45th AIAA Aerospace Sciences Meeting and Exhibit
8 - 11 January 2007, Reno, Nevada

AIAA 2007-1470

One-Shot Pseudo-Time Method for Aerodynamic Shape Optimization Using the Navier-Stokes Equations

S. B. Hazra[1], A. Jameson[2]

[1] Department of Mathematics, University of Trier, D-54286 Trier, Germany

[2] Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA

Abstract. 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 method for state and 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.

I. 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 best results in applications to practical problems.

Aerodynamic shape optimization problems can mathematically be formulated as control problem governed by system of PDEs. For these problems adjoint system of PDEs can be formulated and solved using the same algorithm applicable to forward or state system. The gradient is computed using the state and adjoint solutions much efficiently in comparison to finite-difference methods. Application of this continuous adjoint method is carried out first in[14]-[17] for transonic flow using Euler equations. These 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 over all cost of computation is quite high in these methods. A. Jameson has proposed gradient smoothing. Instead of using the 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 find the search direction. It turns out that this approach is tolerant to use inexact gradient so that neither flow solution nor the adjoint solution need to be fully converged. Extension of the continuous adjoint method for shape optimization problems in viscous compressible flow is carried out in[18]-.[20]

In[9] we proposed a new method for solving such optimization problems using simultaneous pseudo-timestepping. In[6]-[10] we have applied the method for solving aerodynamic shape optimization problems without additional state constraint and in[11]-[13] applied the method to problems with additional state constraints. The overall cost of computation in all the applications has been between 2-8 times as that of the forward simulation runs. Further efficiency is achieved using 'optimization-based' multigrid method in.[5]

All the above mentioned applications of pseudo-timestepping method have been in Euler equations. In this paper we extend the method to viscous compressible flow modeled by Reynolds Averaged Navier-Stokes equations together with algebraic turbulence model of Baldwin and Lomax. While inviscid formulations are useful for the design in transonic cruise conditions, inclusion of viscous effects are essential for optimal design encompassing off-design conditions and high-lift configurations. The computational complexity in viscous design is at least an order of magnitude greater than that in inviscid design since the number of mesh points are to be increased by a factor of two or more to resolve the boundary layer. The convergence of Navier-Stokes solver is much more slower than Euler solver due to discrete stiffness and directional decoupling arising from the highly stretched boundary layer cells. Since we use inaccurate state and costate solutions, and hence inaccurate gradients, in our one-shot pseudo-timestepping method, therefore slow convergence of Navier-Stokes (forward and adjoint) solver may affect the convergence of this method. We investigate that numerically in this paper.

The paper is organized as follows. In the next 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.

II. The optimization problem and pseudo-unsteady formulation of the KKT conditions

The focus of the present work is on aerodynamic shape optimization problems which are large scale PDE constrained optimization problems. These problems can be written in abstract form (see in[9, 10]) as

$$\begin{aligned} &\min\ I(w,q) \\ &\text{s. t.}\quad c(w,q)=0. \end{aligned} \tag{1}$$

Here, $c(w,q)=0$ represents the steady-state flow equations (in our case Navier-Stokes equations) together with 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 purposes of this paper.

The necessary optimality conditions (known as KKT conditions) are

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

$$\nabla_w L(w,q,\lambda)=0, \qquad \textit{(Costate equation)} \tag{2b}$$

$$\nabla_q L(w,q,\lambda)=0. \qquad \textit{(Design equation)} \tag{2c}$$

where

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

is the Lagrangian functional and λ is the Lagrange multiplier or the adjoint variable. 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[15,16,22,23] on structured grids. An application of this method on unstructured grids has been presented in.[1]

One can use, for example, RSQP methods to solve the above set of equations. 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

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

We used in[7,9,10] 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

$$\begin{aligned} \frac{dw}{dt} + c(w,q) &= 0, \\ \frac{d\lambda}{dt} + \nabla_w L(w,q,\lambda) &= 0, \\ \frac{dq}{dt} + \nabla_q L(w,q,\lambda) &= 0. \end{aligned} \tag{5}$$

This formulation is advantageous since the steady-state flow (and adjoint) solution is obtained by integrating the pseudo-unsteady Euler (and adjoint Euler) equations in this problem class. Therefore, one can use the same time-stepping scheme 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.[9]

The pseudo-time embedded system (5) usually results (after semi-discretization) 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 the inverse of the matrix in equation (4) as a preconditioner for the time-stepping process. The pseudo-time embedded system that we consider is

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

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, it is computationally inexpensive. The preconditioner employed is similar to the preconditioners for KKT-systems discussed in[2,3] in the context of Krylov subspace methods and in[4] 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 reduced Hessian. As shown in,[6] the reduced Hessian update based on most recent reduced gradient and parameter update informations is good enough, we use the same update strategy in this paper. We define $s_k := (q_{k+1} - q_k)$ and $z_k := (\nabla I_{k+1} - \nabla I_k)$, where k represents the iteration number. Then the reduced Hessian update is based on the sign of the product $(z_k^T s_k)$. If the sign is positive, the reduced Hessian is approximated by

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

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

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

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.

III. Detailed equations of the aerodynamic shape optimization problem

In this section we explain briefly the state, costate and design equations represented in equations (2) for the shape optimization problem. The formulation of the adjoint and gradient equations for viscous optimization follows the development in references[18] and.[19]

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, y) for two-dimensional flow can be written in integral form for the region $\mathcal{D}$ with boundaries $\mathcal{B}$ as

$$\frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial x_i} = \frac{\partial f_{vi}}{\partial x_i} \text{ in } \mathcal{D}, \tag{7}$$

where

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

For a perfect gas the pressure and total enthalpy is 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.

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

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

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

$$\frac{\partial (Jw)}{\partial t} + \frac{\partial (F_i - F_{vi})}{\partial \xi_i} = 0 \text{ in } \mathcal{D}, \tag{8}$$

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 equations (8) 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.

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}.

Suppose that the performance is measured by a cost function

$$I = \int_{\mathcal{B}} \mathcal{M}(w, S)\, dB_\xi + \int_{\mathcal{D}} \mathcal{P}(w, S)\, dD_\xi, \tag{9}$$

containing both boundary and field contributions where dB_ξ and dD_ξ are the surface and volume elements in the computational domain. In general, $\mathcal{M}$ and $\mathcal{P}$ 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 (8). 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

$$\delta I = \int_{\mathcal{B}} \delta\mathcal{M}(w, S)\, dB_\xi + \int_{\mathcal{D}} \delta\mathcal{P}(w, S)\, dD_\xi. \tag{10}$$

This can be split as

$$\delta I = \delta I_I + \delta I_{II}, \tag{11}$$

with

$$\begin{aligned} \delta \mathcal{M} &= \left[\mathcal{M}_w\right]_I \delta w + \delta \mathcal{M}_{II}, \\ \delta \mathcal{P} &= \left[\mathcal{P}_w\right]_I \delta w + \delta \mathcal{P}_{II}, \end{aligned} \tag{12}$$

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 $\left[\mathcal{M}_w\right]_I$ and $\left[\mathcal{P}_w\right]_I$ represent $\frac{\partial \mathcal{M}}{\partial w}$ and $\frac{\partial \mathcal{P}}{\partial w}$ with the metrics fixed, while $\delta \mathcal{M}_{II}$ and $\delta \mathcal{P}_{II}$ represent the contribution of the metric variations δS to $\delta \mathcal{M}$ and $\delta \mathcal{P}$.

In the steady state, the constraint equation (53) specifies the variation of the state vector δw by

$$\delta R = \frac{\partial}{\partial \xi_i} \delta \left(F_i - F_{vi}\right) = 0. \tag{13}$$

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

$$\begin{aligned} \delta R &= \delta R_I + \delta R_{II} \\ \delta F_i &= \left[F_{iw}\right]_I \delta w + \delta F_{i\,II} \\ \delta F_{vi} &= \left[F_{viw}\right]_I \delta w + \delta F_{vi\,II}. \end{aligned} \tag{14}$$

The inviscid contributions are easily evaluated as

$$\left[F_{iw}\right]_I = S_{ij} \frac{\partial f_i}{\partial w}, \qquad \delta F_{vi\,II} = \delta S_{ij} f_j.$$

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

Multiplying by a co-state vector ψ, which will play an analogous role to the Lagrange multiplier, and integrating over the domain produces

$$\int_{\mathcal{D}} \psi^T \frac{\partial}{\partial \xi_i} \delta \left(F_i - F_{vi}\right) d\mathcal{D}_\xi = 0. \tag{15}$$

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

$$\int_{\mathcal{B}} n_i \psi^T \delta \left(F_i - F_{vi}\right)_I d\mathcal{B}_\xi - \int_{\mathcal{D}} \frac{\partial \psi^T}{\partial \xi_i} \delta \left(F_i - F_{vi}\right)_I d\mathcal{D}_\xi + \int_{\mathcal{D}} \psi^T \delta R_{II} d\mathcal{D}_\xi = 0. \tag{16}$$

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. Since the left hand expression equals zero, it may be subtracted from the variation in the cost function (10) to give

$$\begin{aligned} \delta I &= \delta I_{II} - \int_{\mathcal{D}} \psi^T \delta R_{II} d\mathcal{D}_\xi - \int_{\mathcal{B}} \left[\delta \mathcal{M}_I - n_i \psi^T \delta \left(F_i - F_{vi}\right)_I\right] d\mathcal{B}_\xi \\ &+ \int_{\mathcal{D}} \left[\delta \mathcal{P}_I + \frac{\partial \psi^T}{\partial \xi_i} \delta \left(F_i - F_{vi}\right)_I\right] d\mathcal{D}_\xi. \end{aligned} \tag{17}$$

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 (12) and (14), the variation δw may be eliminated from (17) by equating all field terms with subscript "I" to produce a differential adjoint system governing ψ

$$\frac{\partial \psi^T}{\partial \xi_i} \left[F_{iw} - F_{viw} \right]_I \; + \; \left[\mathcal{P}_w \right]_I = 0 \quad \text{in } \mathcal{D}. \tag{18}$$

The corresponding adjoint boundary condition is produced by equating the subscript "I" boundary terms in equation (17) to produce

$$n_i \psi^T \left[F_{iw} - F_{viw} \right]_I = \left[\mathcal{M}_w \right]_I \quad \text{on } \mathcal{B}. \tag{19}$$

The remaining terms from equation (17) then yield a simplified expression for the variation of the cost function which defines the gradient

$$\delta I = \delta I_{II} + \int_{\mathcal{D}} \psi^T \delta R_{II} d\mathcal{D}_\xi, \tag{20}$$

which 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.

Comparing equations (12) and (14), the variation δw may be eliminated from (17) by equating all field terms with subscript "I" to produce a differential adjoint system governing ψ

$$\frac{\partial \psi^T}{\partial \xi_i} \left[F_{iw} - F_{viw} \right]_I \; + \; \mathcal{P}_w = 0 \quad \text{in } \mathcal{D}. \tag{21}$$

The corresponding adjoint boundary condition is produced by equating the subscript "I" boundary terms in equation (17) to produce

$$n_i \psi^T \left[F_{iw} - F_{viw} \right]_I = \mathcal{M}_w \quad \text{on } \mathcal{B}. \tag{22}$$

The remaining terms from equation (17) then yield a simplified expression for the variation of the cost function which defines the gradient

$$\delta I = \int_{\mathcal{B}} \left\{ \delta \mathcal{M}_{II} - n_i \psi^T \left[\delta F_i - \delta F_{vi} \right]_{II} \right\} d\mathcal{B}_\xi + \int_{\mathcal{D}} \left\{ \delta \mathcal{P}_{II} + \frac{\partial \psi^T}{\partial \xi_i} \left[\delta F_i - \delta F_{vi} \right]_{II} \right\} d\mathcal{D}_\xi. \tag{23}$$

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

$$\delta I = \int_{\mathcal{B}} \mathcal{G} \delta q \; d\mathcal{B}_\xi$$

where q represents the design variables, and $\mathcal{G}$ 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 (22). Consequently, the boundary contribution to the cost function $\mathcal{M}$ cannot be specified arbitrarily. Instead, it must be chosen from the class of functions which allow cancellation of all terms containing δw in the boundary integral of equation (17). On the other hand, there is no such restriction on the specification of the field contribution to the cost function $\mathcal{P}$, since these terms may always be absorbed into the adjoint field equation (21) as source terms.

It is convenient to develop the inviscid and viscous contributions to the adjoint equations separately. Also, for simplicity, it will be assumed that the portion of the boundary that undergoes shape modifications is restricted to the coordinate surface $\xi_2 = 0$. Then equations (17) and (19) may be simplified by incorporating the conditions

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

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

The inviscid and viscous contributions are to be derived separately to get the final form of the adjoint equations. The inviscid contributions are derived in.[15, 17] The viscous contributions are derived in[18, 19] and can be found in[21] as well for 2D case. Determining the contributions from momentum and energy equations, the viscous adjoint field operator will read as follows:

$$\begin{aligned}
\left(\bar{L}\psi\right)_1 &= -\frac{p}{\rho^2}\frac{\partial}{\partial \xi_l}\left(S_{lj}\kappa\frac{\partial \theta}{\partial x_j}\right)\\
\left(\bar{L}\psi\right)_{i+1} &= \frac{\partial}{\partial \xi_l}\left\{S_{lj}\left[\mu\left(\frac{\partial \phi_i}{\partial x_j}+\frac{\partial \phi_j}{\partial x_i}\right)+\lambda\delta_{ij}\frac{\partial \phi_k}{\partial x_k}\right]\right\}\\
&+ \frac{\partial}{\partial \xi_l}\left\{S_{lj}\left[\mu\left(u_i\frac{\partial \theta}{\partial x_j}+u_j\frac{\partial \theta}{\partial x_i}\right)+\lambda\delta_{ij}u_k\frac{\partial \theta}{\partial x_k}\right]\right\}-\sigma_{ij}S_{lj}\frac{\partial \theta}{\partial \xi_l} \quad \text{for} \quad i=1,2\\
\left(\bar{L}\psi\right)_4 &= \frac{1}{\rho}\frac{\partial}{\partial \xi_l}\left(S_{lj}\kappa\frac{\partial \theta}{\partial x_j}\right).
\end{aligned}$$

The conservative viscous adjoint operator is obtained by the transformation

$$L = M^{-1^T}\bar{L}$$

where

$$M^{-1^T} = \begin{pmatrix} 1 & -\frac{u_1}{\rho} & -\frac{u_2}{\rho} & \frac{(\gamma-1)u_iu_i}{2}\\ 0 & \frac{1}{\rho} & 0 & -(\gamma-1)u_1\\ 0 & 0 & \frac{1}{\rho} & -(\gamma-1)u_2\\ 0 & 0 & 0 & (\gamma-1) \end{pmatrix}.$$

The cost function that we choose in the present optimization problem is drag reduction. Hence, the cost function, which corresponds equation (9), reads as

$$I(w,q) := C_D = \frac{1}{C_{ref}}\int_{\mathcal{B}} C_p\left(\frac{\partial y}{\partial \xi}\cos\alpha - \frac{\partial x}{\partial \xi}\sin\alpha\right)d\xi_1, \tag{24}$$

where the surface pressure coefficient is defined by

$$C_p := \frac{2(p-p_\infty)}{\gamma M_\infty^2 p_\infty}. \tag{25}$$

The boundary conditions for the adjoint equations on the solid body, corresponding to equation (22), are of Neumann-type and for the above mentioned cost function they are given by

$$n_{\xi_1}\psi_2 + n_{\xi_2}\psi_3 = -\frac{2}{\gamma M_\infty^2 p_\infty C_{\text{ref}}}(n_{\xi_1}\cos\alpha + n_{\xi_2}\sin\alpha), \quad \text{on } \mathcal{B}. \tag{26}$$

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

$$\delta I = \frac{1}{C_{ref}} \int_{\mathcal{B}} C_p \left(\delta\left(\frac{\partial y}{\partial \xi}\right)\cos\alpha - \delta\left(\frac{\partial x}{\partial \xi}\right)\sin\alpha \right) d\xi_1. \tag{27}$$

Gradient Smoothing: The reduced gradient obtained using inaccurate state and costate solutions are quite non-smooth, specially 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 $\mathcal{G}$ by its smoothed value $\bar{\mathcal{G}}$ in the descent process. This also acts as a preconditioner which allows the use of much larger steps. The gradient smoothing is equivalent to redefining the inner product in a Sobolev space as described in,[20] and the steps in the smoothed gradient direction still guarantee descent towards 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

$$\bar{\mathcal{G}} - \frac{\partial}{\partial \xi_1}\epsilon\frac{\partial}{\partial \xi_1}\bar{\mathcal{G}} = \mathcal{G} \tag{28}$$

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

Implementation of Navier-Stokes Design: In this paper we implement the one-shot approach of the design for Navier-Stokes equations. The method is compared with the continuous adjoint method of A. Jameson and 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' method can be summarized as follows:

1. Solve the flow equations (for 2 to 4 iterations) for ρ, u_1, u_2, p.
2. Solve the adjoint equations (for 2 to 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.

IV. Numerical results and discussion

The numerical method is applied to 2D test cases for drag reduction with constant lift and constant thickness. The computational domain is discretized using $512x64$ C-grid. On this grid pseudo-unsteady state, costate and design equations are solved. SYN103 code of A. Jameson is used for the computations. The code is modified for one-shot optimization. The constraint of constant lift is maintained by changing angle of incidence. All the points on the airfoil are used as design parameters which are 257 in numbers. All the computations are carried out on a Linux machine with Intel(R) Xeon(TM) processor, CPU 3.00GHz and 8MB RAM.

Case 1: RAE 2822 airfoil
In this case the optimization method is applied to RAE 2822 airfoil at Mach number 0.75 and Reynolds number $0.600E + 07$. The constraint of constant lift coefficient is fixed at 0.65. The optimization is started after 80 iterations of state and 40 iterations of costate solver. The design equation is integrated after every 2 iterations of state and costate runs instead of after every iteration of state and costate run as it should be in order to call it one-shot method. Since the grid is very fine, the residual is not reduced to the minimum level required by the one-shot method. That is why we need 2 iterations of 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 state solver is carried out in order to get the force coefficients which are comparable with results obtained by other methods.

For the 'Original' method, the optimization is started after 80 iterations of state and costate solver. The optimization requires 8 iterations to converge. Table 1 presents the comparison of number of iterations, force coefficients and CPU time required in both the methods. As can be seen, optimized drag coefficients are same in both the methods. The lift coefficients are also almost the same but the angle of incidence obtained by one-shot method has smaller value. Also, one-shot method requires 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.

Geometry	Opt. Itr	State Itr	Costate Itr	CD	CL	AL	CPU time
Baseline		300		0.0118	0.6500	1.93276	146.99 Sec
Original	8	230	150	0.0078	0.6498	2.16736	190.16 Sec
One-shot	32	222	102	0.0078	0.6501	2.12036	165.48 Sec

Table 1. *Comparison of number of iterations and force coefficients for baseline and optimized RAE2822 airfoil using different optimization iterations*

Case 2: TAI airfoil

In this case the optimization method is applied to 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 state and 40 iterations of costate solver. The design equation is integrated after every 4 iterations of state and costate runs. The optimization requires 38 iterations to converge. After the convergence of the optimization another 100 iterations of state solver is carried out in order to get the force coefficients which are comparable with results obtained with other methods.

For the 'Original' method, the optimization is started after 80 iterations of state and costate solver. The optimization requires 16 iterations to converge. Table 2 presents the comparison of number of iterations, force coefficients and the CPU time required for the convergence of the methods. In this case force coefficients are almost the same, angle of incidence obtained by one-shot method and the CPU time required by this method is little less than that resulted by the original method. The convergence history of the optimization method is presented in Figure 3. The surface pressure distribution and Mach contour are presented in Figure 4. The optimized quantities obtained by both the methods are again quite similar.

Geometry	Opt. Itr	State Itr	Costate Itr	CD	CL	AL	CPU time
Baseline		500		0.0282	0.7504	2.28576	244.94 Sec
Original	16	310	230	0.0099	0.7508	2.40356	273.77 Sec
One-shot	38	328	188	0.0100	0.7506	2.38676	258.26 Sec

Table 2. *Comparison of number of iterations and force coefficients for baseline and optimized TAI airfoil using different optimization iterations*

V. Conclusions

One-shot pseudo-timestepping method is applied successfully to shape optimization problems in aerodynamics using viscous compressible flow. The method works efficiently as in case of applications using inviscid compressible flow. In the convergence 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 there is a unique 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, no of iterations in the flow and adjoint solutions, etc. Perhaps this is not so surprising because when the original method is run without fully converging the intermediate flow and adjoint solutions it can properly be regarded as a variant of a one shot method. The over all computational cost is less than 2 times the forward simulation runs.

VI. Acknowledgments

The first author would like to thank Dr. Sriram Shankaran for all his discussions and cooperations during his visit to Stanford.

References

[1] Anderson, W. K., Venkatakrishnan, V., *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation*, AIAA 97-0643, 1997.

[2] 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, W. Desch, F. Kappel and K. Kunisch eds., Birkhäuser Verlag, pp.15-32, 1996.

[3] Battermann, A., Sachs, E. W., *Block preconditioners for KKT systems in PDE-governed optimal control problems,* in Fast Solution of Discretized Optimization Problems, K. H. Hoffmann, R. H. W. Hoppe and V. Schulz eds., Birkhäuser Verlag, pp. 1-18, 2001.

[4] Biros, G., Ghattas, O., *Parallel Lagrange-Newton-krylov-schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, Tech. Rep., Laboratory for Mechanics, Algorithms and Computing, Carnegie Mellon University, 2000.

[5] Hazra, S. B., *Multigrid one-shot method for aerodynamic shape optimization*, (submitted), 2006.

[6] Hazra, S. B., *Reduced Hessian updates in simultaneous pseudo-timestepping for aerodynamic shape optimization*, 44th AIAA Aerospace Science Meeting and Exhibit, AIAA 2006-933, 9-12 January, Reno, Nevada, 2006.

[7] Hazra, S. B., *An efficient method for aerodynamic shape optimization*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA paper 2004-4628, Albany, New York, Aug.30-Sep.1, 2004.

[8] Hazra, S. B., Gauger, N., *Simultaneous pseudo-timestepping for aerodynamic shape optimization*, PAMM, Vol.5(1), pp.743-744, 2005.

[9] Hazra, S. B., Schulz, V., *Simultaneous pseudo-timestepping for PDE-model based optimization problems,* Bit Numerical Mathematics, 44(3): 457-472, 2004.

[10] Hazra, S. B., Schulz, V., Brezillon, J., Gauger, N. R., *Aerodynamic shape optimization using simultaneous pseudo-timestepping*, J. Comp. Phys., 204: 46-64, 2005.

[11] Hazra, S. B., Schulz, V., *Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints,* SIAM J. Sci. Comput, Vol.28, No.3, pp. 1078-1099, 2006.

[12] Hazra, S. B., Schulz, V., *Simultaneous pseudo-timestepping for state constrained optimization problems in aerodynamics,* (to appear) in 'Real-Time PDE-Constrained Optimization' (Eds.: L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes and B. van Bloemen Waanders), SIAM, 2006.

[13] Hazra, S. B., Schulz, V., Brezillon, J.,*Simultaneous pseudo-timestepping for 3D aerodynamic shape optimization,* Forschungsbericht Nr. 05-2, FB IV - Mathematik/Informatik, Universitaet Trier, 2005.

[14] Jameson, A., *Aerodynamic design via control theory*, J. Scientific Computing, Vol.3, pp.233-260, 1988.

[15] Jameson, A., *Automatic design of transonic airfoils to reduce shock induced pressure drag*, In Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics, pages 5-17, Tel Aviv, February, 1990.

[16] Jameson, A., *Optimum aerodynamic design using CFD and control theory*, AIAA 12th Computational Fluid Dynamics Conference, AIAA 95-1729-CP, June, 1995.

[17] Jameson, A., *Optimum aerodynamic design using control theory*, Computational Fluid Dynamics Review, pages 495-528, 1995.

[18] Jameson, A., Pierce, N., Martinelli, L., *Optimum aerodynamic design using the Navier-Stokes equations,* AIAA 97-0101, 35th Aerospace Science Meeting and Exhibit, Reno, Nevada, 1997.

[19] Jameson, A., Martinelli, L., Pierce, N., *Optimum aerodynamic design using the Navier-Stokes equations,* Journal of Theoretical Computational Fluid Dynamics, 10: 213-237, 1998.

[20] Jameson, A., *Efficient aerodynamic shape optimization*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA paper 2004-4369, Albany, New York, Aug.30-Sep.1, 2004.

[21] Nadarajah, S. and Jameson, A.,*Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization*, 15th AIAA computational fluid dynamics conference, June 11-14, Anaheim, CA, 2001.

[22] Reuther, J., Jameson, A., *Aerodynamic shape optimization of wing and wing-body configurations using control theory*, AIAA 95-0123, January, 1995.

[23] Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., *Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation*, AIAA 96-0094, January, 1996.

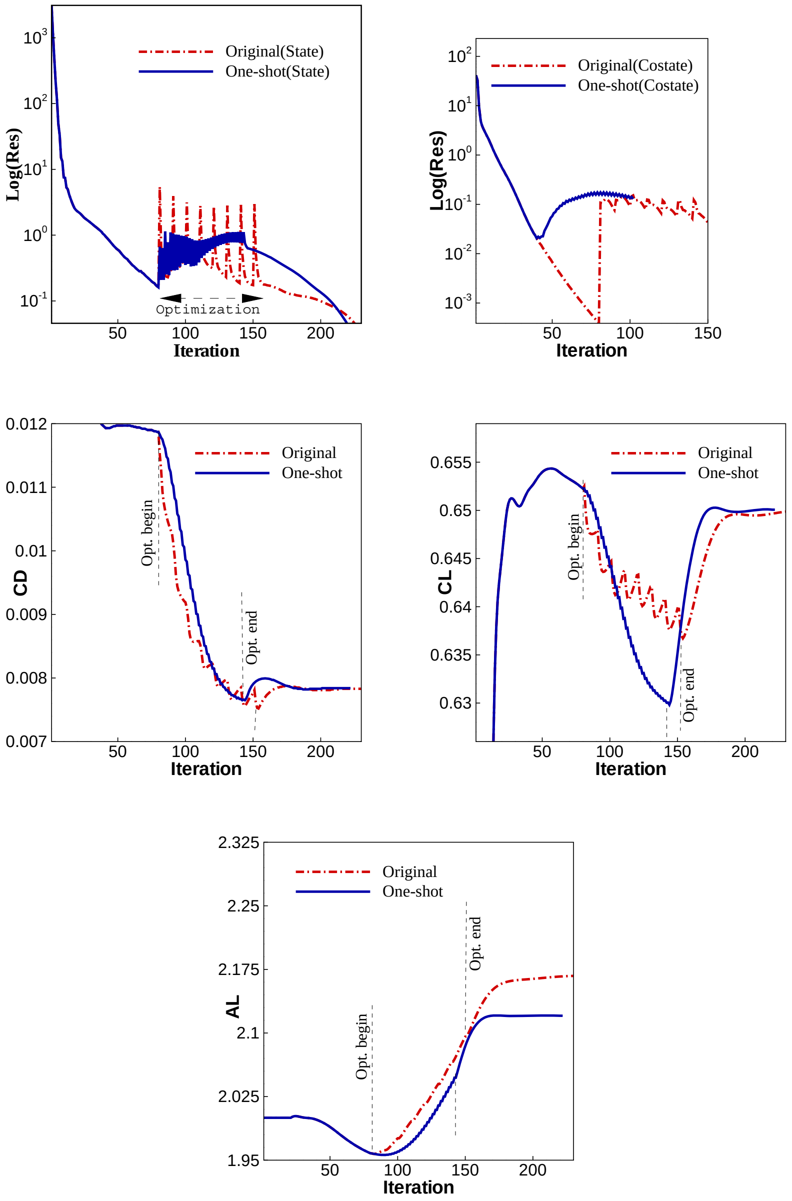
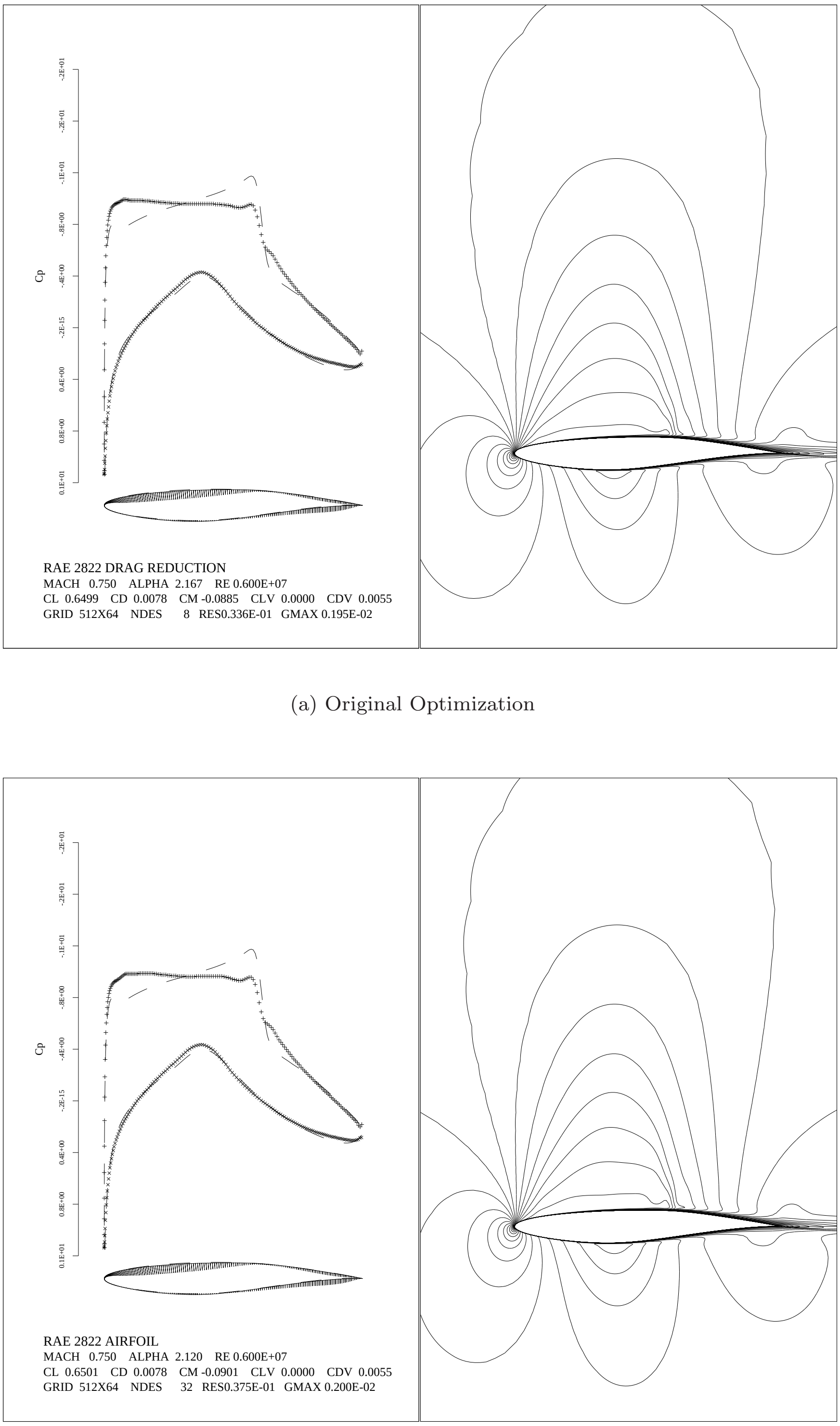


Figure 1. *Convergence history of the optimization iterations (Case 1: RAE2822 airfoil)*



(a) Original Optimization

(b) One-shot optimization

Figure 2. Pressure distribution and Mach contours for the RAE2822 airfoil

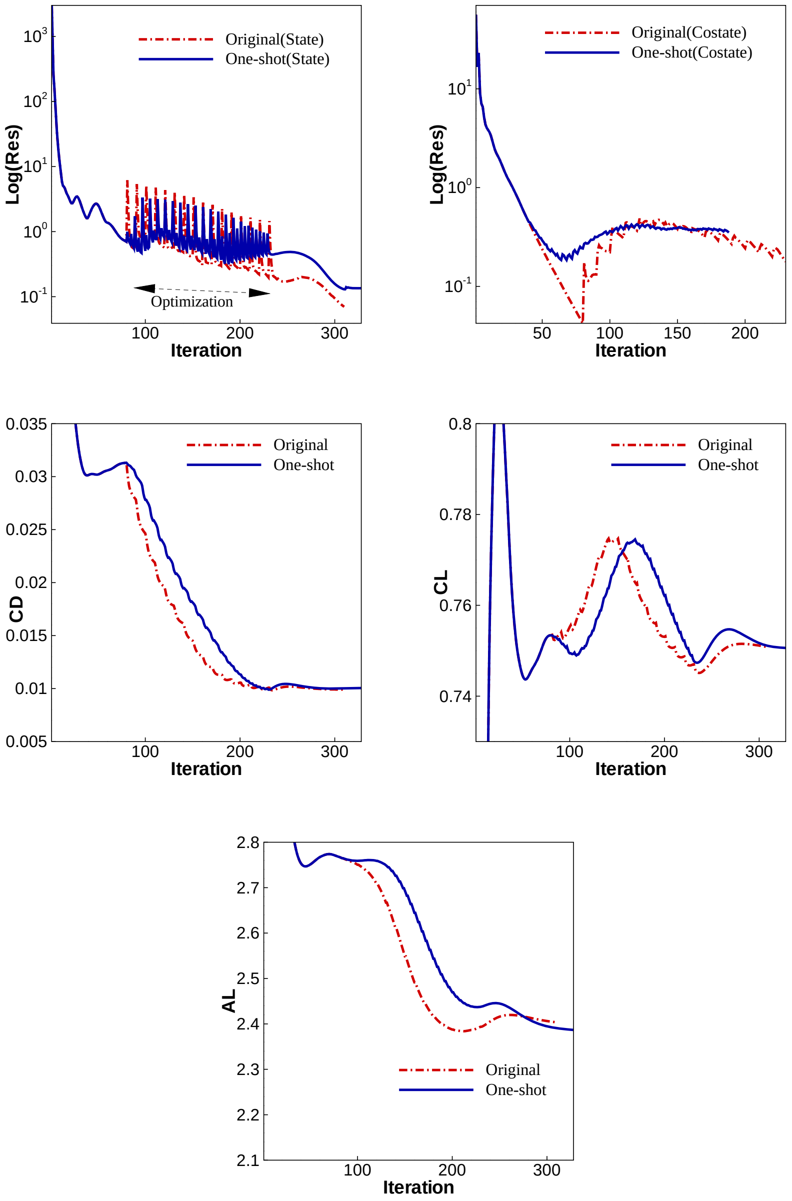
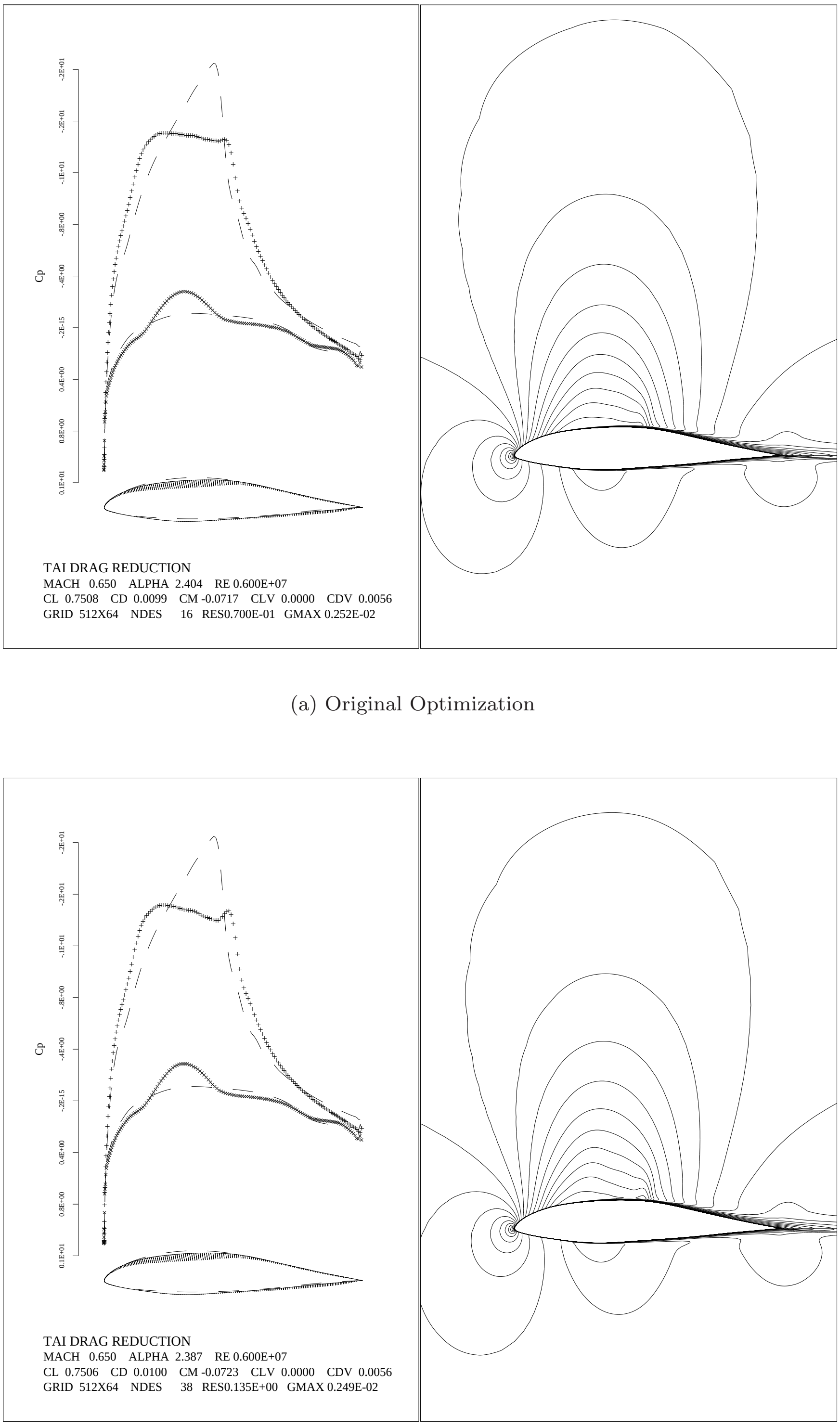


Figure 3. *Convergence history of the optimization iterations (Case 2: TAI airfoil)*



(a) Original Optimization

(b) One-shot optimization

Figure 4. Pressure distribution and Mach contours for the TAI airfoil