

$$M(t) = M_0 + \Delta M \cos(\omega t), \quad (17)$$

where M_0 is a mean value of the freestream Mach number, ΔM is a Mach number amplitude, and ω is a frequency of Mach number oscillations. The thickness of the bump is set to be 10% of its chord length. The flow conditions used in this test problem are: $M_0 = 2.0$, $\omega = 17\pi/9$, and $T_{\text{final}} = 1$. For this test problem, the Mach number amplitude ΔM is used as a control variable. Note that this control variable is independent of time. The time-dependent target lift coefficient $C_{L_{\text{target}}}(t)$ in the objective functional is calculated numerically by solving the unsteady Euler equations (2) with $\Delta M = 0.5$. A solution obtained at $\Delta M = 0.1$ is used as a starting point for the optimization procedure. The optimization is stopped when the absolute value of the difference between the current value of the Lagrangian and its value at the previous optimization iteration is less than 10^{-8} . This test problem is solved on a 61×21 structured grid using a node-centered finite volume code that is first-order accurate both in time and space. At each time step, the nonlinear discrete flow equations are solved by using the Newton's method. The adjoint equations are integrated backward in time and require the solution of the Euler equations to be known at all time steps, over which the optimization problem is solved. In the present implementation, the entire unsteady solution set is held in operating memory. For mid-size 2D problems integrated over $O(10^2)$ time steps, the required solution history, can be stored on a hard drive, as has been reported in [9]. In this case, the speed of I/O operation itself does not have a significant effect on the overall CPU time. Note, however, that for realistic 3D nonperiodic problems, this approach can quickly become prohibitive in terms of the disk memory and the CPU time, so more efficient approaches are needed to solve this class of unsteady optimization problems.

To evaluate the accuracy of calculation of $dL/d\mathbf{D}$ and to check the implementation of the adjoint solver, two different methods for computing the sensitivity derivative are used and compared with each other. The first method is based on a forward mode differentiation of the Lagrangian with respect to the control variable, which is implemented by using the complex variable approach (16). The second method uses the discrete adjoint formulation described in Section III. Note that at each optimization cycle, for the first method, the flow problem should be solved twice for each control variable. In contrast to the first method, the second approach requires a single solution of the Euler and corresponding adjoint equations per optimization cycle, regardless of the number of the control/design variables. For the forward mode differentiation, the complex step size has been chosen to be 10^{-7} . Figure 1 shows the difference between the sensitivity derivatives obtained using the finite difference and adjoint methods. As seen in the figure, the discrepancy is of the order of the round-off error, thus corroborating the validity of the time-dependent adjoint formulation. The history of convergence of the objective functional is presented in Fig. 2. The value of the objective functional drops by an order of magnitude every 2 optimization cycles. Note that this convergence behavior remains

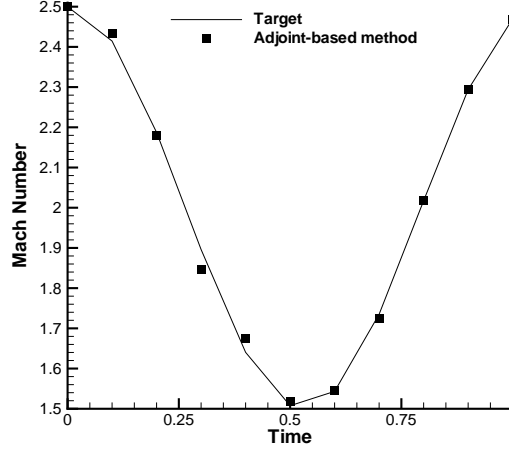


Figure 4. Comparison of the target Mach number and the optimal Mach number computed using the adjoint-based method for the second test problem.

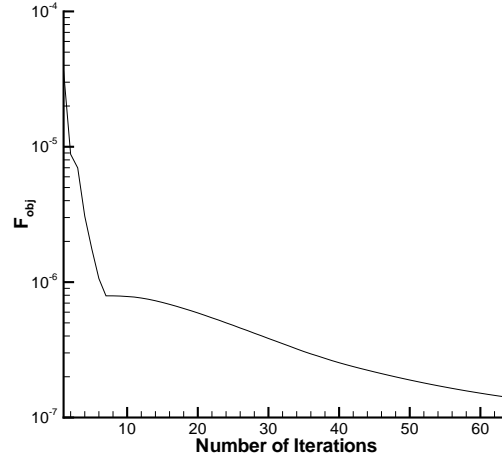


Figure 5. A history of convergence of the adjoint-based optimization method for the second test problem.

practically unchanged until the functional becomes smaller than the specified tolerance when the optimization was stopped. Only 10 optimization iterations were needed to reduce the objective functional by six orders of magnitude. To illustrate that the lift coefficient converges to its target value, time histories of the optimal, target, and initial lift coefficients are depicted in Figure 3. At the first optimization iteration, the maximum value of the time-dependent lift coefficient is about two times less than that of the target $C_L(t)$. After 10 optimization iterations, the time history of the computed lift coefficient is practically indistinguishable from the target solution at all time steps.

The second test problem is similar to the first one, but now, values of the freestream Mach number at each time step $\mathbf{D} = (M_1, \dots, M_N)^T$ are used as control variables. The optimization procedure starts at $M = 2.1$ which is used as an initial guess. The objective functional for this test problem is given by

$$F_{obj} = \sum_{\Gamma_c} (P_j^n - P_{\text{target}}^n)^2 ,$$

where P_j^n and P_{target}^n are computed and target time-dependent pressure profiles at the lower boundary of the computational domain. The target pressure distribution is calculated numerically by solving the same unsteady problem with $M = 2 + 0.5 \cos(17\pi t/9)$. The optimization is stopped when either the relative

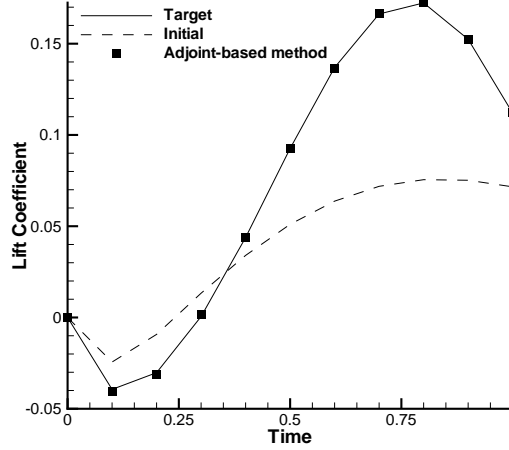


Figure 6. Comparison of the optimal lift coefficient computed using the adjoint-based method with its initial and target values for the second test problem.

change in the value of each control variable becomes smaller than 10^{-4} or the absolute value of the objective functional becomes smaller than 10^{-7} . As in the previous case, the adjoint equations are integrated backward in time and require the solution of the Euler equations to be known at all time steps, which is held in the operating memory. Figure 4 shows the optimal and target Mach number distributions in time. As seen in the figure, the time-dependent optimization method converges to the target solution on the entire time interval considered, thus validating the unsteady adjoint formulation.

A history of convergence of the objective functional obtained with adjoint-based optimization technique is presented in Fig. 5. The total number of optimization cycles required for the adjoint-based optimization method to converge is an order of magnitude larger than that obtained for the first test problem. This is not surprising, because the dimensionality of the design space has also increased by an order of magnitude.

To illustrate that the lift coefficient converges to its target value, the optimal C_L obtained with time-dependent optimization method as well as the initial and target lift coefficients are presented in Figure 6. The relative difference between the initial lift coefficient and its target value is of the order of $O(1)$, while the solution obtained with the adjoint-based method is almost indistinguishable from the target C_L over the entire time interval considered.

VII. Conclusions

We have developed the general adjoint-based methodology for solving a broad spectrum of optimal flow control and design optimization problems. The methodology is directly applicable to both time-dependent optimization problems with control/design variables that are time-dependent and design optimization problems with the control variables that do not depend on time. Nonlinear constraints on the control/design and state variables can be incorporated into the present formulation by introducing the penalty/regularization term in the cost functional. The discrete adjoint operators required for this formulation are computed by using the complex variable approach which is robust for very small step sizes, thus providing adjoint-based optimization capabilities for realistic physical models. The present adjoint-based methodology has been validated using two test problems involving flow matching functionals. Applications of this adjoint-based methodology to more realistic time-dependent design optimization problems involving moving and deforming grids is currently under investigation. Our future research will also focus on developing optimization and computational techniques for reducing the CPU and memory cost of the present time-dependent adjoint-based methodology.

References

- ¹Borggaard, J., Burkardt, J., Gunzburger, M., and Peterson, J., Optimal design and control, Boston: Birkhauser, 1995.

- ²Jameson, A., Pierce, N., Martinelli, L., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theor. Comput. Fluid Dyn.* Vol. 10, No. 1, pp. 213–237, 1998.
- ³Anderson, W.K., Bonhaus, D.L., "Airfoil Design on Unstructured Grids for Turbulent Flows," *AIAA J.* Vol. 37, No. 2, pp. 185–191, 1998.
- ⁴Nielsen, E.J., and Anderson, W.K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes," *AIAA J.* Vol. 40, No. 6, pp. 1155–1163, 2002.
- ⁵Soto, R., and Yang, C., "An Adjoint-based Methodology for CFD Optimization Problems," *AIAA Paper* 2003-0299, 2003.
- ⁶Joslin, R.D., Gunzburger, M.D., Nicolaides, R.A., Erlebacher, G., and Hussaini, M.Y., "Self-Contained Automated Methodology for Optimal Control," *AIAA J.* Vol. 35, No. 5, pp. 816–824, 1997.
- ⁷Nadarajah, S.K., Jameson, A., "Optimal Control of Unsteady Flows Using a Time Accurate Method," *AIAA Paper* 2002-5436, 2002.
- ⁸Nadarajah, S.K., McMullen, M., and Jameson, A., "Non-Linear Frequency Domain Based Optimum Shape Design for Unsteady Three-Dimensional Flow," *AIAA Paper* 2006-1052, 2006.
- ⁹Mani, K., Mavriplis, D. J., "An Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Paper* 2007-60, 2007.
- ¹⁰Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., Ulbrich, S., "Optimal Control of Unsteady Compressible Viscous Flows," *Int. J. Numer. Meth. Fluids*, Vol.40, No. 11, 2002, pp. 1401-1429.
- ¹¹Rumpfkeil, M.P., and Zingg, D.W., "A General Framework for the Optimal Control of Unsteady Flows with Applications," *AIAA paper*, 2007-1128.
- ¹²Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen E. J., "Sensitivity Analysis for the Navier-Stokes Equations on Unstructured Meshes Using Complex Variables, *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63.
- ¹³Nielsen, E. J., and Kleb, W. L., "Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables, *AIAA Journal*, Vol. 44, No. 4, 2005, pp. 827–836.
- ¹⁴Lyness, J. N., "Numerical Algorithms Based on the Theory of Complex Variables," *Proc. ACM 22nd Nat. Conf.*, Thomas Book Co., Washington, D.C., 1967, pp. 124–134.