AIAA 2001–2532

Towards Efficient Aerodynamic Shape Optimization Based on the Navier–Stokes Equations

M. Nemec and D. W. Zingg
University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

15th AIAA Computational Fluid Dynamics Conference

June 11–14, 2001/Anaheim, CA

Towards Efficient Aerodynamic Shape Optimization Based on the Navier–Stokes Equations

M. Nemec* and D. W. Zingg†
University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, ON, M3H 5T6, Canada

An accurate and efficient algorithm for Navier–Stokes aerodynamic shape optimization problems is presented. The gradient of the objective function is obtained via the discrete adjoint and flow-sensitivity methods. The adjoint and flow-sensitivity equations are solved using a preconditioned GMRES solver. A complete linearization of the discretized Navier–Stokes and turbulence model equations results in an accurate and efficient evaluation of the gradient. Furthermore, this linearization is also used to accelerate the convergence rate of the flow solver, which is based on an inexact-Newton strategy. The gradient is generally obtained in less than one third of the time required to fully converge a flow solution. Factors affecting gradient accuracy are examined. The performance of the new algorithm is demonstrated for several design problems, such as inverse design, lift-constrained drag minimization, lift enhancement and maximization of lift-to-drag ratio.

Introduction

THE accuracy and efficiency of gradient-based algorithms for aerodynamic shape optimization problems are most influenced by the performance of the following two components:

- Solution of the flowfield equations.
- Evaluation of objective function gradient with respect to the design variables.

Among the most promising computational methods for the evaluation of the objective function gradient are the direct, or sensitivity, method and the adjoint method. Both methods are further subdivided into the continuous[1] and the discrete approach.[2] The choice between the direct and the adjoint method is mainly influenced by the details of the design problem. Some considerations are as follows:

1. The computational cost of the gradient calculation for the adjoint method is virtually independent of the number of design variables.
2. Since the direct method determines the sensitivities of the flowfield variables with respect to the design variables, its computational cost is dependent on the number of design variables. However, if the design problem contains constraints which are dependent on the flowfield variables, then the optimization algorithm may require the calculation of the constraint gradients. For the adjoint method, this requires the solution of an additional adjoint equation for each such constraint, while for the direct method, the flow sensitivities can be reused at no additional cost.
3. It may be advantageous to implement both methods since the resulting information can be used to accelerate the convergence of the design problem by constructing better approximations of the Hessian matrix.[3–5]

The adjoint method has recently received considerable attention since it is very appealing for design problems with many design variables. Both the continuous and the discrete adjoint approaches have been applied to the Navier–Stokes design problem.[5–11]

Jameson *et al.*[6] derived the viscous adjoint terms for the continuous approach for laminar and turbulent flows on structured grids. Although this formulation neglects the linearization of laminar and turbulent viscosities, it has been successfully applied to a number of aerodynamic shape optimization problems, including two-dimensional high-lift configurations[12] with the Baldwin–Lomax and the one-equation Spalart–Allmaras turbulence models.[13]

Anderson and Venkatakrishnan[8] analyzed both the continuous and discrete adjoint methods for unstructured grids and implemented the discrete approach for viscous design problems. Anderson and Bonhaus[9] extended this work to turbulent flows by differentiating the Spalart–Allmaras turbulence model 'by hand'. They report accurate gradients for turbulent design

*Graduate Student, marian@oddjob.utias.utoronto.ca.
†Professor, Senior Member AIAA, http://goldfinger.utias.utoronto.ca/~dwz/.

cases. Nielsen and Anderson[14] apply the same strategy to three-dimensional turbulent design problems and also demonstrate excellent gradient accuracy. Furthermore, their results show the influence of various simplifying assumptions in the linearization of the discretized governing equations, such as the assumption of constant turbulent viscosity and a linearization based on first-order discretization. They conclude that most of these simplifying assumptions result in significant gradient errors.

In the adjoint and direct methods, the computational cost of the gradient calculation is dominated by the solution of a large linear system of equations which arises from the flow Jacobian matrix. A popular approach to solve the adjoint and flow-sensitivity equations is to use the same scheme which solves the governing flow equations, for example the explicit and point-implicit multistage Runge–Kutta schemes coupled with multigrid[6,11,15] and the approximate factorization scheme.[10] The preconditioned generalized minimum residual (GMRES) algorithm[16] has been used to solve the discrete sensitivity equation for laminar flows[17] and also to solve the discrete adjoint equation in conjunction with a backward-Euler time-marching scheme with multigrid for turbulent flows.[8,14] Generally, the computational effort required to converge the adjoint equation sufficiently in order to obtain accurate gradients is approximately equivalent to one to two flowfield solutions; however, for the discrete adjoint method this effort may be significantly increased if memory limitations prohibit the storage of the flow Jacobian matrix.[18]

In this work, we present an accurate and efficient algorithm for the calculation of the gradient of the objective function via the discrete adjoint and direct methods. We carefully linearize the two-dimensional Navier–Stokes equations coupled with the Spalart–Allmaras turbulence model. We adopt the approach of Pueyo and Zingg[19] to solve the adjoint and flow-sensitivity equations using the preconditioned GMRES algorithm exclusively. Furthermore, the flow solver is based on a fast Newton–Krylov algorithm,[19] where the linearization of the residual equations is reused. We examine the following issues in detail:

- The accuracy of the gradient calculation using the adjoint and direct methods.
- The efficiency of the gradient calculation and the flow solver.

We investigate the performance of the new design algorithm on several representative design problems, namely inverse design, lift-constrained drag minimization, lift enhancement and maximization of lift-to-drag ratio.

Governing Equations and Numerical Method

The aerodynamic shape optimization problem consists of determining values of design variables (X_D), which control the shape of the airfoil, such that the selected objective function ($\mathcal{J}$) is minimized subject to constraint equations, which include the flow equations and also other airfoil geometry constraints.

Objective Functions and Airfoil Constraints

As objective functions we consider the inverse design, lift-constrained drag minimization, lift enhancement and maximization of lift-to-drag ratio problems. In the inverse design problem the objective function in discrete form is given by

$$\mathcal{J} = \frac{1}{2}\sum_{j=1}^{N_A}(C_{p_j} - C^*_{p_j})^2 \tag{1}$$

where C_p^* represents the target pressure distribution which is user specified and N_A denotes the number of nodes on the airfoil. By minimizing $\mathcal{J}$, the optimizer finds the shape of the airfoil which, in the least-squares sense, best matches the target pressure distribution.

For the remaining design problems the objective function has the following form:

$$\mathcal{J} = \mathcal{J}_a + \mathcal{J}_b \tag{2}$$

For the lift-constrained drag minimization and lift enhancement problems we use

$$\mathcal{J}_a = \omega_{\mathrm{D}}\left(1 - \frac{C_{\mathrm{D}}}{C^*_{\mathrm{D}}}\right)^2 + \omega_{\mathrm{L}}\left(1 - \frac{C_{\mathrm{L}}}{C^*_{\mathrm{L}}}\right)^2 \tag{3}$$

For the maximization of lift-to-drag ratio problem we use

$$\mathcal{J}_a = \frac{C_{\mathrm{D}}}{C_{\mathrm{L}}} \tag{4}$$

In Eq. 3, C^*_{D} and C^*_{L} represent the target drag and lift coefficients, respectively. The weights ω_{D} and ω_{L} are user specified constants.

We augment Eq. 2 with the term $\mathcal{J}_b$ in order to impose airfoil thickness constraints. The thickness constraints are cast as inequality constraints given by the following quadratic penalty term

$$\mathcal{J}_b = \begin{cases} \displaystyle\sum_{i=1}^{N_{\mathrm{TC}}}\omega_{\mathrm{TC}}\left(1 - \frac{t(x_i)}{t^*(x_i)}\right)^2 & \text{if } t(x_i) < t^*(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where N_{TC} represents the number of thickness constraints, $t^*(x)$ represents the minimum allowable airfoil thickness at location x and ω_{TC} is a user specified constant. We can also treat the thickness constraints as equality or mixed constraints. The present formulation works well for the design cases presented here,

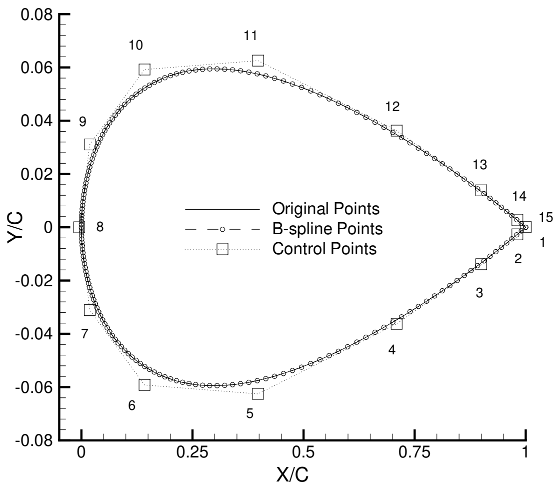


Fig. 1 B-spline curve and control points for the NACA 0012 airfoil

but it should be noted that there are well-known weaknesses of penalty functions[20] and more sophisticated strategies for solving constrained problems are given in Refs. 21, 3, and 22.

Design Variables

The geometry of the airfoil is described with B-spline curves.[23,24] The coordinates of the B-spline control points are used as design variables. An example is shown in Fig. 1, where cubic B-splines constructed from 15 control points are used to approximate the NACA 0012 airfoil. By increasing the number of control points, the accuracy and fidelity of the B-spline curve is improved. Numerical experiments performed on a number of airfoil shapes suggest that 25 control points are sufficient to reach manufacturing tolerances. For the drag minimization problem at fixed lift, the angle of attack (α) becomes a design variable as well. In this study, we only allow displacements in the normal direction for the B-spline control points.

Flow Solver

The governing flow equations are the two-dimensional thin-layer Navier–Stokes equations in generalized coordinates:

$$\frac{\partial \widehat{E}(\widehat{Q}, X_D)}{\partial \xi} + \frac{\partial \widehat{F}(\widehat{Q}, X_D)}{\partial \eta} = \mathcal{R}e^{-1} \frac{\partial \widehat{S}(\widehat{Q}, X_D)}{\partial \eta} \tag{6}$$

where $\widehat{Q} = J^{-1}Q = J^{-1}[\rho, \rho u, \rho v, e]^T$ is the vector of conservative dependent state variables, ξ and η are the streamwise and normal generalized coordinates, respectively, and J is the Jacobian of the coordinate transformation from Cartesian coordinates. Vectors $\widehat{E}$ and $\widehat{F}$ represent the convective flux vectors and the viscous flux vector is given by $\widehat{S}$. Sutherland's law is used to determine the laminar viscosity. The equations are in non-dimensional form. For further details, see Ref. 25.

The turbulent viscosity is modeled with the Spalart–Allmaras turbulence model.[13] All test cases considered in this study are assumed to be fully turbulent, and therefore, the laminar-turbulent trip terms are not used.

The spatial discretization for Eq. 6 is the same as that used in ARC2D.[25] The discretization consists of second-order centered-difference operators with second- and fourth-difference artificial dissipation. A far-field circulation correction is also implemented. The Spalart–Allmaras turbulence model is discretized as described in Ref. 13. Overall, the spatial discretization leads to a nonlinear system of equations

$$R(\widehat{Q}, X_D) = 0 \tag{7}$$

where $\widehat{Q} = J^{-1}Q = J^{-1}[\rho, \rho u, \rho v, e, \tilde{\nu}]^T$ is the new vector of conservative dependent state variables and the turbulence model equation is scaled by J^{-1}.

Eq. 7 is solved in a fully-coupled manner, where convergence to steady state is achieved using the preconditioned GMRES algorithm in conjunction with an inexact Newton strategy.[19] The algorithm can be summarized as follows:

- Matrix-free GMRES(40)
- Preconditioner is based on the first-order Jacobian matrix and block-fill incomplete lower/upper decomposition with fill-in of 2 (BFILU(2)). Right preconditioning is used.
- Reverse Cuthill–McKee reordering of unknowns based on initial double bandwidth ordering.[19]

The first-order Jacobian matrix is formed from the exact Jacobian matrix ($\partial R/\partial \widehat{Q}$) by combining the second- and fourth-difference artificial dissipation coefficients on the right-hand side as follows:

$$\epsilon_2^l = \epsilon_2^r + \sigma \epsilon_4^r \tag{8}$$

where $\sigma = 6.0$. Note that this modification does not affect the steady-state solution. In order to obtain fast and robust GMRES convergence, the preconditioner is frozen after the first Newton iteration and the GMRES convergence tolerance is set to 0.5 for the first 10 Newton iterations and 0.1 for any remaining Newton iterations. Furthermore, the turbulence model is fully linearized and included in the preconditioner. This linearization is later reused in the adjoint and sensitivity methods, as discussed in the following sections.

To avoid Newton startup problems, the approximate-factorization algorithm of ARC2D in diagonal form in conjunction with a subiteration scheme for the turbulence model equation is used to reduce the initial residual by two orders of magnitude. For further details, see Refs. 25, 13, and 19.

Adjoint and Flow-Sensitivity Solvers

The gradient of the objective function $\mathcal{J}(Q, X_D)$ is given by

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} + \frac{\partial \mathcal{J}}{\partial Q}\frac{\mathrm{d}Q}{\mathrm{d}X_D} \tag{9}$$

In Eq. 9 we reduce the vector of design variables (X_D) to a scalar in order to clearly distinguish between partial and total derivatives.

The difficulty in Eq. 9 is the evaluation of the term $\mathrm{d}Q/\mathrm{d}X_D$, referred to as the flow sensitivities. In order to compute the flow sensitivities, differentiate Eq. 7 with respect to the design variables

$$\frac{\mathrm{d}R}{\mathrm{d}X_D} = \frac{\partial R}{\partial X_D} + \frac{\partial R}{\partial \widehat{Q}}\frac{\mathrm{d}\widehat{Q}}{\mathrm{d}X_D} \tag{10}$$

and realize that $\frac{\mathrm{d}R}{\mathrm{d}X_D} = 0$, since for any design variable Eq. 7 is always satisfied. Furthermore, note that $\partial\widehat{Q}/\partial Q = J^{-1}\mathrm{I}$, and consequently Eq. 10 becomes

$$\frac{\partial R}{\partial Q}\frac{\mathrm{d}Q}{\mathrm{d}X_D} = -\frac{\partial R}{\partial X_D} \tag{11}$$

The direct, or sensitivity, method results from solving Eq. 11 for the flow sensitivities $\mathrm{d}Q/\mathrm{d}X_D$ and using these values in Eq. 9 to obtain the gradient. In order to formulate the discrete adjoint method, substitute Eq. 11 into Eq. 9 to obtain

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} - \frac{\partial \mathcal{J}}{\partial Q}\left(\frac{\partial R}{\partial Q}\right)^{-1}\frac{\partial R}{\partial X_D} \tag{12}$$

From the triple-product term in Eq. 12, define the following intermediate problem

$$\frac{\partial R}{\partial Q}^{\mathrm{T}}\psi = \frac{\partial \mathcal{J}}{\partial Q}^{\mathrm{T}} \tag{13}$$

This is known as the adjoint equation where the vector ψ represents the adjoint variables. Substituting the value of ψ into Eq. 12, the expression for the gradient becomes

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} - \psi^{\mathrm{T}}\frac{\partial R}{\partial X_D} \tag{14}$$

Note that Eq. 11 must be solved for each design variable, while Eq. 13 is independent of the design variables. If a direct solver is used to solve Eq. 11, then the LU factorization can be reused with different right-hand-side vectors. Unfortunately, direct solvers are presently only suitable for small two-dimensional problems and the use of iterative solvers leads to re-solving Eq. 11 for each design variable, which is computationally expensive.

We adopt the GMRES strategy from the flow solver to solve both the adjoint and flow-sensitivity equations. We use right preconditioning and the preconditioner is based on the first-order Jacobian matrix as shown in Eq. 8. Reverse Cuthill–McKee reordering of the unknowns is used to improve the performance of the preconditioner. Fast adjoint and flow-sensitivity solutions are obtained with $\sigma = 3.0$, BFILU(6) and GMRES(85), but these settings are conservative and are further discussed in the results and discussion section. For the flow-sensitivity equation, we use matrix-free GMRES with second-order accurate Fréchet derivatives. In addition to memory savings, the matrix-free approach is easier to implement since an accurate linearization of cumbersome functions in the residual equations, such as the pressure switch used for shock capturing and the far-field circulation correction, is 'automatically' provided. Due to the transpose on the left-hand-side of Eq. 13, the matrix-free approach is not possible for the adjoint equation.

For the inverse design objective function the term $\partial\mathcal{J}/\partial Q$ is evaluated analytically, while for the drag minimization objective function it is evaluated using centered differences. The remaining terms in Eq. 14, namely the objective function sensitivity $\partial\mathcal{J}/\partial X_D$ and the residual sensitivity $\partial R/\partial X_D$, are also evaluated using centered differences.

Optimizer

We use an unconstrained BFGS quasi-Newton algorithm[26] with a backtracking line search. At each step of the line search, the objective function value and the gradient value are required in order to construct a local cubic interpolant (for further details, see Ref. 26).

Grid Movement Strategy

C-topology structured grids are used. As the shape of the airfoil evolves during the optimization process, the position of the grid nodes is adjusted to conform to the new shape. The grid movement strategy is summarized below for grid lines in the normal direction. This strategy is similar to one of the strategies outlined in Ref. 27. An analogous formulation holds for the streamwise direction.

Given a displacement of the B-spline control point in the normal direction, the grid movement strategy should preserve the off-wall spacing and the distance to the outer boundary. The interior nodes along a normal grid line are positioned as determined by

$$y_k^{new} = y_k^{old} + \Delta y_1 \left(1 - S_k\right) \tag{15}$$

where Δy_1 represent the airfoil shape change. S_k is the normalized arclength distance given by

$$S_k = \frac{\sum_{i=2}^{k} L_i}{\sum_{i=2}^{kmax} L_i} \tag{16}$$

where L_i is the length of each segment between nodes.

Results and Discussion

The CPU times reported in the following sections are obtained on a 667 MHz Alpha 21264 processor.

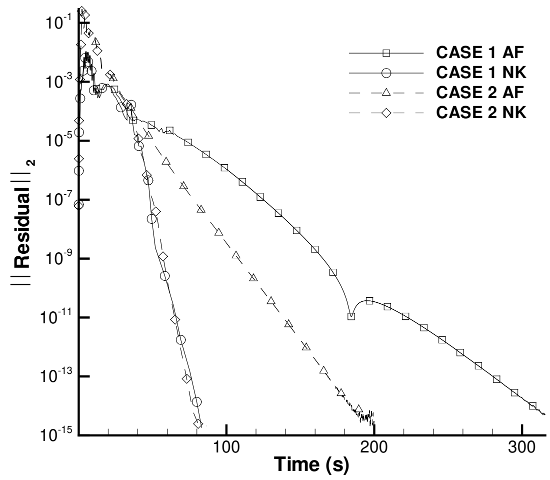


Fig. 2 Performance of the Newton–Krylov algorithm

All subsonic cases are computed on a 265×53 grid, while for transonic cases, a 257×57 grid is used. For all grids, the distance to the outer boundary is 24 chords, the off-wall spacing is 2×10^{-6} chords, the leading-edge clustering is 5×10^{-4} chords and the trailing-edge clustering is 2×10^{-3} chords. The circulation correction is not used unless explicitly stated.

Flow Solver Performance

We evaluated the performance of the Newton–Krylov algorithm on a number of test cases. The following two test cases are good examples:

1. NACA 0012 airfoil at $M_\infty = 0.3$, $\alpha = 6°$ and $\mathcal{R}e = 2.88 \times 10^6$.

2. RAE 2822 airfoil at $M_\infty = 0.729$, $\alpha = 2.31°$ and $\mathcal{R}e = 6.5 \times 10^6$.

Both cases are fully turbulent.

Fig. 2 shows that the Newton–Krylov algorithm (denoted as NK) is approximately two to three times faster than the approximate-factorization algorithm (denoted as AF). For some cases, this speed-up can be even larger. As shown in the figure, the Newton–Krylov algorithm tends to maintain approximately constant performance, regardless of minor variations in the grid and the freestream conditions.

Gradient Accuracy

The factor which influences the accuracy of the gradient calculation most significantly is the linearization of the discretized residual equations, Eq. 7, to obtain the flow Jacobian matrix $(\partial R/\partial \widehat{Q})$. For the adjoint method, we carefully linearize the residual equations 'by hand', including all terms in the Spalart–Allmaras turbulence model. However, exact linearization is complicated by the use of non-differentiable functions such as the max function and the absolute value function. These functions are used in the pressure switch required for the artificial dissipation scheme. The absolute value function is also used in the calculation of the spectral radius, which is again required by the artificial dissipation scheme. Furthermore, the absolute value function is required in the calculation of vorticity and in the first-order upwind discretization of the advective terms in the Spalart–Allmaras turbulence model.

An additional complication is the linearization of the far-field circulation correction since the calculation of the vortex strength leads to coupling between airfoil surface points and the far-field boundary points. The vortex strength calculation can be linearized as described by Korivi *et al.*,[10] but in the present linearization of the residual equations for the adjoint method we treat the vortex strength and the pressure switch used for shock capturing as a constant. We linearize the spectral radius of the artificial dissipation scheme, the calculation of vorticity and the advective terms of the Spalart–Allmaras turbulence model. Note that the analytical linearization of the absolute value function is not valid when the function argument changes sign. The matrix-free implementation of the flow-sensitivity method avoids the above mentioned linearization difficulties.

We examine the accuracy and efficiency of the gradient calculation for two representative test cases:

1. Subsonic inverse design.

2. Transonic drag minimization at fixed lift.

For case 1, the freestream conditions are $M_\infty = 0.3$, $\alpha = 6°$ and $\mathcal{R}e = 2.88 \times 10^6$. The NACA 0012 pressure distribution at the given freestream conditions is used as the target pressure distribution. The initial pressure distribution is obtained by replacing the NACA 0012 leading edge with the RAE 2822 leading edge, which modifies the location of B-spline control points numbered 6, 7, 9 and 10 in Fig. 1.

The gradient of the inverse design objective function (Eq. 1) with respect to the design variables associated with the four control points is calculated using centered differences, the adjoint method and the matrix-free flow-sensitivity method. The finite-difference stepsize is 1×10^{-5} and we converge the flow solution 14 orders of magnitude. The adjoint and flow-sensitivity equations are converged eight orders of magnitude. The calculated gradient values are shown in Table 1, where the agreement between the finite-difference, the adjoint, and the matrix-free flow-sensitivity gradients, labeled as S-MF, is very good.

For case 2, the freestream conditions are $M_\infty = 0.7$, $C_l = 0.4728$ and $\mathcal{R}e = 9 \times 10^6$. The initial airfoil geometry is the NACA 0012 airfoil. We compute the

Control Point #	Finite Difference	Adjoint (% Diff.)	S-MF (% Diff.)
6	-127.82	0	-0.008
7	618.91	0	0
9	-2093.8	0	0
10	-526.58	-0.002	0

Table 1 Gradient accuracy for case 1

Control Point #	Finite Difference	Adjoint (% Diff.)	S-MF (% Diff.)
9	22.465	-7.0	0.004
10	29.592	4.1	-0.01
11	-16.428	-6.2	0.006
12	2.4066	-8.9	0.47
α	0.57486	0.52	0.06

Table 2 Gradient accuracy for case 2

gradient of the objective function, Eq. 3, with respect to control points 9, 10, 11 and 12 (see Fig. 1), as well as the angle of attack (α). The target drag value, C_{D}^{*}, is set equal to 0.0112, which represents a 30% reduction from the initial drag value. No thickness constraints are imposed and the values of ω_{L} and ω_{D} in Eq. 3 are set to 2.0 and 1.0, respectively. Table 2 shows that there is a small error between the finite-difference gradients and the adjoint gradients, which is due to the treatment of the pressure switch. The agreement between the matrix-free flow-sensitivity gradients and the finite-difference gradients is not quite as good as for case 1, but remains excellent.

In order to demonstrate that the differences in Table 2 are due to the treatment of the pressure switch, we perform the following numerical experiment. We first converge the flow solver and store the pressure switch values, and then we reuse these values when we compute the two neighboring states in the centered-difference gradient calculation. Hence, during the finite-difference gradient calculation the pressure switch is treated as a constant, which is consistent with the linearization of the residual equations. The results are summarized in Table 3, where the values obtained from the adjoint method agree well with the finite-difference and matrix-free flow sensitivity values. The minor differences which appear in Table 3 are due to the linearization of the absolute value function.

The effect of the far-field circulation correction on the gradient accuracy is shown in Table 4. We perform a similar numerical experiment to that above; however, we freeze both the pressure switch and all absolute value functions such that we completely isolate the error contribution from the far-field circulation correction in the adjoint equation. The agreement between finite-differences and matrix-free flow sensitivities is very good and the error in the adjoint gradients is small. These results suggest that treating the vortex strength as a constant has a relatively small effect on gradient accuracy.

Control Point #	Finite Difference	Adjoint (% Diff.)	S-MF (% Diff.)
9	20.895	-0.03	-0.005
10	30.803	-0.06	-0.02
11	-15.416	-0.03	0.04
12	2.1940	-0.01	-0.02
α	0.57798	-0.02	0.03

Table 3 Gradient accuracy for case 2 with frozen pressure switch

Control Point #	Finite Difference	Adjoint (% Diff.)	S-MF (% Diff.)
9	17.954	-0.1	0
10	27.992	-0.1	0
11	-12.370	-0.7	-0.008
12	2.7021	1.7	0
α	0.54509	0.3	0

Table 4 Gradient accuracy for case 2 with circulation correction (frozen pressure switch and all absolute value functions)

Efficiency of Gradient Calculation

Figs. 3 and 4 show the convergence history of the adjoint equation and the flow-sensitivity equation for the inverse design problem, labeled as case 1, and the drag minimization problem, labeled as case 2A, both introduced in the previous section. The GMRES solver parameters are $\sigma = 3.0$, BFILU(6) and GMRES(85). The adjoint equation for the inverse design problem takes more iterations to converge than the drag minimization problem. This is due to the increase in the number of search directions required before a reduction in the residual is observed, and furthermore, GMRES is forced to perform a restart on iteration 86 as shown in Fig. 3. Due to the nature of the convergence characteristics, GMRES restarts should be avoided in order to obtain fast performance by selecting a sufficient number of search directions such that the adjoint residual is reduced by three orders of magnitude. The flow-sensitivity equation converges well for both the inverse design problem and the drag minimization problem as shown in Fig. 4.

The values of the fill level in the BFILU decomposition and the number of search directions can be significantly reduced for the drag minimization objective function. For example, consider case 2B shown in Figs. 3 and 4, which is the drag minimization problem evaluated with BFILU(4) and GMRES(60). There is only a minimal reduction in performance, while the memory savings are significant.

The efficiency of the adjoint solver for cases 1, 2A, and 2B is summarized in Table 5. In Table 5, '$\mathrm{T_{PREC}}$'

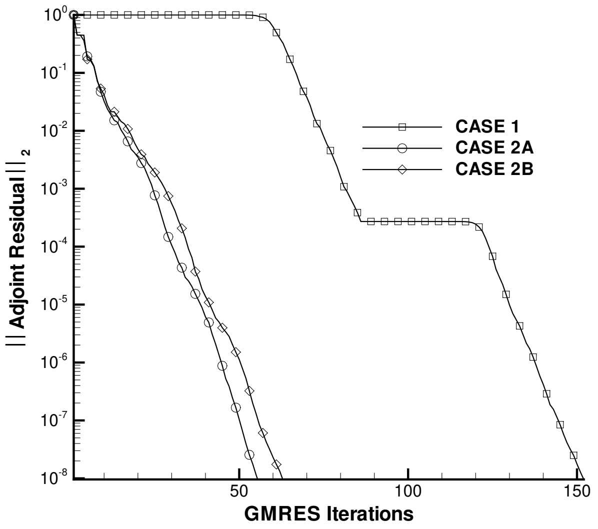


Fig. 3 GMRES convergence for the adjoint equation

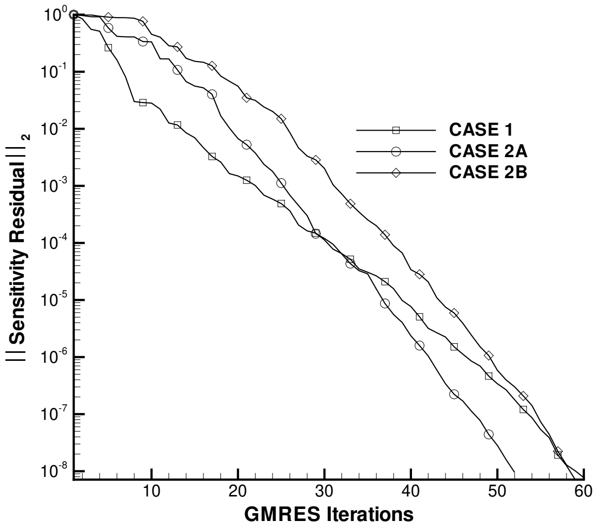


Fig. 4 GMRES convergence for the sensitivity equation (first design variable shown)

refers to time required to form the preconditioner, '$\mathrm{T_{GMRES}}$' refers to time required to reduce the adjoint residual by three orders of magnitude and 'Flow Solve' refers to the time required for a Newton-Krylov flow solve to converge 14 orders of magnitude. A three order-of-magnitude reduction in the residual of the adjoint equation is required to obtain accurate gradients.[18,28] Hence, for case 1 the time to calculate the gradient is 1/3 of the flow solve time, while for case 2 it is less than 1/7 of the flow solve time. Case 2B, which uses BFILU(4) preconditioner, is even faster, since the time to form and apply the preconditioner has been reduced.

The efficiency of the matrix-free flow-sensitivity solver is summarized in Table 6, where '$\mathrm{T_{GMRES}}$' refers to the time *per design variable* required to reduce the

Case	$\mathrm{T_{PREC}}$	$\mathrm{T_{GMRES}}$	Total	Flow Solve
1	4.2	22.9	27.1	83.2
2A	4.2	7.0	11.2	75.7
2B	2.6	6.6	9.2	75.7

Table 5 Efficiency of adjoint solver (CPU time measured in seconds)

Case	$\mathrm{T_{GMRES}}$ (per D. V.)	Total (for all D. V.)
1	5.8	27.4
2A	6.1	34.7
2B	6.5	35.1

Table 6 Efficiency of matrix-free sensitivity solver (CPU time measured in seconds)

residual of the flow-sensitivity equation by three orders of magnitude and 'Total' is the total time required to calculate the gradient given by $\mathrm{T_{PREC}}+N_{\mathrm{DV}}\cdot\mathrm{T_{GMRES}}$, where N_{DV} represents the number of design variables. The time to form the preconditioner and the flow solve time are shown in Table 5. For the present number of design variables, the flow-sensitivity gradients require approximately half of the flow solve time.

In terms of memory, the gradient calculation on the present grids requires approximately 275 Mbytes of storage for the adjoint method, which may be prohibitive for large three-dimensional problems, but is certainly acceptable for two-dimensional calculations. Since the matrix-free flow-sensitivity method avoids the storage of the flow Jacobian matrix, its memory requirements are approximately 20% less than the adjoint method.

Design Examples

Having demonstrated the accuracy and efficiency of the gradient calculation, we now solve the aerodynamic shape optimization problem for cases 1 and 2. Since both cases are cast as an unconstrained problem for the BFGS algorithm, the gradient of the objective function should vanish at the optimum solution.

Figs. 5 and 6 show the adjoint optimization results for case 1, which is the inverse design problem with four design variables. The optimizer finds the target pressure distribution in 23 flow solves and gradient evaluations. In terms of CPU time, 33.8 minutes are required for the 23 design iterations of which 10.4 minutes are used for gradient evaluations and 23.4 minutes for flowfield solutions. Note that only the first flow solve is converged from freestream conditions, the rest are 'warm started' from the previous solution. The optimization is converged to plotting accuracy within 11 design iterations or approximately 16 minutes.

Case 2 is the drag minimization problem at fixed lift with five design variables, namely B-spline control points numbered 9, 10, 11 and 12, and the angle of

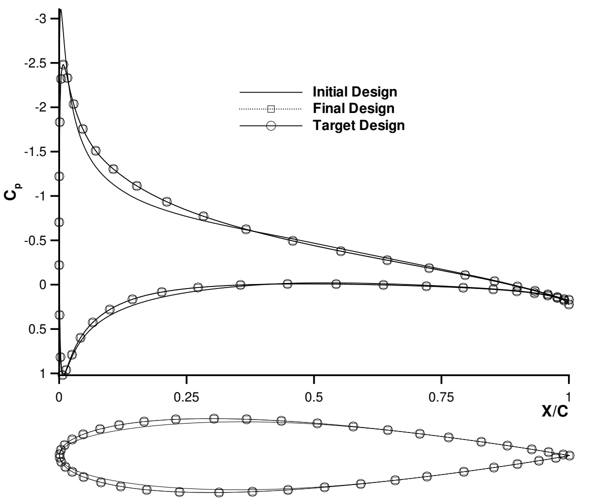


Fig. 5 Case 1 pressure distribution and airfoil shape summary

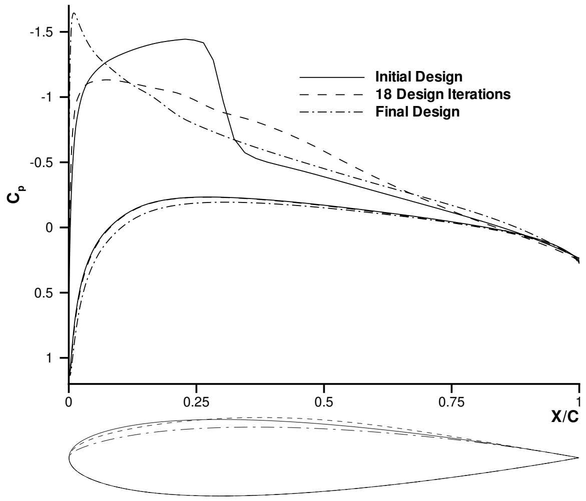


Fig. 7 Pressure distribution and airfoil shape summary for case 2 obtained with the adjoint method

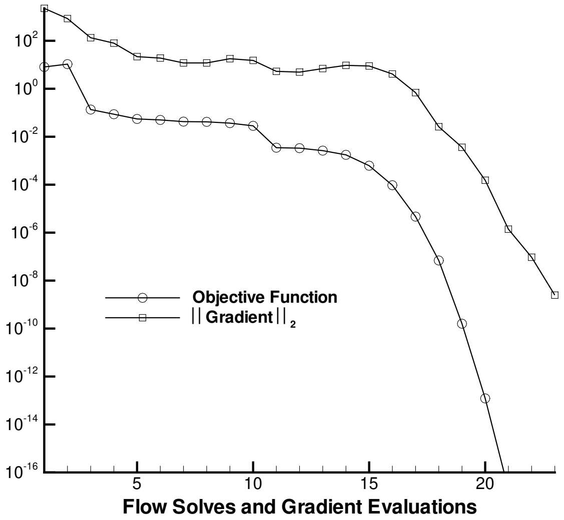


Fig. 6 Case 1 optimization convergence history

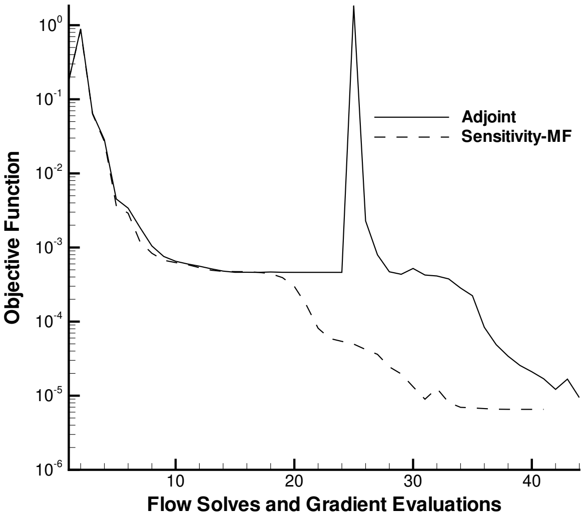


Fig. 8 Comparison of the adjoint method and matrix-free sensitivity method for case 2

attack. The goal of the optimization is to reduce C_D to 0.0112 which represents a 30% reduction from the initial drag value while holding C_L at 0.4728. Note that this problem does not necessarily have a unique solution. We solve this problem with the adjoint and matrix-free flow-sensitivity method in order to examine the impact of gradient accuracy on the optimization problem.

Fig. 7 indicates that after 18 design iterations the upper surface shock is eliminated. We only plot the adjoint results since results obtained with the matrix-free flow-sensitivity method are very similar. The values of C_L, C_D and α at this stage are 0.4713, 0.01144 and 2.83, respectively, for the adjoint method and 0.4724, 0.01144, and 2.84, respectively, for the sensitivity method. So far, the adjoint method does not appear to be significantly affected by the errors in its gradient calculation. Unfortunately, the convergence of the objective function stalls during the subsequent search direction for the adjoint method, as shown in Fig. 8. The BFGS algorithm is restarted from the steepest descent direction and the adjoint method catches up to the matrix-free flow-sensitivity method after 40 design iterations. The final values of C_L, C_D and α are 0.4727, 0.01123, and 3.33, respectively, for the adjoint method and 0.4726, 0.01123, and 3.28, respectively, for the matrix-free flow-sensitivity method. The L_2 norm of the gradient is reduced by 3.5 orders of magnitude for the adjoint method and 4 orders of magnitude for the matrix-free flow-sensitivity method. The final pressure distribution and the airfoil shape is shown in Fig. 7.

Three additional design examples are provided in or-

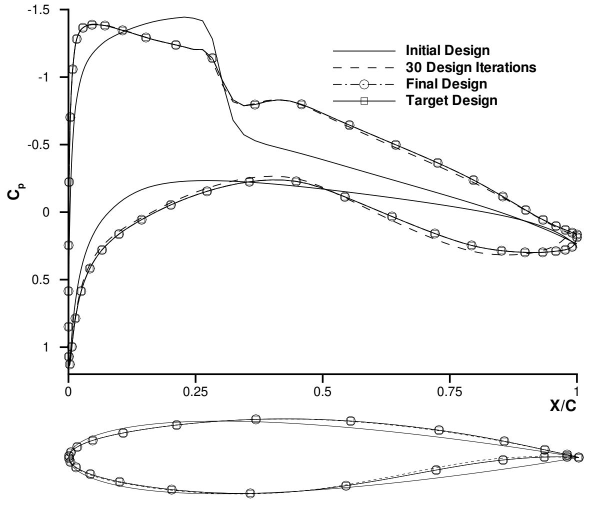


Fig. 9 Pressure distribution and airfoil shape summary for the inverse design problem (12 design variables)

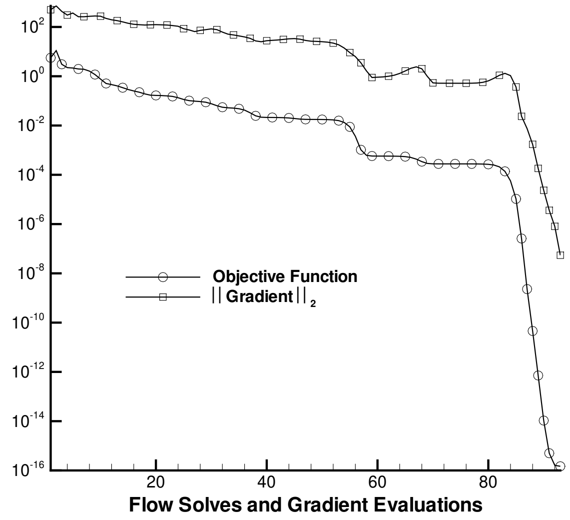


Fig. 10 Optimization convergence history for the inverse design problem (12 design variables)

der to demonstrate the accuracy and efficiency of the present aerodynamic shape optimization algorithm for more complex design problems. The adjoint method is used for all examples. The first example is an inverse design at transonic speed. The initial pressure distribution corresponds to the NACA 0012 airfoil and the target pressure distribution corresponds to the RAE 2822 airfoil at $M_\infty = 0.7$, $\alpha = 3°$ and $\mathcal{R}e = 9 \times 10^6$. The airfoil shape is described with 15 B-spline control points of which 12 are used as design variables. The control point at the leading edge and the two control points at the trailing edge (points labeled 1, 8, and 15 in Fig. 1) are kept constant during the optimization.

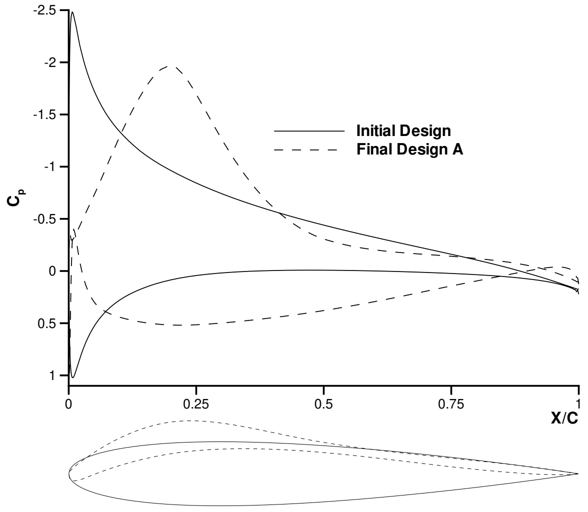
Fig. 9 shows the initial pressure distribution corresponding to the NACA 0012 airfoil, the target pressure distribution corresponding to the RAE 2822 airfoil, and the final design pressure distribution, as well as the corresponding airfoil shapes. Also shown in the figure is the pressure distribution and airfoil shape after 30 design iterations which is very close to the target. The optimization history is summarized in Fig. 10. Note that about 90 flow solves and gradient evaluations are required to reduce the L_2 norm of the gradient by 10 orders of magnitude, although plotting accuracy is achieved within 60 design iterations. In terms of CPU time, plotting accuracy is achieved in approximately 1.5 hours.

The second optimization example involves the design problem of attaining specified lift while holding drag constant. The initial airfoil is the NACA 0012 and the freestream conditions are $M_\infty = 0.3$, $\alpha = 6°$ and $\mathcal{R}e = 2.88 \times 10^6$. The corresponding lift and drag coefficients are 0.6694 and 0.01493, respectively. For the optimization problem we specify a target lift coefficient of 0.9 and a target drag coefficient the same as the initial drag coefficient. The initial airfoil shape is described with 19 B-spline control points and we use 14 control points as design variables, as well as the angle of attack. The B-spline control point at the leading edge and the four B-spline control points at the trailing edge are kept constant during the optimization. The values of ω_{L} and ω_{D} in Eq. 3 are set to 2.0 and 1.0, respectively.

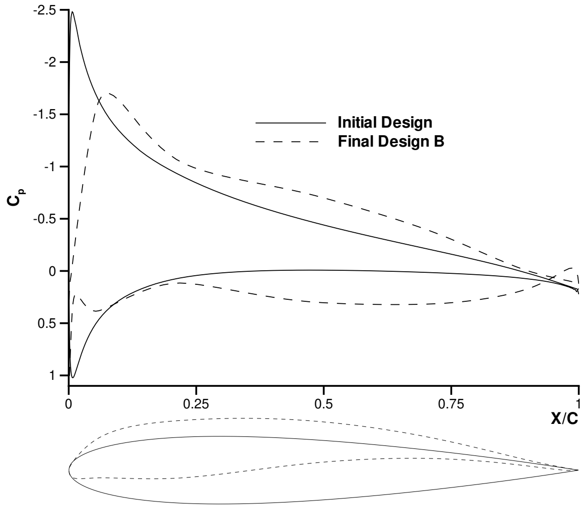
Fig. 11(a) shows the initial pressure distribution, the final pressure distribution (labeled as design A) and the corresponding airfoil shapes for the optimization problem with no active thickness constraints on the final design. In order to prevent the lower and upper airfoil surfaces from crossing during the optimization process, it was necessary to impose 0.5% thickness constraints at 60% and 70% chord. For the final design, the angle of attack is reduced to 4.33°. Fig. 12 shows that the optimization converges in 25 design iterations. The CPU time for the optimization is just under 30 minutes.

In order to obtain a more practical design, we specify four thickness constraints. The value of ω_{TC} in Eq. 5 is set to 1.0. Note that for truly practical designs, multi-point optimization must be used.[28, 29] Fig. 11(b) shows the final pressure distribution (labeled as design B) and the corresponding airfoil shape. Fig. 12 shows that the optimization required 90 design iterations to converge, which is significantly more than the unconstrained case. Since the objective function value is reduced by 15 orders of magnitude, all of the thickness constraints are satisfied. The angle of attack is reduced from 6.0° to 3.56°.

The third and final design problem we consider is the maximization of the lift-to-drag ratio. The initial airfoil is the NACA 0012, and the freestream conditions are $M_\infty = 0.25$, $\alpha = 9°$ and $\mathcal{R}e = 2.88 \times 10^6$. The airfoil shape is described with 15 B-spline control points



a) No thickness constraints



b) With thickness constraints

Fig. 11 Pressure distribution and airfoil shape summary for the lift enhancement problem with and without thickness constraints (15 design variables)

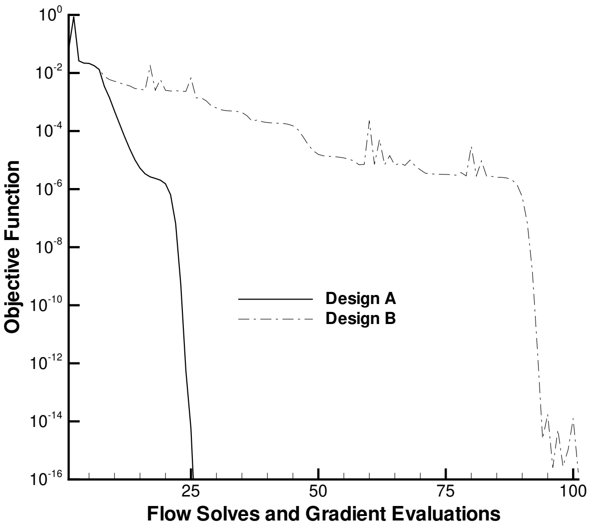


Fig. 12 Objective function convergence history for the lift enhancement problem with and without thickness constraints (15 design variables)

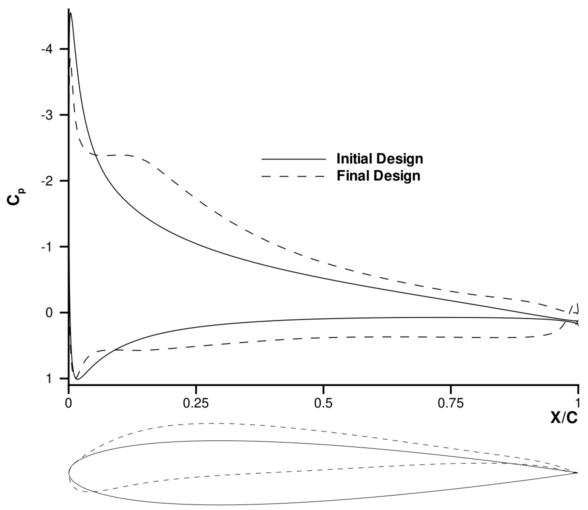


Fig. 13 Pressure distribution and airfoil shape summary for the maximization of C_L/C_D with thickness constraints (10 design variables)

of which 10 are used as design variables. The angle of attack is fixed during the optimization. This case is similar to one of the cases considered by Liebeck.[30] Five thickness constraints are specified, with the minimum allowable thickness at 25% chord equal to 10%. The value of ω_{TC} in Eq. 5 is set to 0.05.

Fig. 13 shows the initial pressure distribution corresponding to the NACA 0012 airfoil, the final design pressure distribution and the corresponding airfoil shapes. The lift coefficient is increased from 0.967 to 1.49, while the drag coefficient increases from 0.01969 to 0.02443. Correspondingly, the lift-to-drag ratio increases from 49.1 to 61.1, and the L_2 norm of the gradient is reduced by 5 orders of magnitude as shown in Fig. 14. Three of the five thickness constraints are active. The largest thickness violation occurs at 25% chord where the final thickness is 9.85%. The coefficient of skin friction indicates that the flow is separated over the last 1.5% of chord.

Concluding Remarks

We have examined the accuracy of the gradient calculation for the discrete adjoint and matrix-free flow-sensitivity methods. The agreement between the matrix-free flow-sensitivity gradient and the finite-difference gradient is excellent for all design problems studied. For subsonic design problems, this is also

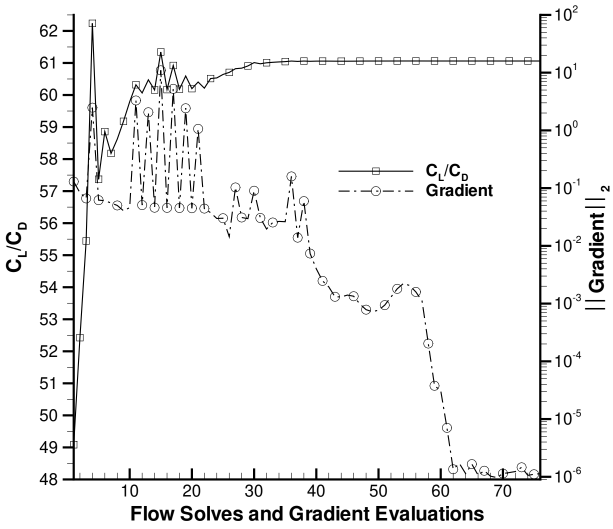


Fig. 14 Optimization convergence history for the maximization of C_L/C_D with thickness constraints (10 design variables)

the case for the adjoint method. For transonic design problems with the adjoint method, the error in the gradient calculation is less than 10% when compared with finite-difference gradients. This error is primarily caused by treating the pressure switch used for shock capturing and the vortex strength associated with the far-field circulation correction as constants during the linearization of the discrete residual equations. We have demonstrated that these small errors in the gradient do not significantly affect the final design.

Furthermore, careful linearization of the residual equations provides a good basis for the GMRES preconditioner, which not only results in fast convergence of the adjoint and flow-sensitivity equations, but also in a fast convergence of the flowfield equations. Depending on the objective function, the adjoint gradient is obtained in 1/3 to 1/7 of the time required for a flowfield solution. We have demonstrated the accuracy and efficiency of the new aerodynamic shape optimization algorithm on four representative design problems.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada, as well as Bombardier Aerospace and an OGS grant from the Government of Ontario. The authors gratefully acknowledge Todd Chisholm for helpful discussions and suggestions.

References

[1] Jameson, A., "Aerodynamic Design via Control Theory," Institute for Computer Applications in Science and Engineering NASA CR-181749, 1988.

[2] Baysal, O. and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *J. Fluids Engineering*, Vol. 113, No. 4, 1991, pp. 681–688.

[3] Drela, M., "Design and Optimization Method for Multi-Element Airfoils," AIAA Paper 93–0969, 1993.

[4] Jou, W., Huffman, W., and Young, D., "Practical Considerations in Aerodynamic Design Optimization," AIAA Paper 95–1730, 1995.

[5] Sherman, L. L., Taylor III, A. C., Green, L. L., Newman, P. A., Hou, G. W., and Korivi, V. M., "First- and Second-Order Aerodynamic Sensitivity Derivatives via Automatic Differentiation with Incremental Iterative Methods," *J. Computational Physics*, Vol. 129, 1996, pp. 307–331.

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

[7] Soemarwoto, B. I. and Labrujère, T. E., "Airfoil Design and Optimization Methods: Recent Progress at NLR," *Inter. J. for Numerical Methods in Fluids*, Vol. 30, No. 2, 1999, pp. 217–228.

[8] Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 97–0643, 1997.

[9] Anderson, W. K. and Bonhaus, D. L., "Aerodynamic Design on Unstructured Grids for Turbulent Flows," Tech. rep., NASA TM 112867, June 1997.

[10] Korivi, V. M., Taylor III, A. C., Newman, P. A., Hou, G. W., and Jones, H. E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," *J. Computational Physics*, Vol. 113, 1994, pp. 336–346.

[11] Elliott, J. and Peraire, J., "Progress Towards a 3D Aerodynamic Shape Optimization Tool for the Compressible High-Re Navier Stokes Equations Discretized on Unstructured Meshes," AIAA Paper 98–2897, June 1998.

[12] Kim, S., Alonso, J. J., and Jameson, A., "Two–dimensional High–Lift Aerodynamic Optimization Using the Continuous Adjoint Method," AIAA Paper 2000–4741, Sept. 2000.

[13] Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92–0439, 1992.

[14] Nielsen, E. J. and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.

[15] Giles, M. B., "On the Use of Runge-Kutta Time-Marching and Multigrid for the Solution of Steady Adjoint Equations," Tech. Rep. 00–10, Oxford University Computing Laboratory, June 2000.

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

[17] Eleshaky, M. E. and Baysal, O., "Discrete Aerodynamic Sensitivity Analysis on Decomposed Computational Domains," *Computers & Fluids*, Vol. 23, No. 4, 1994, pp. 595–611.

[18] Nielsen, E. J., *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier Stokes Equations and a Discrete Adjoint Formulation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 1998.

[19] Pueyo, A. and Zingg, D., "Efficient Newton-Krylov Solver for Aerodynamic Computations," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1991–1997.

[20] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press Inc., 1981.

[21] Elliott, J. and Peraire, J., "Constrained, multipoint shape optimisation for complex 3D configurations," *Aeronautical Journal*, Vol. 102, No. 1017, 1998, pp. 365–376.

[22] Wrenn, G. A., "An Indirect Method for Numerical Optimization Using the Kreisselmeier-Steinhauser Function," Tech. rep., NASA CR 4220, March 1989.

[23]de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, 1978.

[24]Farin, G. E., *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, Academic Press, 1997.

[25]Pulliam, T. H., "Efficient Solution Methods for the Navier-Stokes Equations," Tech. rep., Lecture Notes for the Von Karman Institute for Fluid Dynamics Lecture Series, Brussels, Jan. 1986.

[26]Dennis Jr., J. E. and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Inc., 1983.

[27]Burgreen, G. W. and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal*, Vol. 34, No. 9, 1996, pp. 1761–1770.

[28]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," *J. Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[29]Drela, M., "Pros & Cons of Airfoil Optimization," *Frontiers of Computational Fluid Dynamics 1998*, edited by D. A. Caughey and M. M. Hafez, World Scientific, Singapore, 1998, pp. 363–381.

[30]Liebeck, R. H., "A Class of Airfoils Designed for High Lift in Incompressible Flow," *J. Aircraft*, Vol. 10, No. 10, 1973, pp. 610–617.