INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2012; **68**:564–581
Published online 31 January 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.2521

One-shot pseudo-time method for aerodynamic shape optimization using the Navier–Stokes equations

S. B. Hazra[1,*,†,‡] and A. Jameson[2]

[1]*Department of Mechanical Engineering, Technical University of Darmstadt, D-64287 Darmstadt, Germany*
[2]*Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, U.S.A.*

SUMMARY

This paper presents a numerical method for aerodynamic shape optimization problems in compressible viscous flow. It is based on simultaneous pseudo-time stepping in which stationary states are obtained by solving the pseudo-stationary system of equations representing the state, costate and design equations. The main advantages of this method are that it blends in nicely with previously existing pseudo-time-stepping methods for the state and the costate equations, that it requires no additional globalization in the design space, and that a preconditioner can be used for convergence acceleration which stems from the reduced SQP methods. For design examples of 2D problems, the overall cost of computation can be reduced to less than 2 times the forward simulation runs. Copyright © 2011 John Wiley & Sons, Ltd.

Received 14 December 2009; Revised 10 September 2010; Accepted 14 November 2010

KEY WORDS: aerodynamic shape optimization; simultaneous pseudo-time stepping; Navier–Stokes equations; adjoint equations; preconditioner; one-shot method; airfoil

1. INTRODUCTION

Automatic aerodynamic shape optimization using numerical methods is an established area of scientific research. The numerical methods involve the complexity of the numerical algorithms for the optimization problem as well as the complexity of the numerical methods for non-linear partial differential equations (PDEs). Parallel research has been going on to improve the efficiency and applicability of the algorithms in both the areas. Recently, both the communities are working together to achieve the best results in applications to practical problems.

Aerodynamic shape optimization problems can be mathematically formulated as control problems governed by a system of PDEs. Pioneering theoretical works on the methodology for solving such problems have been presented in [1–4]. For these problems an adjoint system of PDEs can be formulated and solved using the same algorithm as applicable to the state system. The gradient is computed very efficiently using the state and adjoint solutions in comparison to finite-difference methods, which require a number of state solutions proportional to the number of design variables to evaluate the gradient. Application of this continuous adjoint method was carried out first in [5–8] for transonic flow using Euler equations. Traditional gradient methods act only in the design space and require very accurate flow (or state) and adjoint (or costate) solutions. Despite using efficient computational fluid dynamics (CFD) techniques for the state and costate solutions, the overall cost of computation is quite high in these methods. Jameson has proposed gradient smoothing as a

*Correspondence to: S. B. Hazra, Institute IWAR, Department of Civil Engineering and Geodesy, Technical University of Darmstadt, D-64287 Darmstadt, Germany.
†E-mail: hazra@fdy.tu-darmstadt.de
‡Institute of Fluid Dynamics.

remedy for this difficulty. This is equivalent to defining the gradient with respect to a Sobolev innerproduct [9]. Instead of using the direct gradient information from the adjoint solution, the gradient is smoothed implicitly via second-order (or fourth-order) differential equations and the smoothed gradient is used to define the search direction. It turns out that this approach is tolerant to the use of inexact gradients, so that neither the flow solution nor the adjoint solution needs to be fully converged, leading to a drastic reduction of the computational cost. The extension of the continuous adjoint method for shape optimization problems in viscous compressible flow is carried out in [9–11]. Use of less accurate state and adjoint solutions for Euler equations has also been performed in Iollo *et al.* [12].

In [13], Ta'asan proposed an approach in which pseudo-time embedding is suggested for the state and costate equations and the design equation is solved as an additional boundary condition, especially for boundary control problems. The resulting system of equations in his formulation is still a system of differential algebraic equations, where one still has to provide some separate means to solve the design equation. In [14], we superseded that formulation by constructing a system containing entirely of ordinary differential equations (ODEs) and proposed a new method for solving such optimization problems using simultaneous pseudo-time stepping. In [14–18], we have applied the method for solving aerodynamic shape optimization problems without additional state constraints and in [19–21] applied the method to problems with additional state constraints. A simultaneous iterative approach has been performed in [22] but the cost of that method is proportional to the number of design variables. In [23, 24], another simultaneous or one-shot method has been used where design variables are updated in a hierarchical manner. The overall cost of computation using simultaneous pseudo-time stepping in all the applications has been between 2 and 8 times that of the forward simulation runs. Further efficiency is achieved using the 'optimization-based' multigrid method in [25].

All the above-mentioned applications of the pseudo-time-stepping method have been in inviscid compressible flow. In this paper, we extend the method to viscous compressible flow modeled by Reynolds averaged Navier–Stokes equations together with the algebraic turbulence model of Baldwin and Lomax [26]. While inviscid formulations are useful for the design in transonic cruise conditions, inclusion of viscous effects is essential for optimal design encompassing off-design conditions and high-lift configurations. The computational complexity of viscous design is at least one order of magnitude greater than that of inviscid design since the number of mesh points must be increased by a factor of two or more to resolve the boundary layer. The convergence of the Navier–Stokes solvers is much slower than the Euler solvers due to discrete stiffness and directional decoupling arising from the highly stretched boundary layer cells. In the inviscid case, the one-shot pseudo-time-stepping method requires a single iteration of the flow and the adjoint solvers at each optimization step. The slower convergence of the Navier–Stokes (flow and adjoint) solvers may lead to a requirement of more than 1 iteration at each optimization step. We investigate that numerically in this paper.

The paper is organized as follows. In the following section we discuss the abstract formulation of the shape optimization problem and its reduction to the preconditioned pseudo-stationary system of PDEs. Section 3 presents the state, costate and design equations. Numerical results are presented in Section 4. We draw our conclusions in Section 5.

2. THE OPTIMIZATION PROBLEM AND PSEUDO-UNSTEADY FORMULATION OF THE KARUSH–KUHN–TUCKER (KKT) CONDITIONS

The focus of this work is on aerodynamic shape optimization problems that are large-scale PDE-constrained optimization problems. These problems can be written in abstract form (see in [14, 18]) as

$$\begin{aligned} &\text{minimize} \quad I(w,q), \\ &\text{subject to} \quad c(w,q)=0. \end{aligned} \tag{1}$$

DOI: 10.1002/fld

Here, $c(w,q)=0$ represents the steady-state flow equations (in our case the Navier–Stokes equations) together with the boundary conditions, w is the vector of dependent variables and q is the vector of design variables. The objective $I(w,q)$ is the drag of an airfoil for the purpose of this paper.

The necessary optimality conditions (known as KKT conditions) for the optimization problem (1) are

$$c(w,q)=0 \text{ (State equation)}, \tag{2a}$$

$$\nabla_w L(w,q,\psi)=0 \text{ (Costate equation)}, \tag{2b}$$

$$\nabla_q L(w,q,\psi)=0 \text{ (Design equation)}, \tag{2c}$$

where

$$L(w,q,\psi)=I(w,q)-\psi^\top c(w,q) \tag{3}$$

is the Lagrangian functional and ψ is the vector of Lagrange multipliers or the adjoint variables. Adjoint-based gradient methods have been used in many practical applications for solving the above system of equations. In these methods the state and costate equations have to be solved quite accurately in each design update. Computational results based on these methodologies have been presented, among others, in [6, 7, 27, 28] on structured grids. An application of this method on unstructured grids have been presented in [29–31].

One can use, for example, the reduced sequential quadratic programming (rSQP) method to solve the above set of equations (2). A step of this method can also be interpreted as an approximate Newton step for the necessary conditions of finding the extremum of problem (1), since the updates of the variables are computed according to the linear system

$$\begin{pmatrix} 0 & 0 & A^\top \\ 0 & B & \left(\dfrac{\partial c}{\partial q}\right)^\top \\ A & \dfrac{\partial c}{\partial q} & 0 \end{pmatrix} \begin{pmatrix} \Delta w \\ \Delta q \\ \Delta \psi \end{pmatrix} = \begin{pmatrix} -\nabla_w L \\ -\nabla_q L \\ -c \end{pmatrix}, \tag{4}$$

where A is the approximation to the Jacobian $\partial c/\partial w$ and B is the reduced Hessian.

We used in [14, 16, 18] a new method for solving the above problem (2) using simultaneous pseudo-time stepping. In this method, to determine the solution of (2), we look for the steady-state solutions of the following pseudo-time embedded evolution equations:

$$\begin{aligned} \frac{\mathrm{d}w}{\mathrm{d}t}+c(w,q)&=0,\\ \frac{\mathrm{d}\psi}{\mathrm{d}t}+\nabla_w L(w,q,\psi)&=0,\\ \frac{\mathrm{d}q}{\mathrm{d}t}+\nabla_q L(w,q,\psi)&=0. \end{aligned} \tag{5}$$

This formulation is advantageous since the steady-state flow (and adjoint) solution is obtained by integrating the pseudo-unsteady Navier–Stokes (and adjoint Navier–Stokes) equations in this problem class. Therefore, one can use the same time-stepping philosophy for the whole set of equations and preconditioners can be used to accelerate the convergence. The preconditioner that we have used stems from rSQP methods as discussed above and in detail in [14].

The pseudo-time embedded system (5) usually results (after semi-discretization) in a stiff system of ODEs. Therefore, explicit time-stepping schemes may converge very slowly or might even diverge. In order to accelerate convergence, this system needs some preconditioning. We use

DOI: 10.1002/fld

the inverse of the matrix in Equation (4) as a preconditioner for the time-stepping process. The pseudo-time embedded system that we consider reads as

$$\begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 0 & 0 & A^\top \\ 0 & B & \left(\dfrac{\partial c}{\partial q}\right)^\top \\ A & \dfrac{\partial c}{\partial q} & 0 \end{bmatrix}^{-1} \begin{pmatrix} -\nabla_w L \\ -\nabla_q L \\ -c \end{pmatrix}. \tag{6}$$

This seems natural since Equation (4) can be considered as an explicit Euler discretization for the corresponding time stepping that we envision. Also, due to its block structure, the preconditioner is computationally inexpensive. The preconditioner employed is similar to the preconditioners for KKT-systems discussed in [32, 33] in the context of Krylov subspace methods and in [34] in the context of Lagrange–Newton–Krylov–Schur methods.

Within the inexact rSQP-preconditioner, one has to look for an appropriate approximation of the inverse of the reduced Hessian. As shown in [15], the reduced Hessian update based on the most recent reduced gradient and parameter update information is good enough, so we use the same update strategy in this paper. We define $s_k := (q_{k+1} - q_k)$ and $z_k := (\nabla g_{k+1} - \nabla g_k)$, where k represents the iteration number and g is the reduced gradient. Then, the reduced Hessian update is based on the sign of the product $(z_k^\top s_k)$. If the sign is positive, the reduced Hessian is approximated by

$$B_k = \bar{\beta} \Gamma_k \delta_{ij} \quad \text{with } \Gamma_k = \frac{z_k^\top s_k}{z_k^\top z_k},$$

where $\bar{\beta}$ is a constant. Otherwise, it is approximated by $\beta \delta_{ij}$, where β is another constant. Additionally, we impose upper and lower limits on the factor so that

$$\beta_{\min} < \bar{\beta} \frac{z_k^\top s_k}{z_k^\top z_k} < \beta_{\max}.$$

This prevents the optimizer from taking steps that are too small or too large. The constants can be chosen, e.g. depending on the accuracy achieved in one time step by the forward and adjoint solver.

2.1. The optimization problem with additional state constraint

Practical aerodynamic shape optimization problems involve additional state constraints such as minimization of the drag for a given lift. In addition, it is generally necessary to maintain sufficient thickness to meet structural requirements and also provide enough fuel volume. Problems with additional state constraints can be written in mathematical form as

$$\begin{aligned} \text{minimize} \quad & I(w, q, \alpha), \\ \text{subject to} \quad & c(w, q, \alpha) = 0, \\ & h(w, q, \alpha) = h_0, \end{aligned} \tag{7}$$

where α is the angle of incidence. We distinguish α from the geometric design parameter q since it is changed between the optimization iterations to achieve the constraint of constant lift. The lift constraint is typically of the form

$$h(w, q, \alpha) \geqslant h_0.$$

DOI: 10.1002/fld

Since we know *a priori* that this constraint is active, we treat it as equality constraint. As discussed in detail in [21], this problem can be reformulated as

$$\begin{aligned} &\underset{(w,q)}{\text{minimize}} \quad I(w,q,\alpha)-\nu(h(w,q,\alpha)-h_0), \\ &\text{subject to} \quad c(w,q,\alpha)=0 \end{aligned} \tag{8}$$

for (ν,α) given. The objective function incorporates the lift constraint in a 'Lagrangian' way to avoid the drag reduction by reducing the lift. Note that (8) looks like a penalty representation, but it differs from that insofar as ν (which is the Lagrange multiplier corresponding to the lift constraint) does not tend toward ∞. The necessary optimality conditions can be formulated following the discussion above which are then solved using the simultaneous pseudo-time-stepping method.

3. DETAILED EQUATIONS OF THE AERODYNAMIC SHAPE OPTIMIZATION PROBLEM

In this section, we explain briefly the state, costate and design equations represented in Equation (2) for the shape optimization problem. The formulation of the adjoint and gradient equations for viscous optimization follows the development in References [10, 11].

3.1. State equations

Since we are interested in the steady flow, a proper approach for numerical modeling is to integrate the unsteady Navier–Stokes equations in time until a steady state is reached. These equations in Cartesian coordinates (x_1,x_2) for two-dimensional flow can be written in differential form for the region $\mathscr{D}$ with boundaries $\mathscr{B}$ as

$$\frac{\partial w}{\partial t}+\frac{\partial f_i}{\partial x_i}=\frac{\partial f_{vi}}{\partial x_i} \quad (i=1,2) \quad \text{in } \mathscr{D}, \tag{9}$$

where

$$w:=\begin{bmatrix}\rho\\ \rho u_1\\ \rho u_2\\ \rho E\end{bmatrix}, \quad f_i:=\begin{bmatrix}\rho u_i\\ \rho u_i u_1+p\delta_{i1}\\ \rho u_i u_2+p\delta_{i2}\\ \rho u_i H\end{bmatrix} \quad \text{and} \quad f_{vi}:=\begin{bmatrix}0\\ \sigma_{ij}\delta_{j1}\\ \sigma_{ij}\delta_{j2}\\ u_j\sigma_{ij}+k\dfrac{\partial T}{\partial x_i}\end{bmatrix}.$$

For a perfect gas the pressure and the total enthalpy are given by

$$p=(\gamma-1)\rho\left\{E-\frac{1}{2}(u^2+v^2)\right\}, \quad H=E+\frac{p}{\rho},$$

respectively. The viscous stresses may be written as

$$\sigma_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial u_k}{\partial x_k},$$

where μ and λ are the first and second coefficients of viscosity. The coefficient of thermal conductivity and the temperature are computed as

$$k=\frac{c_p\mu}{Pr}, \quad T=\frac{p}{R\rho},$$

where Pr is the Prandtl number, c_p is the specific heat at constant pressure and R is the universal gas constant.

DOI: 10.1002/fld

For discussion of real applications using a discretization on a body-conforming structured mesh, it is useful to consider a transformation to the computational coordinates (ξ_1, ξ_2) defined by the metrics

$$K_{ij} = \left[\frac{\partial x_i}{\partial \xi_j}\right], \quad J = \det(K), \quad K_{ij}^{-1} = \left[\frac{\partial \xi_i}{\partial x_j}\right].$$

The Navier–Stokes equations can then be written in computational space as

$$\frac{\partial (Jw)}{\partial t} + \frac{\partial (F_i - F_{vi})}{\partial \xi_i} = 0 \quad \text{in } \mathscr{D}, \tag{10}$$

where the inviscid and viscous flux contributions are now defined with respect to the computational cell faces by $F_i = S_{ij} f_j$ and $F_{vi} = S_{ij} f_{vj}$, and the quantity $S_{ij} = J K_{ij}^{-1}$ represents the projection of the ξ_i cell face along the x_j axis. In obtaining Equation (10) we have made use of the property that

$$\frac{\partial S_{ij}}{\partial \xi_i} = 0,$$

which represents the fact that the sum of the face areas over a closed volume is zero, as can be readily verified by a direct examination of the metric terms.

The boundary conditions used to solve these equations are the 'no slip' condition on the solid wall, and the farfield boundary is treated by considering the incoming and outgoing characteristics based on the one-dimensional Riemann invariants, as described in [35, p. 194], [36].

3.2. Costate equations

Aerodynamic optimization is based on the determination of the effect of shape modifications on some performance measure which depends on the flow. For convenience, the coordinates ξ_i describing the fixed computational domain are chosen so that each boundary conforms to a constant value of one of these coordinates. Variations in the shape then result in corresponding variations in the mapping derivatives defined by K_{ij}.

3.2.1. Formulation of continuous adjoint equations. Suppose that the performance is measured by a cost function

$$I = \int_{\mathscr{B}} \mathscr{M}(w, S)\, \mathrm{d}B_\xi + \int_{\mathscr{D}} \mathscr{P}(w, S)\, \mathrm{d}D_\xi \tag{11}$$

containing both boundary and field contributions where $\mathrm{d}B_\xi$ and $\mathrm{d}D_\xi$ are the surface and volume elements in the computational domain. In general, $\mathscr{M}$ and $\mathscr{P}$ will depend on both the flow variables w and the metrics S defining the computational space. The design problem is now treated as a control problem where the boundary shape represents the control function, which is chosen to minimize I subject to the constraints defined by the flow equations (10). A shape change produces a variation in the flow solution δw and the metrics δS which in turn produce a variation in the cost function

$$\delta I = \int_{\mathscr{B}} \delta \mathscr{M}(w, S)\, \mathrm{d}B_\xi + \int_{\mathscr{D}} \delta \mathscr{P}(w, S)\, \mathrm{d}D_\xi. \tag{12}$$

This can be split into

$$\delta I = \delta I_I + \delta I_{II} \tag{13}$$

with

$$\begin{aligned} \delta \mathscr{M} &= [\mathscr{M}_w]_I \delta w + \delta \mathscr{M}_{II}, \\ \delta \mathscr{P} &= [\mathscr{P}_w]_I \delta w + \delta \mathscr{P}_{II}, \end{aligned} \tag{14}$$

DOI: 10.1002/fld

where we continue to use the subscripts I and II to distinguish between the contributions associated with the variation of the flow solution δw and those associated with the metric variations δS. Thus, $[\mathscr{M}_w]_I$ and $[\mathscr{P}_w]_I$ represent $\partial\mathscr{M}/\partial w$ and $\partial\mathscr{P}/\partial w$ with the metrics fixed, while $\delta\mathscr{M}_{II}$ and $\delta\mathscr{P}_{II}$ represent the contribution of the metric variations δS to $\delta\mathscr{M}$ and $\delta\mathscr{P}$.

In the steady state, the constraint equation (10), denoting $R(w)=0$, specifies the variation of the state vector δw by

$$\delta R=\frac{\partial}{\partial\xi_i}\delta(F_i-F_{vi})=0. \tag{15}$$

Here also δR, δF_i and δF_{vi} can be split into contributions associated with δw and δS using the notations

$$\begin{aligned}\delta R &= \delta R_I+\delta R_{II},\\ \delta F_i &= [F_{i\,w}]_I\delta w+\delta F_{i\,II},\\ \delta F_{vi} &= [F_{vi\,w}]_I\delta w+\delta F_{vi\,II}.\end{aligned} \tag{16}$$

The inviscid contributions are easily evaluated as

$$[F_{i\,w}]_I=S_{ij}\frac{\partial f_i}{\partial w},\quad \delta F_{i\,II}=\delta S_{ij}f_j.$$

The details of the viscous contributions are complicated by the additional level of derivatives in the stress and heat flux terms.

Multiplying by a costate vector ψ and integrating over the domain produces

$$\int_{\mathscr{D}}\psi^\top\frac{\partial}{\partial\xi_i}\delta(F_i-F_{vi})\,\mathrm{d}\mathscr{D}_\xi=0. \tag{17}$$

Assuming that ψ is differentiable, the terms with subscript I may be integrated by parts to give

$$\int_{\mathscr{B}}n_i\psi^\top\delta(F_i-F_{vi})_I\,\mathrm{d}\mathscr{B}_\xi-\int_{\mathscr{D}}\frac{\partial\psi^\top}{\partial\xi_i}\delta(F_i-F_{vi})_I\,\mathrm{d}\mathscr{D}_\xi+\int_{\mathscr{D}}\psi^\top\delta R_{II}\,\mathrm{d}\mathscr{D}_\xi=0. \tag{18}$$

This equation results directly from taking the variation of the weak form of the flow equations, where ψ is taken to be an arbitrary differentiable test function with compact support. Since the left-hand expression equals zero, it may be subtracted from the variation of the cost function (12) to give

$$\begin{aligned}\delta I=\delta I_{II}&-\int_{\mathscr{D}}\psi^\top\delta R_{II}\mathrm{d}\mathscr{D}_\xi-\int_{\mathscr{B}}[\delta\mathscr{M}_I-n_i\psi^\top\delta(F_i-F_{vi})_I]\,\mathrm{d}\mathscr{B}_\xi\\ &+\int_{\mathscr{D}}\left[\delta\mathscr{P}_I+\frac{\partial\psi^\top}{\partial\xi_i}\delta(F_i-F_{vi})_I\right]\mathrm{d}\mathscr{D}_\xi.\end{aligned} \tag{19}$$

Now, since ψ is an arbitrary differentiable function, it may be chosen in such a way that δI no longer depends explicitly on the variation of the state vector δw. The gradient of the cost function can then be evaluated directly from the metric variations without having to recompute the variation δw resulting from the perturbation of each design variable.

Comparing Equations (14) and (16), the variation δw may be eliminated from (19) by equating all field terms with subscript 'I' to produce a differential adjoint system governing ψ

$$\frac{\partial\psi^\top}{\partial\xi_i}[F_{i\,w}-F_{vi\,w}]_I+[\mathscr{P}_w]_I=0\quad\text{in }\mathscr{D}. \tag{20}$$

DOI: 10.1002/fld

The corresponding adjoint boundary condition is produced by equating the subscript 'I' boundary terms in Equation (19) as

$$n_i \psi^\top [F_{i\,w} - F_{vi\,w}]_I = [\mathscr{M}_w]_I \quad \text{on } \mathscr{B}. \tag{21}$$

The remaining terms in Equation (19) then yield a simplified expression for the variation of the cost function which defines the gradient

$$\begin{aligned} \delta I &= \delta I_{II} + \int_{\mathscr{D}} \psi^\top \delta R_{II}\, \mathrm{d}\mathscr{D}_\xi \\ &= \int_{\mathscr{B}} \{\delta \mathscr{M}_{II} - n_i \psi^\top [\delta F_i - \delta F_{vi}]_{II}\}\, \mathrm{d}\mathscr{B}_\xi + \int_{\mathscr{D}} \left\{ \delta \mathscr{P}_{II} + \frac{\partial \psi^\top}{\partial \xi_i} [\delta F_i - \delta F_{vi}]_{II} \right\} \mathrm{d}\mathscr{D}_\xi \end{aligned} \tag{22}$$

that consists purely of the terms containing variations in the metrics with the flow solution fixed. Hence, an explicit formula for the gradient can be derived once the relationship between mesh perturbations and shape variations is defined.

The details of the formula for the gradient depend on the way in which the boundary shape is parameterized as a function of the design variables, and the way in which the mesh is deformed as the boundary is modified. Using the relationship between the mesh deformation and the surface modification, the field integral is reduced to a surface integral by integrating along the coordinate lines emanating from the surface. Thus, the expression for δI is finally reduced to the form

$$\delta I = \int_{\mathscr{B}} \mathscr{G} \delta q\, \mathrm{d}\mathscr{B}_\xi,$$

where q represents the design variables, and $\mathscr{G}$ is the gradient, which is a function defined over the boundary surface.

The boundary conditions satisfied by the flow equations restrict the form of the left-hand side of the adjoint boundary condition (21). Consequently, the boundary contribution to the cost function $\mathscr{M}$ cannot be specified arbitrarily. Instead, it must be chosen from the class of functions that allow the cancellation of all terms containing δw in the boundary integral of Equation (19). On the other hand, there is no such restriction on the specification of the field contribution to the cost function $\mathscr{P}$, since these terms may always be absorbed into the adjoint field equation (20) as source terms.

For simplicity, it is assumed that the portion of the boundary that undergoes shape modifications is restricted to the coordinate surface $\xi_2 = 0$. Then, Equations (19) and (21) may be simplified by incorporating the conditions

$$n_1 = 0, \quad n_2 = 1, \quad \mathrm{d}\mathscr{B}_\xi = \mathrm{d}\xi_1$$

so that only the variations δF_2 and δF_{v2} need to be considered at the wall boundary.

3.2.2. Derivation of the continuous inviscid and viscous adjoint terms. It is convenient to develop the inviscid and viscous contributions to the adjoint equations separately. The inviscid contributions are derived in [6, 8] and will not be discussed here. The viscous contributions are derived in [9–11] and can be found in [37] as well for a 2D case.

In these papers, the viscous terms are derived under the assumption that the viscosity and heat conduction coefficients μ and k are essentially independent of the flow, and that their variations may be neglected. This simplification has been successfully used for many aerodynamic problems of interest. In the case of some turbulent flows, there is the possibility that the flow variations could result in significant changes in the turbulent viscosity, and it may then be necessary to account for its variation in the calculation.

DOI: 10.1002/fld

3.2.3. Transformation to primitive variables. The derivation of the viscous adjoint terms is simplified by transforming into the primitive variables

$$\tilde{w}^\top = (\rho, u_1, u_2, p)^\top$$

because the viscous stresses depend on the velocity derivatives $\partial u_i / \partial x_j$, while the heat flux can be expressed as

$$\kappa \frac{\partial}{\partial x_i} \left(\frac{p}{\rho} \right),$$

where $\kappa = k/R = \gamma\mu/Pr(\gamma - 1)$. The relationship between the conservative and primitive variations is defined by the expressions

$$\delta w = M \delta \tilde{w}, \quad \delta \tilde{w} = M^{-1} \delta w,$$

which make use of the transformation matrices $M = \partial w / \partial \tilde{w}$ and $M^{-1} = \partial \tilde{w} / \partial w$. These matrices are provided in transposed form for future convenience

$$M^\top = \begin{bmatrix} 1 & u_1 & u_2 & \dfrac{u_i u_i}{2} \\ 0 & \rho & 0 & \rho u_1 \\ 0 & 0 & \rho & \rho u_2 \\ 0 & 0 & 0 & \dfrac{1}{\gamma - 1} \end{bmatrix}$$

$$M^{-1^\top} = \begin{bmatrix} 1 & -\dfrac{u_1}{\rho} & -\dfrac{u_2}{\rho} & \dfrac{(\gamma-1) u_i u_i}{2} \\ 0 & \dfrac{1}{\rho} & 0 & -(\gamma - 1) u_1 \\ 0 & 0 & \dfrac{1}{\rho} & -(\gamma - 1) u_2 \\ 0 & 0 & 0 & \gamma - 1 \end{bmatrix}.$$

The conservative and primitive adjoint operators L and $\tilde{L}$ corresponding to the variations δw and $\delta \tilde{w}$ are then related by

$$\int_{\mathscr{D}} \delta w^\top L \psi \, \mathrm{d}\mathscr{D}_\xi = \int_{\mathscr{D}} \delta \tilde{w}^\top \tilde{L} \psi \, \mathrm{d}\mathscr{D}_\xi$$

with

$$\tilde{L} = M^\top L$$

so that after determining the primitive adjoint operator by direct evaluation of the viscous portion of (20), the conservative operator may be obtained by the transformation $L = M^{-1^\top} \tilde{L}$. Since the continuity equation contains no viscous terms, it makes no contribution to the viscous adjoint system. The details of derivations for contributions from the momentum and the energy equations are given in the references mentioned above.

DOI: 10.1002/fld

3.2.4. The viscous adjoint field operator. Collecting together the contributions from the momentum and the energy equations, the viscous adjoint operator in primitive variables can finally be expressed as

$$(\bar{L}\psi)_1 = -\frac{p}{\rho^2}\frac{\partial}{\partial\xi_l}\left(S_{lj}\kappa\frac{\partial\theta}{\partial x_j}\right)$$

$$(\bar{L}\psi)_{i+1} = \frac{\partial}{\partial\xi_l}\left\{S_{lj}\left[\mu\left(\frac{\partial\phi_i}{\partial x_j}+\frac{\partial\phi_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial\phi_k}{\partial x_k}\right]\right\} + \frac{\partial}{\partial\xi_l}\left\{S_{lj}\left[\mu\left(u_i\frac{\partial\theta}{\partial x_j}+u_j\frac{\partial\theta}{\partial x_i}\right)+\lambda\delta_{ij}u_k\frac{\partial\theta}{\partial x_k}\right]\right\} - \sigma_{ij}S_{lj}\frac{\partial\theta}{\partial\xi_l} \quad \text{for} \quad i=1,2$$

$$(\bar{L}\psi)_4 = \frac{1}{\rho}\frac{\partial}{\partial\xi_l}\left(S_{lj}\kappa\frac{\partial\theta}{\partial x_j}\right).$$

3.2.5. The cost function and the adjoint boundary conditions. The cost function that we choose in the present optimization problem is drag reduction. Hence, the cost function, which corresponds to Equation (11), reads as

$$I(w,q) := C_D = \frac{1}{C_{\text{ref}}}\int_{\mathscr{B}} C_p\left(\frac{\partial y}{\partial\xi}\cos\alpha - \frac{\partial x}{\partial\xi}\sin\alpha\right)\mathrm{d}\xi_1, \tag{23}$$

where C_{ref} is the chord length of the airfoil and the surface pressure coefficient is defined by

$$C_p := \frac{2(p-p_\infty)}{\gamma M_\infty^2 p_\infty}. \tag{24}$$

The boundary conditions for the adjoint equations on the solid body, corresponding to Equation (21), for the above-mentioned cost function, are given by

$$n_{\xi_1}\psi_2 + n_{\xi_2}\psi_3 = -\frac{2}{\gamma M_\infty^2 p_\infty C_{\text{ref}}}(n_{\xi_1}\cos\alpha + n_{\xi_2}\sin\alpha) \quad \text{on } \mathscr{B}. \tag{25}$$

3.3. Design equation

For the design equation (2c), we need an expression for the derivative of the Lagrangian with respect to the geometry of the airfoil. However, in the actual computation instead of actual gradient (derivative of the Lagrangian) the reduced gradient (derivative of the cost function with respect to the geometry of the airfoil) is used. Hence for the above-mentioned cost function, the reduced gradient is given by

$$\delta I = \frac{1}{C_{\text{ref}}}\int_{\mathscr{B}} C_p\left(\delta\left(\frac{\partial y}{\partial\xi}\right)\cos\alpha - \delta\left(\frac{\partial x}{\partial\xi}\right)\sin\alpha\right)\mathrm{d}\xi_1. \tag{26}$$

3.4. Gradient smoothing

The reduced gradient obtained using inaccurate state and costate solutions are quite non-smooth, especially near the leading and trailing edges. In order to make sure that each new shape in the optimization sequence remains smooth, it proves essential to smooth the gradient and to replace $\mathscr{G}$ by its smoothed value $\bar{\mathscr{G}}$ in the descent process. This also acts as a preconditioner that allows the use of much larger steps. The gradient smoothing is equivalent to redefining the inner product in a Sobolev space as described in [9], and the steps in the smoothed gradient direction still guarantee

DOI: 10.1002/fld

descent toward the optimum. To apply second-order smoothing in the ξ_1 direction, for example, the smoothed gradient $\bar{\mathcal{G}}$ may be calculated from a discrete approximation to

$$\bar{\mathcal{G}} - \frac{\partial}{\partial \xi_1} \varepsilon \frac{\partial}{\partial \xi_1} \bar{\mathcal{G}} = \mathcal{G}, \tag{27}$$

where ε is the smoothing parameter. For higher order smoothing, a similar equation of higher order needs to be solved.

3.5. Implementation of Navier–Stokes design

In this paper, we implement the pseudo-time-stepping approach of the design for Navier–Stokes equations. The method is compared with the continuous adjoint method of Jameson [9] (termed as 'Original' in what follows). His design procedure can be summarized as follows:

1. Solve the flow equations (for 10 iterations) for ρ, u_1, u_2, p.
2. Solve the adjoint equations (for 10 iterations) for ψ subject to appropriate boundary conditions.
3. Evaluate $\mathcal{G}$.
4. Project $\mathcal{G}$ into an allowable subspace that satisfies any geometric constraints.
5. Smooth the gradient to get $\bar{\mathcal{G}}$.
6. Update the shape based on the direction of steepest descent.
7. Return to 1 until convergence is reached.

The design procedure of 'one-shot' pseudo-time-stepping method can be summarized as follows:

1. Solve the flow equations (for 2–4 iterations) for ρ, u_1, u_2, p.
2. Solve the adjoint equations (for 2–4 iterations) for ψ subject to appropriate boundary conditions.
3. Evaluate $\mathcal{G}$.
4. Project $\mathcal{G}$ into an allowable subspace that satisfies any geometric constraints.
5. Smooth the gradient to get $\bar{\mathcal{G}}$.
6. Approximate the reduced Hessian B.
7. Integrate the preconditioned design equation.
8. Update the shape.
9. Return to 1 until convergence is reached.

4. NUMERICAL RESULTS AND DISCUSSION

The numerical method has been tested by applying it to 2D test cases for reduction of total drag while maintaining constant lift and constant thickness. The computational domain is discretized using 512×64 C-grid. On this grid pseudo-unsteady state, costate and design equations are integrated in time. The SYN103 code of Jameson is used for the computations. The code has been modified and enhanced for the preconditioned pseudo-time-stepping optimization method. The constraint of constant lift is maintained by changing the angle of incidence at each iteration of the flow solution. Also, the additional constraint that the thickness cannot be reduced below a prescribed threshold is enforced by blocking shape modifications which would penetrate the threshold. Usually, the threshold is set as the thickness of the original profile. All the grid points on the airfoil are used as design parameters which are 257 in number. All the computations are carried out on a Linux machine with Intel(R) Xeon(TM) processor, CPU 3.00 GHz and 8 MB RAM. Comparisons are presented between the results of Jameson's original method as implemented in the unmodified version of SYN103. This method may also be properly regarded as a one-shot method because it is normally run without fully converging the flow and adjoint solutions except at the first design step.

DOI: 10.1002/fld

Table I. Comparison of number of iterations and force coefficients for baseline and optimized RAE2822 airfoil using different optimization iterations.

Geometry	Opt. Itr.	State Itr.	Adj. Itr.	CD (total)	CD_P	CD_V	CL	AL	CPU (time, s)
Baseline		300		0.0173	0.0118	0.0055	0.6500	1.93276	146.99
Original	8	230	150	0.0133	0.0078	0.0055	0.6498	2.16736	190.16
One-shot	32	222	102	0.0133	0.0078	0.0055	0.6501	2.12036	165.48

4.1. Case 1: *RAE 2822 airfoil*

In this case the optimization method is applied to an RAE2822 airfoil at Mach number 0.75 and Reynolds number $0.600E+07$. The constraint of constant lift coefficient is fixed at 0.65, while the minimum thickness constraint is fixed as the initial thickness with the consequence that there can be no reduction in thickness. The optimization is started after 80 iterations of the state and 40 iterations of the costate solver. The design equation is integrated after every 2 iterations of the state and costate runs instead of after every iteration as in the case of inviscid applications. Since the grid is very fine, the residual is not reduced to the minimum level required by the pseudo-time-stepping method. That is why we need 2 iterations of the state and costate solver in each optimization update. The optimization requires 32 iterations to converge. After the convergence of the optimization another 80 iterations of the state solver is carried out in order to get the force coefficients which are comparable to the results obtained by other optimization methods.

For the 'Original' method, the optimization is started after 80 iterations of the state and the costate solver. The optimization requires 8 iterations to converge. Table I presents a comparison of the number of iterations, force coefficients, angle of incidence (AL) and CPU time required in both the methods. As can be seen, optimized total drag coefficient (CD), which is the sum of the pressure drag (CD_P) and the viscous drag (CD_V), is the same in both the methods. The lift coefficients (CL) are also almost the same but the angle of incidence obtained by the pseudo-time-stepping method has a smaller value. Also, the pseudo-time stepping method requires slightly less CPU time to converge. The convergence histories of both the optimization methods are presented in Figure 1. The surface pressure distributions and Mach contours are presented in Figure 2. Both the optimized airfoils, surface pressure distributions and Mach contours look quite similar.

4.2. Case 2: *TAI airfoil*

In this case the optimization method is applied to a TAI airfoil at Mach number 0.65 and Reynolds number $0.600E+07$. The constraint of constant lift coefficient is fixed at 0.75. The optimization is started after 80 iterations of the state and 40 iterations of the costate solver. The design equation is integrated after every 4 iterations of the state and costate runs. The optimization requires 38 iterations to converge. After the convergence of the optimization another 100 iterations of the state solver are carried out in order to get the force coefficients which are comparable to the results obtained using other optimization methods.

For the 'Original' method, the optimization is started after 80 iterations of the state and the costate solver. The optimization requires 16 iterations to converge. Table II presents a comparison of the number of iterations, force coefficients, angle of incidence and the CPU time required for the convergence of the methods. In this case also the force coefficients are almost the same, the angle of incidence obtained by the pseudo-time-stepping method and the CPU time required by this method is a little less than that resulted by the original method. The convergence histories of the optimization methods are presented in Figure 3. The surface pressure distributions and the Mach contours are presented in Figure 4. The optimized quantities obtained by both the methods are again very similar.

DOI: 10.1002/fld

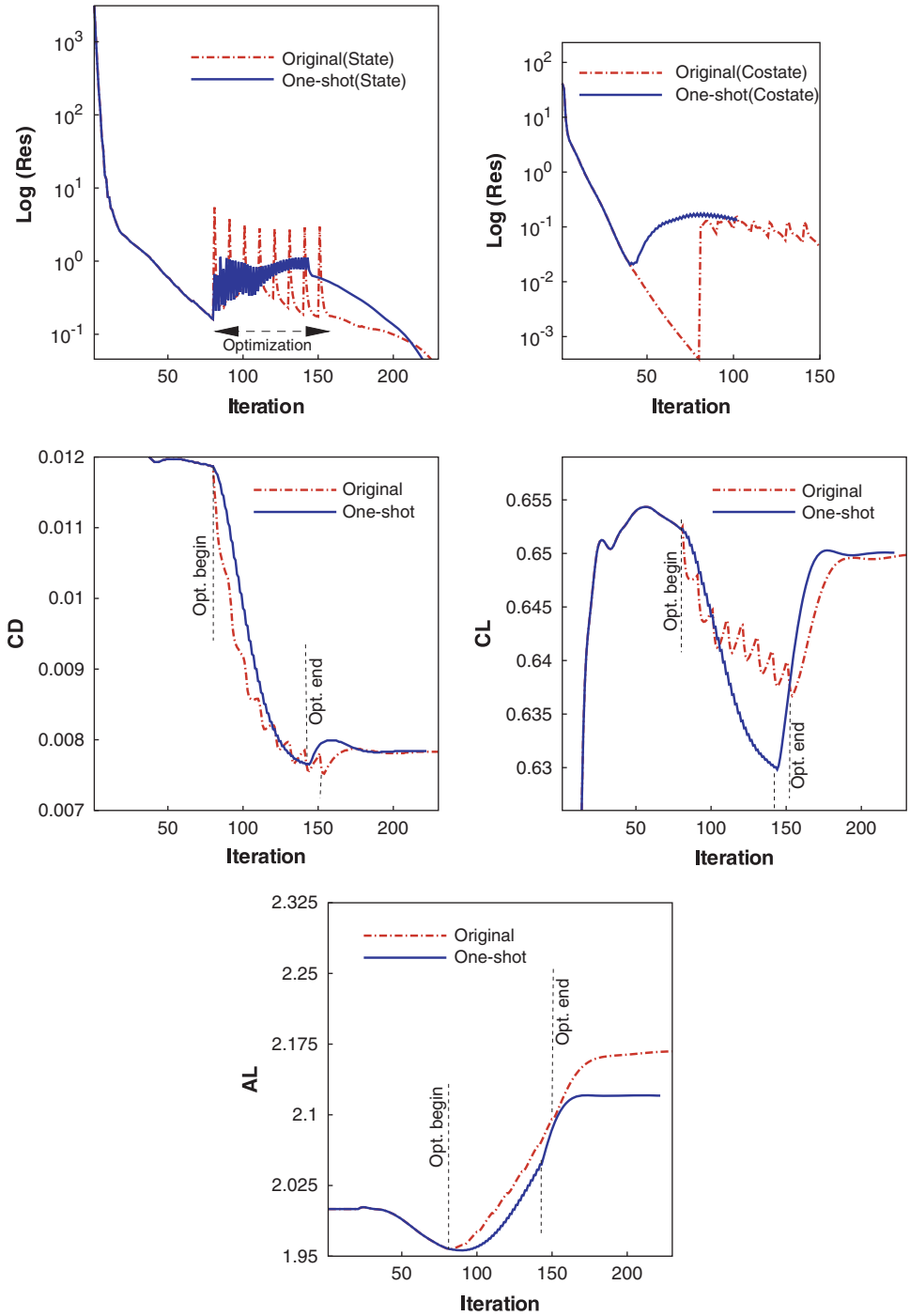
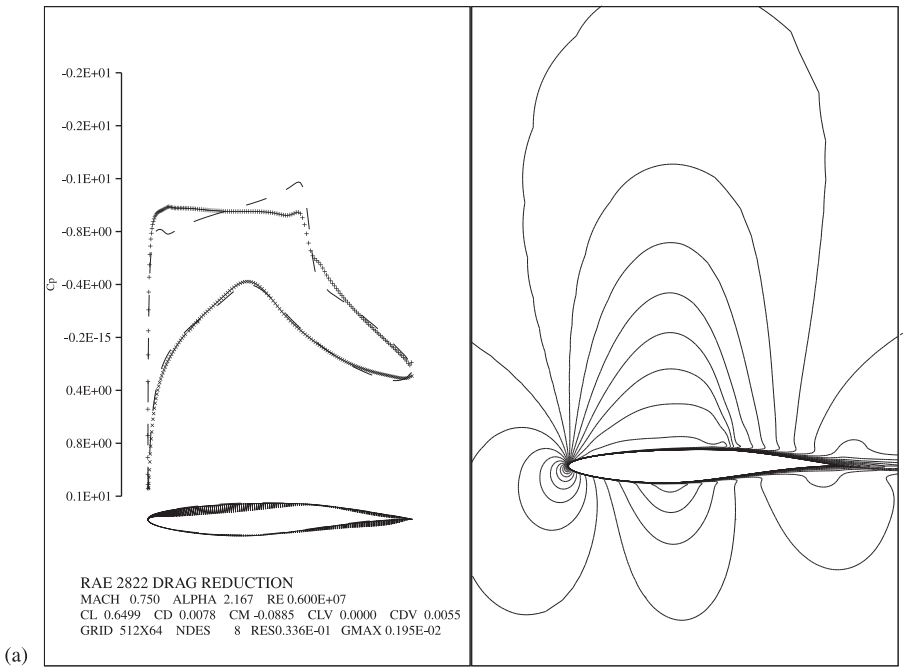


Figure 1. Convergence history of the optimization iterations (Case 1: RAE2822 airfoil).

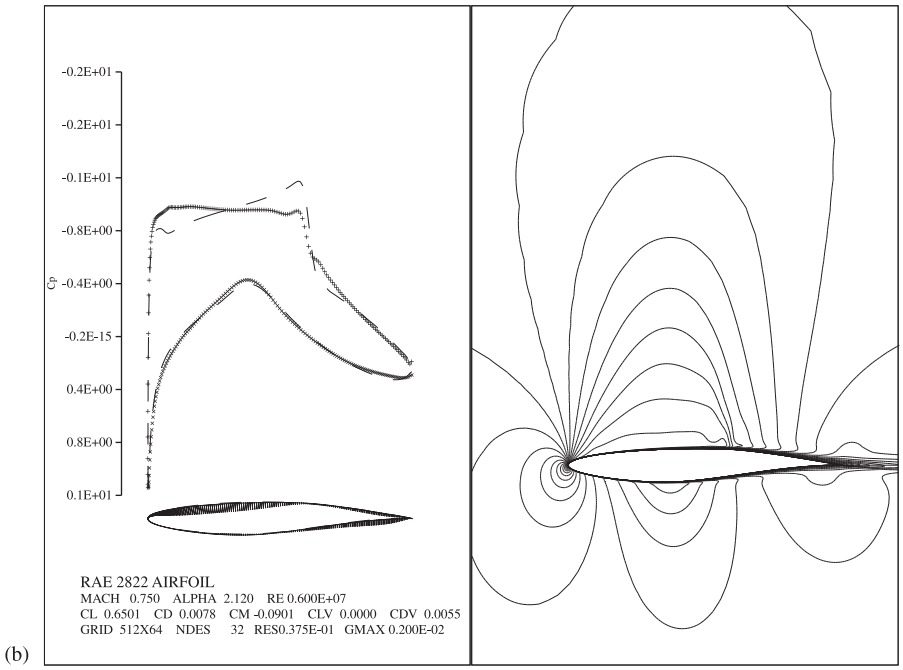
5. CONCLUSIONS

The one-shot pseudo-time-stepping method has been applied successfully to shape optimization problems in aerodynamics using viscous compressible flow. The method works efficiently as has previously been demonstrated for applications using inviscid compressible flow. In the convergence

DOI: 10.1002/fld



(a)



(b)

Figure 2. Pressure distribution and Mach contours for the RAE2822 airfoil: (a) original optimization and (b) one-shot optimization.

Table II. Comparison of number of iterations and force coefficients for baseline and optimized TAI airfoil using different optimization iterations.

Geometry	Opt. Itr.	State Itr.	Adj. Itr.	CD (Total)	CD_P	CD_V	CL	AL	CPU (time, s)
Baseline		500		0.0335	0.0282	0.0053	0.7504	2.28576	244.94
Original	16	310	230	0.0155	0.0099	0.0056	0.7508	2.40356	273.77
One-shot	38	328	188	0.0156	0.0100	0.0056	0.7506	2.38676	258.26

DOI: 10.1002/fld

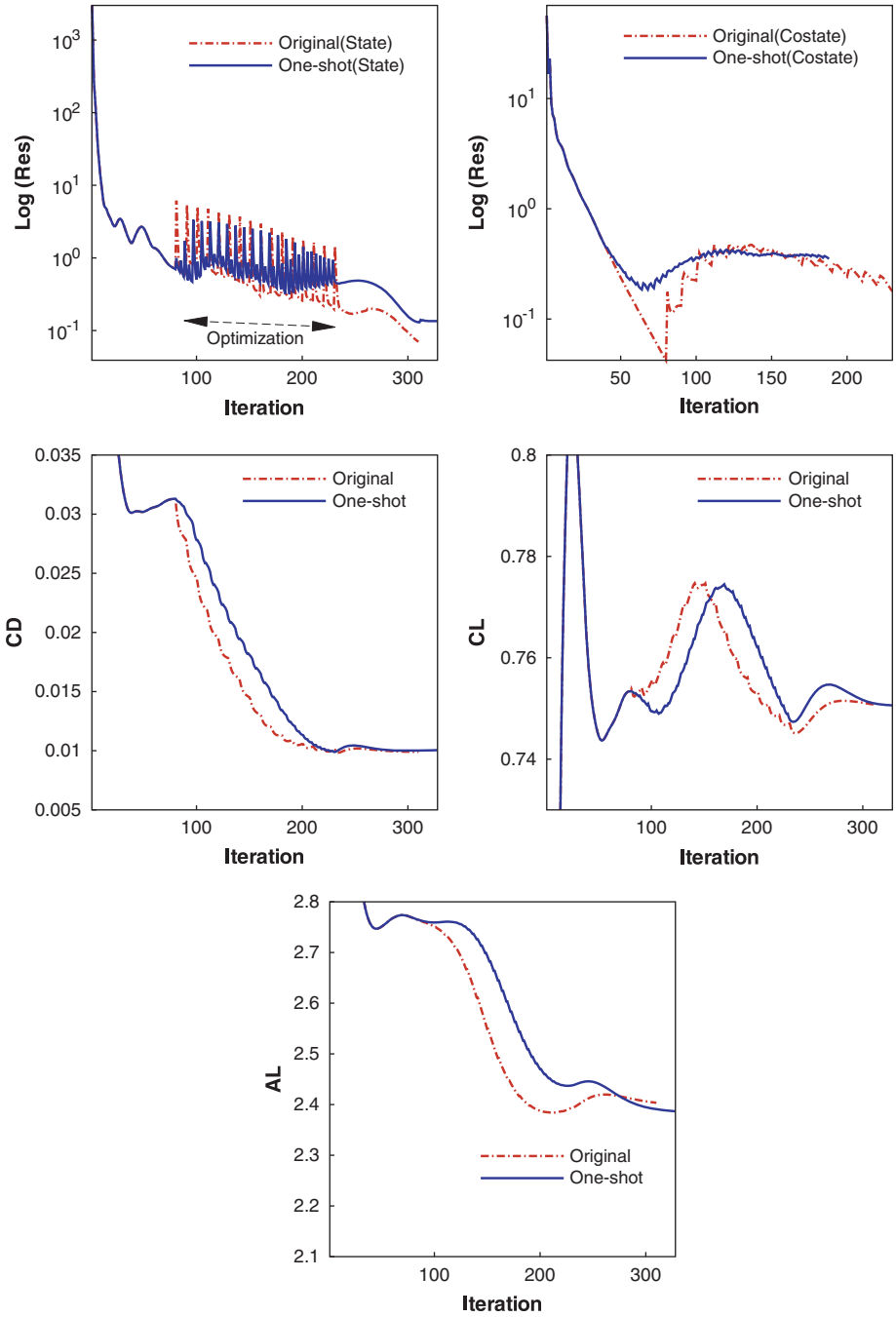


Figure 3. Convergence history of the optimization iterations (Case 2: TAI airfoil).

histories of the one-shot method, linear convergence with respect to the objective function is observed. The iteration step in the design space is so small that the process truly reflects a continuous behavior.

The optimized shapes from the two methods are very similar and have the same performance in this respect it is important to note that there is no reason to believe in the uniqueness of the optimum shape since any shock-free airfoil should have the same performance as long as the skin friction remains the same. The two methods have roughly equal computational costs, depending on tuning data parameters such as step size, smoothing parameters, number of iterations in the

DOI: 10.1002/fld

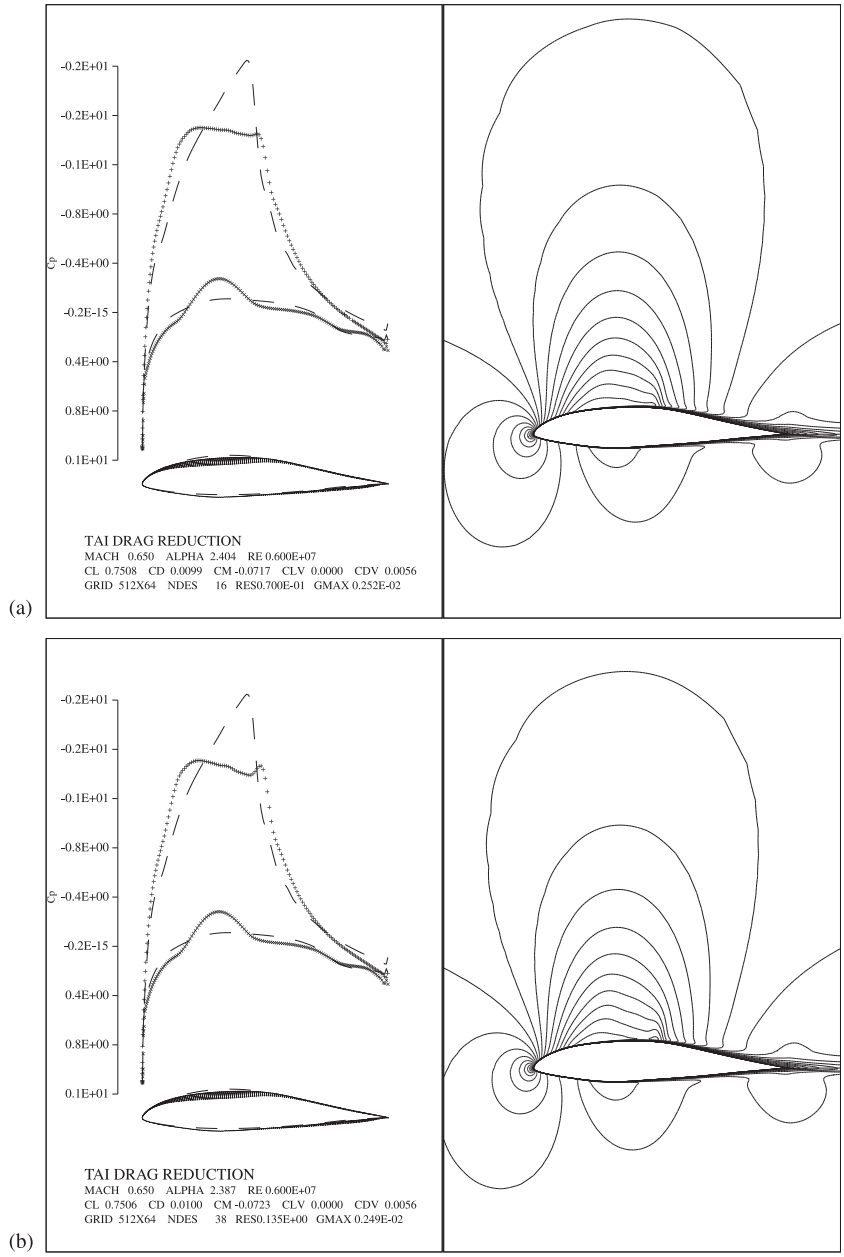


Figure 4. Pressure distribution and Mach contours for the TAI airfoil: (a) original optimization and (b) one-shot optimization.

flow and adjoint solutions, etc. Perhaps this is not so surprising when it is recalled that the original method can also be regarded as a variant of a one-shot method. The overall computational cost is less than 2 times the forward simulation runs. The future goal is to extend the method to more challenging problems in the viscous compressible flow.

ACKNOWLEDGEMENTS

The first author thanks Dr Sriram Shankaran for all his discussions and cooperations during his visit to Stanford. The second author is grateful for the support of NASA Dryden through an STTR contract under Dr Kajal Gupta. Thanks are due to the anonymous referees for their comments and suggestions on this work.

DOI: 10.1002/fld

REFERENCES

1. Lions JL. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer: New York, 1971.
2. Pironneau O. On optimum design in fluid mechanics. *Journal of Fluid Mechanics* 1974; **64**:97–110.
3. Pironneau O. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics* 1973; **59**:117–128.
4. Pironneau O. *Optimal Shape Design for Elliptic Systems*. Springer: New York, 1982.
5. Jameson A. Aerodynamic design via control theory. *Journal of Scientific Computing* 1988; **3**:233–260.
6. Jameson A. Automatic design of transonic airfoils to reduce shock induced pressure drag. *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics*, Tel Aviv, February 1990; 5–17.
7. Jameson A. Optimum aerodynamic design using CFD and control theory. *AIAA 12th Computational Fluid Dynamics Conference*, San Diego, *AIAA 95-1729-CP*, June 1995.
8. Jameson A. Optimum aerodynamic design using control theory. In *Computational Fluid Dynamics Review*, Hafez M, Oshima K (eds). Wiley: New York, 1995; 495–528.
9. Jameson A. Efficient aerodynamic shape optimization. *The 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, *AIAA Paper 2004-4369*, 30 August–1 September 2004.
10. Jameson A, Pierce N, Martinelli L. Optimum aerodynamic design using the Navier–Stokes equations. *The 35th Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 97-0101*, 1997.
11. Jameson A, Martinelli L, Pierce N. Optimum aerodynamic design using the Navier–Stokes equations. *Journal of Theoretical Computational Fluid Dynamics* 1998; **10**:213–237.
12. Iollo A, Kuruvila G, Ta'asan S. Pseudo-time method for optimal shape design using Euler equations. *AIAA Journal* 1996; **34**(9):1807–1813.
13. Ta'asan S. Pseudo-time methods for constrained optimization problems governed by PDE. *ICASE Report No. 95-32*, 1995.
14. Hazra SB, Schulz V. Simultaneous pseudo-timestepping for PDE-model based optimization problems. *Bit Numerical Mathematics* 2004; **44**(3):457–472.
15. Hazra SB. Reduced Hessian updates in simultaneous pseudo-timestepping for aerodynamic shape optimization. *The 44th AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 2006-933*, 9–12 January 2006.
16. Hazra SB. An efficient method for aerodynamic shape optimization. *The 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, *AIAA Paper 2004-4628*, 30 August–September 1 2004.
17. Hazra SB, Gauger N. Simultaneous pseudo-timestepping for aerodynamic shape optimization. *PAMM* 2005; **5**(1):743–744.
18. Hazra SB, Schulz V, Brezillon J, Gauger NR. Aerodynamic shape optimization using simultaneous pseudo-timestepping. *Journal of Computational Physics* 2005; **204**:46–64.
19. Hazra SB, Schulz V. Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. *SIAM Journal on Scientific Computing* 2006; **28**(3):1078–1099.
20. Hazra SB, Schulz V. Simultaneous pseudo-timestepping for state constrained optimization problems in aerodynamics. In *Real-time PDE-constrained Optimization*, Biegler L, Ghattas O, Heinkenschloss M, Keyes D, van Bloemen Waanders B (eds). SIAM: Philadelphia, PA, 2007; 169–180.
21. Hazra SB, Schulz V, Brezillon J. Simultaneous pseudo-timestepping for 3D aerodynamic shape optimization. *Journal of Numerical Mathematics* 2008; **16**(2):139–161.
22. Rizk MH. Optimizing advanced propeller designs by simultaneously updating flow variables and design parameters. *AIAA Journal* 1989; **26**(6):515–522.
23. Kuruvila G, Ta'asan S, Salas M. Airfoil optimization by the one-shot method. *AGARD-FDP-VKI*, Special Course on Optimum Design Methods in Aerodynamics, April 1994.
24. Ta'asan S, Kuruvila G, Salas MD. Aerodynamic design and optimization in one shot. *30th Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 92-0025*, January 1992.
25. Hazra SB. Multigrid one-shot method for aerodynamic shape optimization. *SIAM Journal on Scientific Computing* 2008; **30**(3):1527–1547.
26. Baldwin B, Lomax H. Thin layer approximation and algebraic model for separated turbulent flow. *16th AIAA Aerospace Science Meeting*, Reno, NV, *AIAA Paper 78-257*, 1978.
27. Reuther J, Jameson A. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *33rd AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 95-0123*, January 1995.
28. Reuther J, Jameson A, Farmer J, Martinelli L, Saunders D. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *34th AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 96-0094*, January 1996.
29. Anderson WK, Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *35th AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA 97-0643*, 1997.
30. Shankaran S, Jameson A. Aerodynamic and aeroelastic applications of a parallel, multigrid, unstructured flow solver. *The 41st AIAA Aerospace Science Meeting and Exhibit*, Reno, NV, *AIAA Paper 2003-0555*, 2003.
31. Shankaran S, Jameson A. Numerical analysis and design of upwind sails. *The 21st AIAA Applied Aerodynamics Conference*, Orlando, *AIAA Paper 2003-3531*, 2003.
32. Battermann A, Sachs EW. Block preconditioners for KKT systems in PDE-governed optimal control problems. *Fast Solution of Discretized Optimization Problems*, Hoffmann KH, Hoppe RHW, Schulz V (eds). Birkhäuser, 2001; 1–18.

DOI: 10.1002/fld

33. Battermann A, Heinkenschloss M. Preconditioners for Karush–Kuhn–Tucker systems arising in the optimal control of distributed systems. In *Optimal Control of PDE*, *Vorau 1996*, Desch W, Kappel K, Kunisch K (eds). Birkhäuser, 1996; 15–32.
34. Biros G, Ghattas O. Parallel Lagrange–Newton–krylov–schur methods for PDE-constrained optimization. Part I: the Krylov–Schur solver. *Technical Report*, *Laboratory for Mechanics*, *Algorithms and Computing*, Carnegie Mellon University, 2000.
35. Jameson A. Transonic flow calculations for aircraft. In *Numerical Methods in Fluid Dynamics*, Brezzi F (ed.). Lecture Notes in Mathematics, vol. 1127. Springer: Berlin, 1983; 156–242.
36. Thomas JL, Salas MD. Far field boundary conditions for transonic lifting solutions to the Euler equations. *AIAA Journal* 1996; **24**(7):1074–1080.
37. Nadarajah S, Jameson A. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *The 15th AIAA Computational Fluid Dynamics Conference*, Anaheim, CA, 11–14 June 2001.

DOI: 10.1002/fld