AIAA JOURNAL
Vol. 40, No. 6, June 2002

Newton–Krylov Algorithm for Aerodynamic Design Using the Navier–Stokes Equations

M. Nemec* and D. W. Zingg†
University of Toronto, Toronto, Ontario M3H 5T6, Canada

A Newton–Krylov algorithm is presented for two-dimensional Navier–Stokes aerodynamic shape optimization problems. The algorithm is applied to both the discrete-adjoint and the discrete flow-sensitivity methods for calculating the gradient of the objective function. The adjoint and flow-sensitivity equations are solved using a novel preconditioned generalized minimum residual (GMRES) strategy. Together with a complete linearization of the discretized Navier–Stokes and turbulence model equations, this results in an accurate and efficient evaluation of the gradient. Furthermore, fast flow solutions are obtained using the same preconditioned GMRES strategy in conjunction with an inexact Newton approach. The performance of the new algorithm is demonstrated for several design examples, including inverse design, lift-constrained drag minimization, lift enhancement, and maximization of lift-to-drag ratio. In all examples, the norm of the gradient is reduced by several orders of magnitude, indicating that a local minimum has been obtained. By the use of the adjoint method, the gradient is obtained in from one-fifth to one-half of the time required to converge a flow solution.

Introduction

THE accuracy and efficiency of gradient-based algorithms for aerodynamic design problems are influenced by the performance of the following components: 1) the solution of the flowfield equations and 2) the evaluation of the objective function gradient. Although still a subject of research, current algorithms for the solution of the steady two-dimensional Navier–Stokes equations are accurate and efficient. Among the fastest algorithms are the Newton–Krylov solvers (see Refs. 1–4). For example, promising results are presented by Pueyo and Zingg,[4] who used the preconditioned generalized minimum residual (GMRES)[5] Krylov subspace method in conjunction with an inexact Newton strategy. A critical component in this approach is a fast solution of the linear system at each Newton iteration, which is provided by the preconditioned GMRES algorithm. For the aerodynamic shape optimization problem, such Newton–Krylov algorithms are very appealing because they not only provide fast solutions to the flowfield equations, but the preconditioned GMRES algorithm can also be used to compute the objective function gradient.

Among the most promising computational methods for the evaluation of the objective function gradient are the flow-sensitivity, or direct, method and the adjoint method.[6] Both methods have been applied to the Navier–Stokes design problem and can be further subdivided into the continuous[7–11] and the discrete approach.[12–19] The main advantage of the adjoint method is that the computational cost of the gradient calculation is virtually independent of the number of design variables. However, flow sensitivities can be useful for design problems that contain constraints that are dependent on the flowfield variables. Furthermore, it may be advantageous to implement both methods because the resulting information can be used to accelerate the convergence of the design problem by constructing better approximations of the Hessian matrix.[16,20,21]

Jameson et al.[8] 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[22] with the Baldwin–Lomax and the one-equation Spalart–Allmaras[23] turbulence models.

Anderson and Venkatakrishnan[13] analyzed both the continuous- and discrete-adjoint methods for unstructured grids and implemented the discrete approach for viscous design problems. Anderson and Bonhaus[14] extended this work to turbulent flows by differentiating the Spalart–Allmaras turbulence model[23] by hand. They report accurate gradients for turbulent design cases. Nielsen and Anderson[24] 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. Similar results are obtained by Kim et al.[18]

In the adjoint and flow-sensitivity methods, the computational cost of the gradient calculation is dominated by the solution of the large linear system of equations that arises from the flow Jacobian matrix. A popular approach to solve the adjoint and flow-sensitivity equations is to use the same scheme that solves the governing flow equations, for example, the explicit and point-implicit multistage Runge–Kutta schemes coupled with multigrid (see Refs. 8, 17, and 19), the approximate-factorization scheme,[15] and also the lower-upper symmetric Gauss–Seidel (LU-SGS) scheme (see Ref. 18). The GMRES algorithm has been used to solve the discrete sensitivity equation for laminar flows[25] and also to solve the discrete adjoint equation in conjunction with a backward-Euler time-marching scheme with multigrid for turbulent flows.[13,24] Generally, the computational effort required to converge the adjoint equation sufficiently 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.[18,26]

In this work, we present a new algorithm for the calculation of the gradient of the objective function via the discrete-adjoint and discrete flow-sensitivity methods. We carefully linearize the two-dimensional Navier–Stokes equations coupled with the Spalart–Allmaras turbulence model.[23] We adopt the approach of Pueyo and Zingg[4] to solve the adjoint and flow-sensitivity equations using the GMRES algorithm in conjunction with a novel preconditioner based on an approximation of the flow Jacobian matrix. Furthermore, the same preconditioned GMRES algorithm is also used within a Newton–Krylov solver (see Ref. 4) for the solution of the flowfield equations.

Received 29 June 2001; revision received 5 January 2002; accepted for publication 22 January 2002.

*Graduate Student, Institute for Aerospace Studies, 4925 Dufferin Street; marian@oddjob.utias.utoronto.ca.

†Professor, Institute for Aerospace Studies, 4925 Dufferin Street; dwz@oddjob.utias.utoronto.ca. Senior Member AIAA.

The objective of this paper is to examine the following issues in detail: 1) the accuracy of the gradient calculation using the adjoint and flow-sensitivity methods and 2) 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.

Problem Formulation

The aerodynamic shape optimization problem consists of determining values of design variables X_D, such that the specified objective function $\mathcal{J}$ is minimized subject to constraint equations C_j:

$$\min_{X_D} \mathcal{J}[X_D, Q(X_D)] \tag{1a}$$

so that

$$C_j[X_D, Q(X_D)] \leq 0, \qquad j = 1, \ldots, N_c \tag{1b}$$

where Q, the flowfield variables, satisfy the governing flowfield equations. In this work, the constraint equations represent airfoil thickness constraints that are used to ensure feasible designs. Hence, they are only a function of the design variables, that is, $C_j(X_D) \leq 0$.

Objective Functions

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

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

where C^*_p represents the target pressure distribution that 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 that, in the least-squares sense, best matches the target pressure distribution.

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

$$\mathcal{J} = \omega_D\left(1 - C_D/C^*_D\right)^2 + \omega_L\left(1 - C_L/C^*_L\right)^2 \tag{3}$$

where C^*_D and C^*_L represent the target drag and lift coefficients, respectively. The weights ω_D and ω_L are user-specified constants. For the maximization of the lift-to-drag ratio problem, we use

$$\mathcal{J} = C_D/C_L \tag{4}$$

Design Variables

The geometry of the airfoil is described with B-spline curves.[27,28] 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. 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 vertical direction for the B-spline control points.

Flowfield Equations

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

$$\frac{\partial \hat{E}(\hat{Q}, X_D)}{\partial \xi} + \frac{\partial \hat{F}(\hat{Q}, X_D)}{\partial \eta} = Re^{-1}\frac{\partial \hat{S}(\hat{Q}, X_D)}{\partial \eta} \tag{5}$$

where $\hat{Q} = J^{-1}Q = J^{-1}[\rho, \rho u, \rho v, e]^T$ 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 $\hat{E}$ and $\hat{F}$ represent the convective flux vectors, and the viscous flux vector is given by $\hat{S}$. Sutherland's law is used to determine the laminar viscosity. The equations are in nondimensional form. For further details, see Ref. 29.

The turbulent viscosity is modeled with the Spalart–Allmaras turbulence model.[23] All test cases considered in this study are assumed to be fully turbulent, and, therefore, the laminar-turbulent trip terms are not used.

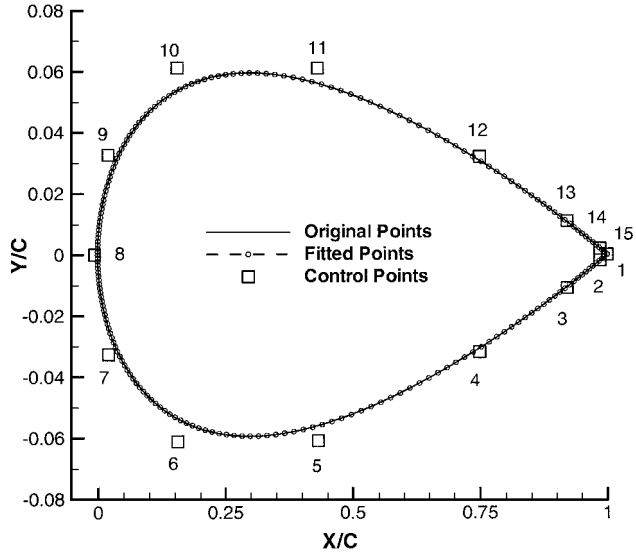

Fig. 1 B-spline curve and control points for the NACA 0012 airfoil.

Numerical Method

Airfoil Thickness Constraints

A penalty method is used to impose airfoil thickness constraints by combining the objective function with the constraint equation:

$$\mathcal{J} = \mathcal{J}_O + \mathcal{J}_T \tag{6}$$

where $\mathcal{J}_O$ denotes the objective functions defined in Eqs. (3) and (4). The thickness constraints are cast as inequality constraints given by the following quadratic penalty term:

$$\mathcal{J}_T = \begin{cases} \omega_T \displaystyle\sum_{i=1}^{N_T}\left[1 - \frac{t(x_i)}{t^*(x_i)}\right]^2 & \text{if} \quad t(x_i) < t^*(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where N_T is the number of thickness constraints, $t^*(x)$ is the minimum allowable airfoil thickness at location x, and ω_T is a user-specified constant. We can also treat the thickness constraints as equality or mixed constraints. Additional constraints or design variables that are useful for practical design include the leading-edge radius and the trailing-edge angle. The present formulation works well for the design cases presented here, but note that there are well-known weaknesses of penalty methods,[30] and more sophisticated strategies for solving constrained problems are given in Refs. 20, 31, and 32.

Flow Solver

The spatial discretization for Eq. (5) is the same as that used in ARC2D.[29] The discretization consists of second-order centered-difference operators with second- and fourth-difference scalar artificial dissipation:

$$\frac{\partial \hat{E}}{\partial \xi} \approx \delta_\xi \hat{E} - \nabla_\xi D_1 D_2 \tag{8}$$

with

$$D_1 = 2\sigma_{j+\frac{1}{2},k} J^{-1}_{j+\frac{1}{2},k}$$

$$D_2 = \epsilon^{(2)}_{j+\frac{1}{2},k}\Delta_\xi Q_{j,k} - \epsilon^{(4)}_{j+\frac{1}{2},k}\Delta_\xi \nabla_\xi \Delta_\xi Q_{j,k}$$

$$\sigma_{j,k} = |U| + c\sqrt{\xi_x^2 + \xi_y^2}, \qquad \epsilon^{(2)}_{j,k} = \kappa_2 \max(\Upsilon_{j+1,k}, \Upsilon_{j,k}, \Upsilon_{j-1,k})$$

$$\epsilon^{(4)}_{j,k} = \max\left(0, \kappa_4 - \epsilon^{(2)}_{j,k}\right), \qquad \Upsilon_{j,k} = \frac{|p_{j+1,k} - 2p_{j,k} + p_{j-1,k}|}{|p_{j+1,k} + 2p_{j,k} + p_{j-1,k}|}$$

where δ_ξ is a second-order centered difference operator, Δ_ξ and ∇_ξ are first-order forward and backward difference operators, U is a

contravariant velocity component, and κ_2 and κ_4 are constants. Typical values of κ_2 and κ_4 are 1.0 and 0.01, respectively. The scalar coefficient σ is the spectral radius of the flux Jacobian matrix. The term $\Upsilon_{j,k}$ is a pressure switch to control the use of first-order dissipation near shock waves. An analogous term appears in the η direction. A far-field circulation correction is also implemented.

The Spalart–Allmaras turbulence model is discretized as described in Ref. 23. Overall, the spatial discretization leads to a nonlinear system of equations:

$$R(\hat{Q}, X_D) = 0 \tag{9}$$

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

Equation (9) 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.[4] The algorithm can be summarized as follows:

1) For matrix-free GMRES(40), the matrix-vector products required at each GMRES iteration are formed using first-order finite differences.

2) The preconditioner is a block-fill incomplete lower/upper decomposition with a fill-in of 2 [BFILU(2)] of an approximate Jacobian matrix. Right preconditioning is used.

3) The reverse Cuthill–McKee reordering of unknowns is based on initial double-bandwidth ordering (see Ref. 4).

The approximate Jacobian is formed from the exact Jacobian $(\partial R/\partial \hat{Q})$ by treating the artificial dissipation coefficients given in Eq. (8), including the spectral radius, as constants and combining the second- and fourth-difference dissipation coefficients as follows:

$$\epsilon_l^{(2)} = \epsilon_r^{(2)} + \phi \epsilon_r^{(4)} \tag{10}$$

where $\phi = 6.0$, the subscript r denotes the contribution from the right-hand side, and the subscript l denotes the resulting left-hand side value used in the preconditioner. Note that this modification does not affect the steady-state solution. To avoid Newton startup problems, the approximate-factorization algorithm of ARC2D in diagonal form,[29] in conjunction with a subiteration scheme[23] for the turbulence model equation, is used to reduce the initial residual by two orders of magnitude. 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. For further details, see Ref. 4. 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.

Adjoint and Flow-Sensitivity Solvers

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

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} + \frac{\partial \mathcal{J}}{\partial Q}\frac{\mathrm{d}Q}{\mathrm{d}X_D} \tag{11}$$

where we reduce the vector of design variables X_D to a scalar to distinguish clearly between partial and total derivatives.

The difficulty in Eq. (11) is the evaluation of the term $\mathrm{d}Q/\mathrm{d}X_D$, referred to as the flow sensitivities. To compute the flow sensitivities, differentiate Eq. (9) with respect to the design variables:

$$\frac{\mathrm{d}R}{\mathrm{d}X_D} = \frac{\partial R}{\partial X_D} + \frac{\partial R}{\partial \hat{Q}}\frac{\mathrm{d}\hat{Q}}{\mathrm{d}X_D} \tag{12}$$

and realize that $\mathrm{d}R/\mathrm{d}X_D = 0$ because, for any design variable, Eq. (9) is always satisfied. Furthermore, note that $\partial \hat{Q}/\partial Q = J^{-1}I$, where I is the identity matrix, and consequently Eq. (12) becomes

$$\frac{\partial R}{\partial Q}\frac{\mathrm{d}Q}{\mathrm{d}X_D} = -\frac{\partial R}{\partial X_D} \tag{13}$$

The direct, or flow-sensitivity, method results from solving Eq. (13) for the flow sensitivities $\mathrm{d}Q/\mathrm{d}X_D$ and using these values in Eq. (11) to obtain the gradient. To formulate the discrete-adjoint method, substitute Eq. (13) into Eq. (11) to obtain

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} - \frac{\partial \mathcal{J}}{\partial Q}\left(\frac{\partial R}{\partial Q}\right)^{-1}\frac{\partial R}{\partial X_D} \tag{14}$$

From the triple-product term in Eq. (14), define the following intermediate problem:

$$\frac{\partial R}{\partial Q}^T \psi = \frac{\partial \mathcal{J}}{\partial Q}^T \tag{15}$$

This is known as the adjoint equation, and the vector ψ represents the adjoint variables. By the substitution of ψ into Eq. (14), the expression for the gradient becomes

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}X_D} = \frac{\partial \mathcal{J}}{\partial X_D} - \psi^T\frac{\partial R}{\partial X_D} \tag{16}$$

Note that Eq. (13) must be solved for each design variable, whereas Eq. (15) is independent of the design variables. If a direct solver is used to solve Eq. (13), then the lower/upper (LU) factorization can be reused with different right-hand-side vectors. Unfortunately, direct solvers are presently only suitable for small, two-dimensional problems. A straightforward implementation of iterative solvers leads to resolving Eq. (13) for each design variable, which is computationally expensive. See Ref. 33 for modifications to iterative solvers that focus on linear systems with multiple right-hand sides; however, even with these solvers, the computational overhead is still significant.

We adopt the GMRES strategy from the flow solver to solve both the adjoint and flow-sensitivity equations. We use right preconditioning with the preconditioner based on the first-order Jacobian matrix described earlier. Fast adjoint and flow-sensitivity solutions are obtained with $\phi = 3.0$, 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 finite differences. In addition to memory savings, the matrix-free approach is easier to implement because an accurate linearization of cumbersome functions in the residual equations, such as the pressure switch [Eq. (8)] and the far-field circulation correction, is automatically provided. Because of the transpose on the left-hand side of Eq. (15), 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, whereas for the drag minimization objective function, it is evaluated using centered differences. The remaining terms in Eq. (16), 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. Note that the evaluation of residual sensitivities includes the evaluation of grid sensitivities because the design variables do not explicitly appear in the residual equations, except for the angle-of-attack design variable. The computational cost of the gradient calculation could be reduced by neglecting grid sensitivities for grid points sufficiently far from the airfoil. However, this approach can introduce substantial errors in the gradient calculation.[24] Because the computational cost of the regrid procedure (discussed hereafter) and of the residual evaluation is only a small fraction of the overall gradient calculation, we evaluate the residual sensitivities at every node in the domain.

Optimizer

The optimizer used to solve the aerodynamic design problem defined by Eq. (1) can have a significant impact on the efficiency of the optimization procedure.[34] Note that by using the penalty method to incorporate constraints, we cast the optimization problem as an unconstrained problem. We solve the unconstrained problem using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton algorithm coupled with a backtracking line search. A detailed description of the optimizer is provided in Ref. 35. At each step of the line search, the objective function value and the gradient value are required to construct a local cubic interpolant. The stopping criterion for the optimization is based on an appropriately scaled L_2 norm of

the gradient.[35] We require a reduction of at least five orders of magnitude in the scaled L_2 norm of the gradient to ensure convergence to a local optimum.

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 hereafter for grid lines in the normal direction. This strategy is similar to one of the strategies outlined in Ref. 36. An analogous formulation holds for the streamwise direction.

Given a displacement of the B-spline control point in the vertical direction, the grid movement strategy preserves the location of the outer boundary. The interior nodes along a normal grid line are positioned as determined by

$$y_k^{\text{new}} = y_k^{\text{old}} + \Delta y_1 (1 - S_k) \tag{17}$$

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

$$S_k = \frac{\sum_{i=2}^{k} L_i}{\sum_{i=2}^{k_{\max}} L_i} \tag{18}$$

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 (SPECfp 2000 rating of 562 peak). All subsonic cases are computed on a 265×53 grid, whereas 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. These grids are very similar to those used for detailed accuracy studies presented in Ref. 37 and provide sufficient numerical accuracy for the design cases considered here. The circulation correction is not used unless explicitly stated.

Flow Solver Performance

We evaluate the performance of the Newton–Krylov algorithm on the following test cases: 1) NACA 0012 airfoil at $M_\infty = 0.3$, $\alpha = 6$ deg, and $Re = 2.88 \times 10^6$; and 2) RAE 2822 airfoil at $M_\infty = 0.729$, $\alpha = 2.31$ deg, and $Re = 6.5 \times 10^6$. Both cases are fully turbulent.

Figure 2 shows that the Newton–Krylov (NK) algorithm is approximately from two to three times faster than the approximate-factorization (AF) algorithm. For many cases, this speedup can be even larger. Initially, the convergence rate of both algorithms is identical because the AF algorithm is used as a startup procedure for the NK algorithm.

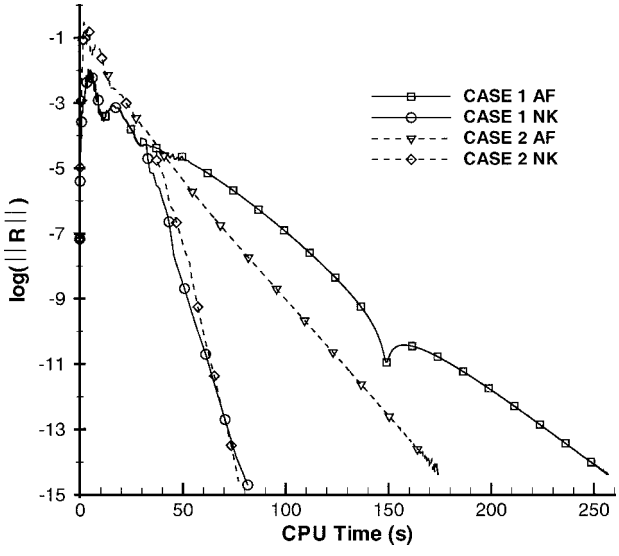

Fig. 2 Performance of the NK algorithm vs the AF algorithm.

One of the main difficulties associated with Newton's method is the startup procedure. This startup procedure can be quite expensive, as shown in Fig. 2 for case 2, where the startup time takes almost half of the flow solve time. The NK algorithm is particularly well suited for the design problem because, once we obtain the solution for the initial airfoil shape, we warm start the remaining flow solves. If the stepsizes during the linesearch procedure are sufficiently small, the startup procedure is not necessary. The warm started flow solves typically converge in two-thirds of the original flow solve time, or roughly 60 s for the cases considered here.

Gradient Accuracy

The factor that influences the accuracy of the gradient calculation most significantly is the linearization of the discretized residual equations, Eq. (9), to obtain the flow Jacobian matrix $(\partial R / \partial \hat{Q})$. For the adjoint method, we carefully linearize the residual equations by hand, including all terms in the Spalart–Allmaras turbulence model.[23] However, exact linearization is complicated by the use of nondifferentiable functions such as the maximum and absolute value functions. These functions are used in the calculation of the pressure switch and spectral radius, as shown in Eq. (8). 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 because the calculation of the vortex strength leads to coupling between airfoil surface points and far-field boundary points. The vortex strength calculation can be linearized as described by Korivi et al.,[15] 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 constants. We linearize the spectral radius of the artificial dissipation scheme, the calculation of vorticity, and the advective terms of the Spalart–Allmaras turbulence model.[23] Note that the derivative of the absolute value function is not defined when the function argument changes sign. The matrix-free implementation of the flow-sensitivity method avoids these linearization difficulties.

We examine the accuracy and efficiency of the gradient calculation for two representative test cases: 1) subsonic inverse design and 2) transonic drag minimization at fixed lift.

For case 1, the freestream conditions are $M_\infty = 0.3$, $\alpha = 6$ deg, and $Re = 2.88 \times 10^6$. 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 Royal Aerospace Establishment (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. (2)] 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, adjoint, and matrix-free flow-sensitivity (S-MF) gradients is very good.

For case 2, the freestream conditions are $M_\infty = 0.7$, $C_L = 0.4728$, and $Re = 9 \times 10^6$. The initial airfoil geometry is the NACA 0012 airfoil. We compute the gradient of the objective function, Eq. (3), with respect to control points 9, 10, 11, and 12 (see Fig. 1), as

Table 1 Gradient accuracy for case 1

Control point no.	Finite difference	Adjoint, % difference	S-MF, % difference
6	−127.82	0	−0.008
7	618.91	0	0
9	−2093.8	0	0
10	−526.58	−0.002	0

Table 2 Gradient accuracy for case 2

Control point no.	Finite difference	Adjoint, % difference	S-MF, % difference
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 3 Gradient accuracy for case 2 with frozen pressure switch

Control point no.	Finite difference	Adjoint, % difference	S-MF, % difference
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 4 Gradient accuracy for case 2 with circulation correction (frozen pressure switch and all absolute value functions)

Control point no.	Finite difference	Adjoint, % difference	S-MF, % difference
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

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 some error in the adjoint gradients relative to the finite difference gradients, which is due to the treatment of the pressure switch. Note that the pressure switch was not used for case 1. 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.

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 that 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 just performed; 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 (S-MF) 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.

Efficiency of Gradient Calculation

Figs. 3 and 4 show the convergence histories 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 preceding section. The GMRES solver parameters are $\phi = 3.0$, BFILU(6), and GMRES(85). The adjoint equation for the inverse design problem takes more iterations to converge than the drag minimization problem, and note that GMRES is forced to perform a restart on iteration 86 as shown in Fig. 3. The flow-sensitivity equation converges well for both the inverse design problem and the drag minimization problem as shown in Fig. 4. The reason for the slower convergence of the inverse design adjoint equation is not fully understood. However, our experience suggests that the slower convergence rate is not due to the given flow conditions. Furthermore, not every inverse design adjoint equation suffers from the slower convergence behavior.

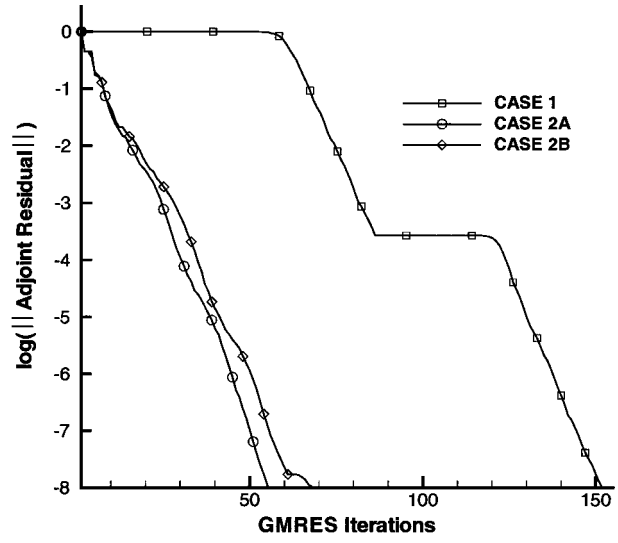

Fig. 3 GMRES convergence for the adjoint equation.

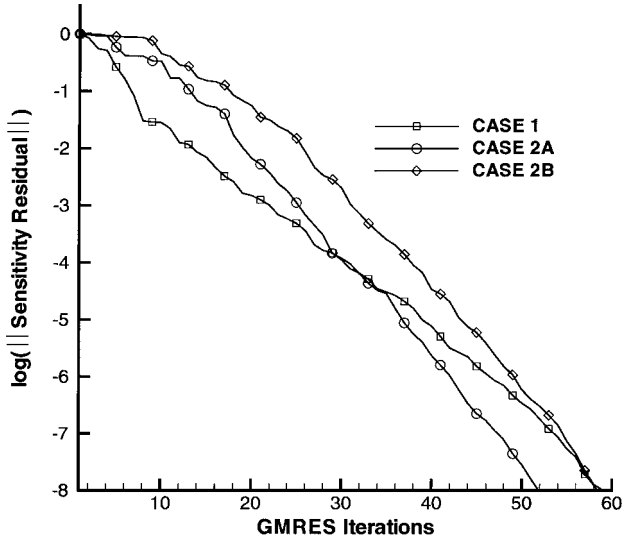

Fig. 4 GMRES convergence for the sensitivity equation (first design variable shown).

Note that residuals of both the adjoint and sensitivity equations should be reduced by three orders of magnitude to obtain gradients of sufficient accuracy.[26,38] For fast GMRES performance, the number of search directions should be selected such that at least a three-order-of-magnitude reduction in the residual is obtained without restarting GMRES.

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, whereas the memory savings are significant.

The efficiency of the adjoint solver for cases 1, 2a, and 2b is summarized in Table 5. T_{prec} refers to the time required to form the preconditioner, T_{GMRES} refers to the time required to reduce the adjoint residual by three orders of magnitude, and flow solve refers

Table 5 Efficiency of adjoint solver[a]

Case	T_{prec}	T_{GMRES}	Total	Flow solve
1	4.2	22.9	27.1	75.2
2a	4.2	7.0	11.2	65.0
2b	2.6	6.6	9.2	65.0

[a]CPU time measured in seconds.

Table 6 Efficiency of matrix-free flow-sensitivity solver[a]

Case	T_{GMRES}, per DV[b]	Total, for all DV
1	5.8	27.4
2a	6.1	34.7
2b	6.5	35.1

[a]CPU time measured in seconds.
[b]Design variables.

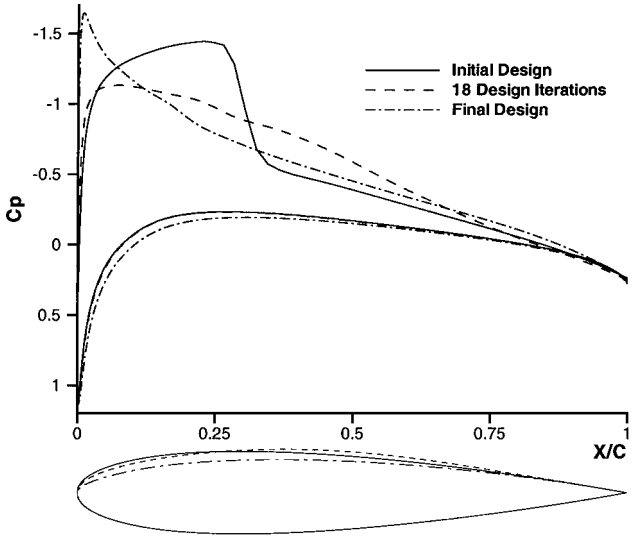

Fig. 5 Pressure distribution and airfoil shape summary for case 2 obtained with the adjoint method.

to the time required for a Newton–Krylov flow solve to converge 10 orders of magnitude. In all design examples in this study, we converge the flowfield solution at least 10 orders of magnitude, and, therefore, the flow solve times in Table 5 provide a good reference. Overall, for case 1, the time to calculate the gradient is just over one-third of the flow solve time, whereas for case 2 it is close to one-sixth of the flow solve time. Case 2b, which uses the BFILU(4) preconditioner, is even faster because 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 T_{GMRES} refers to the time per design variable required to reduce the residual of the flow-sensitivity equation by three orders of magnitude, and total is the total time required to calculate the gradient given by $T_{\text{prec}} + N_{\text{DV}} \cdot T_{\text{GMRES}}$, where N_{DV} is the number of design variables. The time to form the preconditioner and the flow solve time are shown in Table 5.

Design Examples

Having demonstrated the accuracy and efficiency of the gradient calculation, we now solve the aerodynamic shape optimization problem for case 2, which is the drag minimization problem at fixed lift described earlier. Note that this problem does not have a unique solution. We solve this problem with the adjoint and S-MF methods to examine the impact of gradient accuracy on the optimization problem.

Figure 5 indicates that after 18 design iterations the upper surface shock is eliminated. We only plot the adjoint results because those obtained with the S-MF 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,

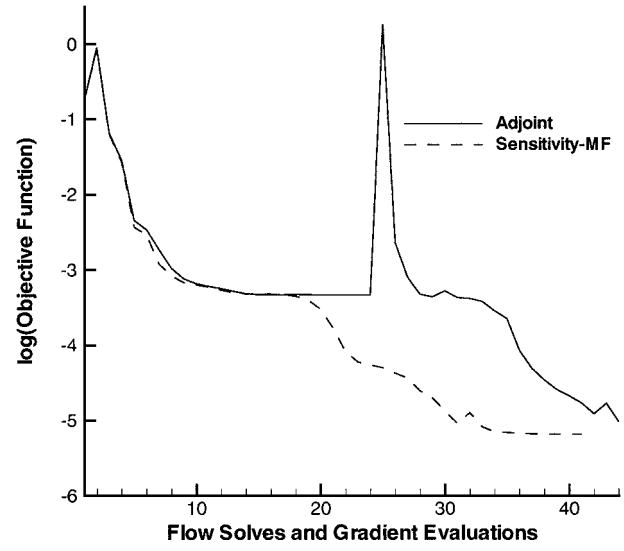

Fig. 6 Comparison of the adjoint method and matrix-free sensitivity method for case 2.

for the flow-sensitivity method. At this point, the adjoint method does not appear to be significantly affected by the errors in its gradient calculation. The convergence of the objective function stalls during the subsequent search direction for the adjoint method, as shown in Fig. 6. The BFGS algorithm is restarted using the steepest descent direction, and the adjoint method catches up to the S-MF 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 M-FS method. The final pressure distribution and the airfoil shape are shown in Fig. 5.

Note that the stalling of the adjoint method appears to be case dependent, and we find that for most design problems the two methods have very similar convergence histories. The BFGS algorithm generates positive-definite approximations to the Hessian matrix only if exact line searches are performed. Hence, the corrupted search direction that causes the stall in the optimization procedure may be a result of inexact line searches and numerical error, as well as gradient inaccuracy.

Three additional design examples are provided 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 a B-spline approximation of the RAE 2822 airfoil at $M_\infty = 0.7$, $\alpha = 3$ deg, and $Re = 9 \times 10^6$. 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 1, 8, and 15 in Fig. 1) are kept constant during the optimization.

Figure 7 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 Fig. 7 are the pressure distribution and airfoil shape, after 30 design iterations, that are very close to the target. The optimization history is summarized in Fig. 8. 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 h.

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 $M_\infty = 0.3$, $\alpha = 6$ deg, and $Re = 2.88 \times 10^6$. 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 equal to the initial drag coefficient. The initial airfoil

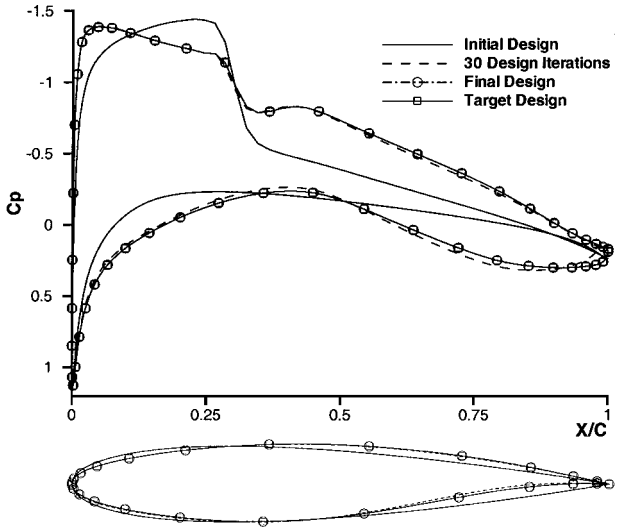

Fig. 7 Pressure distribution and airfoil shape summary for the inverse design problem (12 design variables).

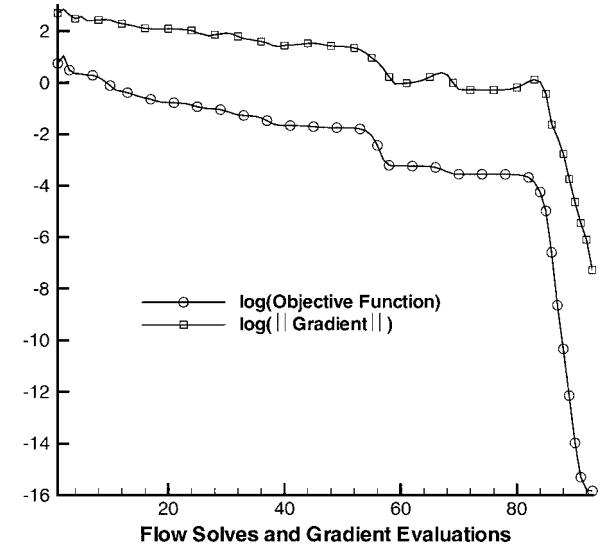

Fig. 8 Optimization convergence history for the inverse design problem (12 design variables).

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. In addition, we specify four thickness constraints. The value of ω_T in Eq. (7) is set to 1.0. Note that, for truly practical designs, multipoint optimization must be used.[38,39]

Figure 9 shows the final pressure distribution and the corresponding airfoil shape. Figure 10 shows that the optimization required 90 design iterations to converge. Because 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 deg.

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 $M_\infty = 0.25$, $\alpha = 9$ deg, and $Re = 2.88 \times 10^6$. The airfoil shape is described with 15 B-spline control points, 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.[40] Five thickness constraints are specified, with the minimum allowable thickness at 25% chord equal to 10%. The value of ω_T in Eq. (7) is set to 0.05.

Figure 11 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

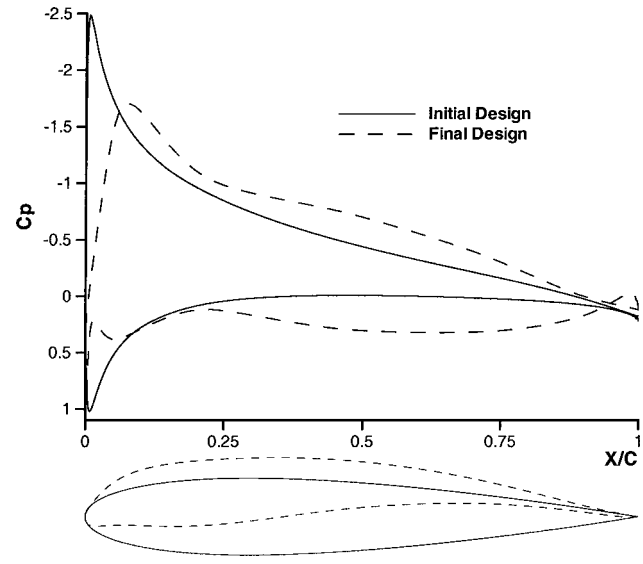

Fig. 9 Pressure distribution and airfoil shape summary for the lift enhancement problem with thickness constraints (15 design variables).

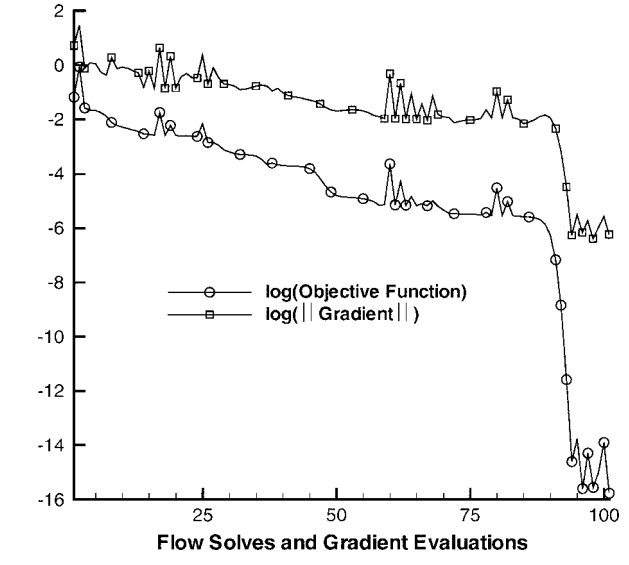

Fig. 10 Objective function convergence history for the lift enhancement problem with thickness constraints (15 design variables).

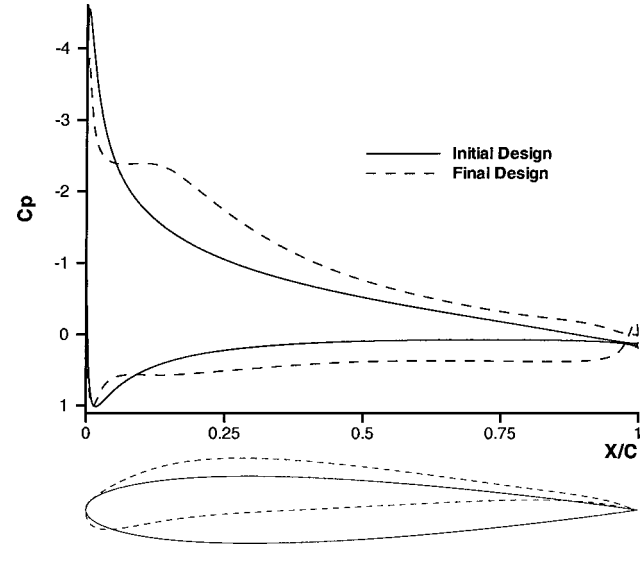

Fig. 11 Pressure distribution and airfoil shape summary for the maximization of C_L/C_D with thickness constraints (10 design variables).

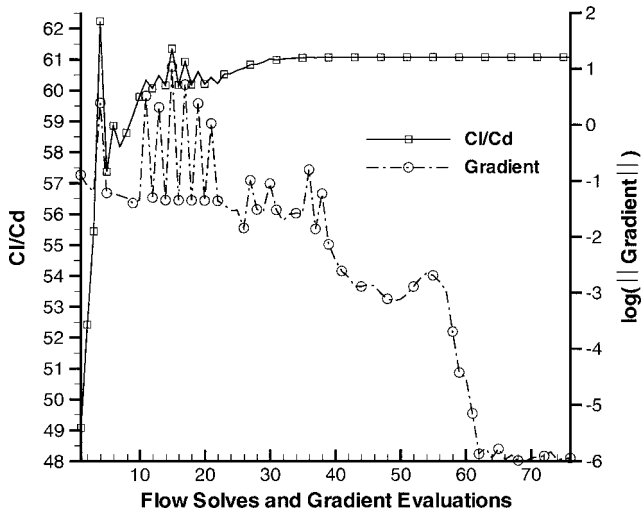

Fig. 12 Optimization convergence history for the maximization of C_L/C_D with thickness constraints (10 design variables).

0.967 to 1.49, whereas 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 five orders of magnitude as shown in Fig. 12. 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.

Conclusions

We have presented a novel algorithm for two-dimensional aerodynamic shape optimization, including both the discrete-adjoint and discrete flow-sensitivity approaches. Based on our results, we can draw the following conclusions:

1) The adjoint gradients show excellent accuracy for subsonic cases, with some error in transonic cases. The 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 residual equations. The small resulting error does not significantly affect the final design.

2) The flow-sensitivity gradients show excellent accuracy in all cases as a result of the matrix-free implementation of GMRES, which is not possible with the adjoint method. However, for problems with a large number of design variables, the adjoint method is more efficient.

3) By the use of the adjoint method, the objective function gradient is calculated in one-fifth to one-half the cost of a warm-started flow solve. Note that the adjoint residual is reduced at least three orders of magnitude to ensure sufficiently accurate gradients.

4) For all design examples, the L_2 norm of the gradient is reduced by several orders of magnitude, which indicates convergence to a local optimum.

The new algorithm provides an efficient means of applying the discrete-adjoint method to aerodynamic design problems governed by the Navier–Stokes equations. This is particularly relevant to problems in which viscous drag and boundary-layer separation are important. With the present algorithm, the relative expense of the gradient calculation is significantly less than that of the flow solver. Future work should concentrate on the optimizer used to update the design variables to reduce the number of flow solves and gradient calculations required to reach the optimum solution.

Acknowledgments

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

References

[1]Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes," *Journal of Computational Physics*, Vol. 105, No. 1, 1993, pp. 83–91.

[2]Barth, T. J., and Linton, S. W., "Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation," AIAA Paper 95-0221, Jan. 1995.

[3]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 391–408.

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

[5]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.

[6]Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, 1974, pp. 97–110.

[7]Jameson, A., "Aerodynamic Design via Control Theory," NASA CR-181749, Nov. 1988.

[8]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.

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

[10]Nadarajah, S., and Jameson, A., "Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," AIAA Paper 2000-0667, Jan. 2000.

[11]Sung, C., and Kwon, J. H., "Aerodynamic Design Optimization Using the Navier–Stokes and Adjoint Equations," AIAA Paper 2001-0266, Jan. 2001.

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

[13]Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Computers and Fluids*, Vol. 28, No. 4–5, 1999, pp. 443–480.

[14]Anderson, W. K., and Bonhaus, D. L., "Airfoil Design on Unstructured Grids for Turbulent Flows," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 185–191.

[15]Korivi, V. M., Taylor, A. C., III, Newman, P. A., Hou, G. W., and Jones, H. E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," *Journal of Computational Physics*, Vol. 113, No. 2, 1994, pp. 336–346.

[16]Sherman, L. L., Taylor, A. C., III, 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," *Journal of Computational Physics*, Vol. 129, No. 2, 1996, pp. 307–331.

[17]Elliott, J., and Peraire, J., "Progress Towards a Three-Dimensional Aerodynamic Shape Optimization Tool for the Compressible High Reynolds Number Navier–Stokes Equations Discretized on Unstructured Meshes," AIAA Paper 98-2897, June 1998.

[18]Kim, C. S., Kim, C., and Rho, O. H., "Sensitivity Analysis for the Navier–Stokes Equations with Two-Equation Turbulence Models," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 838–845.

[19]Giles, M. B., Duta, M. C., and Müller, J.-D., "Adjoint Code Developments Using the Exact Discrete Approach," AIAA Paper 2001-2596, June 2001.

[20]Drela, M., "Design and Optimization Method for Multielement Airfoils," AIAA Paper 93-0969, 1993.

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

[22]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.

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

[24]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.

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

[26]Nielsen, E. J., "Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier–Stokes Equations and a Discrete Adjoint Formulation," Ph.D. Dissertation, Dept. of Aerospace Engineering, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, Dec. 1998.

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

[28]Farin, G. E., *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed., Academic Press, San Diego, CA, 1997.

[29]Pulliam, T. H., "Efficient Solution Methods for the Navier–Stokes Equations," Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, von Kármán Inst. for Fluid Dynamics, Brussels, Jan. 1986.

[30]Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, Toronto, ON, Canada, 1981.

[31]Elliott, J., and Peraire, J., "Constrained, Multipoint Shape Optimisation for Complex 3D Configurations," *Aeronautical Journal*, Vol. 102, No. 1017, 1998, pp. 365–376.

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

[33]Simoncini, V., and Gallopoulos, E., "An Iterative Method for Nonsymmetric Systems with Multiple Right-Hand Sides," *SIAM Journal on Scientific and Statistical Computing*, Vol. 16, No. 4, 1995, pp. 917–933.

[34]Jameson, A., and Vassberg, J. C., "Studies of Alternative Numerical Optimization Methods Applied to the Brachistochrome Problem," *Computational Fluid Dynamics Journal*, Vol. 9, No. 3, 2000, pp. 281–296.

[35]Dennis, J. E., Jr., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[36]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.

[37]Zingg, D. W., Rango, S. D., Nemec, M., and Pulliam, T. H., "Comparison of Several Spatial Discretizations for the Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 160, No. 2, 2000, pp. 683–704.

[38]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," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[39]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.

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

J. Kallinderis
Associate Editor