AIAA Journal
Vol. 50, No. 3, March 2012

Aerodynamic Shape Optimization of Wings Using a Parallel Newton–Krylov Approach

Timothy M. Leung* and David W. Zingg†
University of Toronto, Toronto, Ontario M3H 5T6, Canada

DOI: 10.2514/1.J051192

A Newton–Krylov algorithm for aerodynamic shape optimization in three dimensions is presented for both single-point and multipoint optimization. An inexact Newton method is used to solve the Euler equations, a discrete adjoint method is used to compute the gradient, and an optimizer based on a quasi-Newton method is used to find the optimal geometry. The flexible generalized minimal residual method is used with approximate Schur preconditioning to solve both the flow equation and the adjoint equation. The wing geometry is parameterized by B-spline surfaces, and a fast algebraic algorithm is used for grid movement at each iteration. An effective strategy is presented to enable simultaneous optimization of planform variables and section shapes. Optimization results are presented with up to 225 design variables to demonstrate the capabilities and efficiency of the approach.

Nomenclature

b	=	wing span
C_L, C_D	=	lift and drag coefficients
C_j	=	geometric constraint
c	=	wing chord
E, F, G	=	inviscid fluxes
$\mathbf{G}_S$	=	matrix containing the x, y, or z coordinates of the surface grid
$\mathcal{G}$	=	gradient vector
$\mathcal{H}$	=	Hessian matrix
I	=	integral of the objective function over a range of operating conditions
$\mathcal{J}$	=	objective function
M	=	mach number
$\mathcal{M}$, $\mathcal{N}$	=	B-spline basis functions
P	=	weighting function for multipoint optimization
$\mathcal{P}$	=	search direction
Q	=	continuous state (flow) variables
$\mathbf{Q}$	=	discretized state (flow) variables
$\mathbf{R}$	=	residual vector
$\mathbf{T}$	=	diagonal matrix containing reciprocals of local time steps
$\mathcal{X}$	=	design variables
$\mathcal{X}_B$	=	matrix containing the coordinates of B-spline control points
α	=	angle of attack
Γ	=	dihedral angle of a wing
Λ_{LE}, Λ_{TE}	=	leading-edge and trailing-edge weep angle
λ	=	taper ratio of a wing
$\boldsymbol{\Psi}$	=	flow adjoint vector
Ω	=	geometric twist of a wing
Æ	=	aspect ratio of a wing

I. Introduction

In THE aerodynamic design of aircraft, there have been two major breakthroughs in design methodology. The first is the development of computational fluid dynamics (CFD). In the last decade, CFD has emerged as an indispensable design tool for aircraft aerodynamics, complementing and sometimes replacing wind tunnel testing. Improvements in computer hardware have allowed engineers to solve larger and more complex CFD problems. However, using CFD as an analysis tool alone still relies on a cut-and-try approach, which does not necessarily tell engineers where and how design improvements can be achieved. Inverse design methods have been popular, but they rely heavily on the expertise and experience of aerodynamicists.

The incorporation of high-fidelity CFD codes into automated optimization tools represents the second breakthrough. Unlike inverse design, aerodynamic shape optimization seeks to directly improve the performance measures such as lift and drag coefficients. Such tools are useful in the refinement of existing designs; they can also serve as an inexpensive numerical testbed for unconventional configurations. Beginning with the early work of Hicks et al. [1] and Hicks and Henne [2], aerodynamic shape optimization has become popular. The speed and effectiveness of aerodynamic shape optimization has improved significantly through the development of the adjoint method [3,4]. The main advantage of the adjoint method is that the time required for each gradient computation is nearly independent of the number of design variables. Adjoint methods are further divided into continuous [5–10] and discrete [11–19] approaches. Both have been implemented successfully in aerodynamic shape optimization.

Many examples in wing design optimization consider only one operating condition, such as a fixed Mach number and a fixed lift coefficient (single-point optimization). This serves to prove the effectiveness of an algorithm but is not a practical approach to design. A wing must operate in a range of conditions in the flight envelope, for example, a range of Mach numbers and lift coefficients. Tradeoffs and compromises must be made between different operating conditions to achieve an overall optimum. Nemec et al. [15] studied multipoint optimization of airfoils using a discrete adjoint method. Zingg and Elias [20] investigated automatic selection of sampling points and weights. The work is extended in Zingg and Billing [21] as well as Buckley et al. [22] to include multiple cruise and dive conditions. Studies in multipoint wing design include those by Reuther et al. [7,8], Cliff et al. [23], and Leoviriyakit and Jameson [10]. Jameson et al. [24] studied multipoint aerostructural optimization of wings.

The objective of this paper is to present a novel algorithm for aerodynamic shape optimization of wings and to demonstrate its effectiveness through several examples. The algorithm is an

Presented at the AIAA Fluid Dynamics Conference, San Antonio, 22–25 June 2009; received 24 January 2011; revision received 29 August 2011; accepted for publication 15 September 2011. Copyright © 2011 by Timothy M. Leung and David W. Zingg. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/12 and $10.00 in correspondence with the CCC.

*Research Engineer, Institute for Aerospace Studies. Member AIAA.

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics, J. Armand Bombardier Chair in Aerospace Flight. Associate Fellow AIAA.

extension of the robust algorithm developed by Nemec and Zingg [14,15] for two-dimensional airfoil optimization. Numerical solution of the three-dimensional Euler equations is accomplished using a parallel Newton–Krylov–Schur algorithm developed by Hicken and Zingg [25]. Hicken and Zingg [26] used the same flow solver in an optimization algorithm that is designed to enable large shape changes as might be encountered in the design of unconventional aircraft configurations. The present approach is intended specifically for wing design. To this end, an effective strategy is developed to enable simultaneous optimization of both planform variables and section shapes. Examples are presented to demonstrate its effectiveness in optimizing a wing geometry to minimize both wave and induced drag.

II. Problem Formulation

The goal of aerodynamic shape optimization is to find a shape parameterized by a set of design variables $\mathcal{X}$ such that a scalar objective function $\mathcal{J}$ is minimized:

$$\min_{\mathcal{X}} \mathcal{J}(\mathcal{X}, \mathbf{Q}) \tag{1}$$

where $\mathbf{Q}$ is the vector of state variables representing the flow solution at each node of the computational grid. To ensure the optimizer yields a physically realistic shape, a set of geometric constraints is imposed. We consider only constraints that are functions of design variables:

$$C_j(\mathcal{X}) \le 0 \tag{2}$$

In addition, the discrete steady Euler equations must also be satisfied:

$$\mathbf{R}\,(\mathcal{X}, \mathbf{Q}) = 0 \tag{3}$$

A. Objective Function

For aerodynamic shape-optimization problems, objective functions are generally based on performance measures, such as lift and drag coefficients (C_L, C_D). In the design problems presented in this paper, our objective is to minimize drag while maintaining a required lift. Introducing the lift constraint as a penalty leads to the following objective function [14]:

$$\mathcal{J}_0 = \omega_L\left(1 - \frac{C_L}{C_L^*}\right)^2 + \omega_D\left(1 - \frac{C_D}{C_D^*}\right)^2 \tag{4}$$

Targets in lift and drag are specified by the user in C_L^* and C_D^*, whereas weights ω_L and ω_D are specified such that the lift constraint is adequately satisfied. If the target lift is attainable and target drag is not, this objective function represents lift-constrained drag minimization with the lift constraint appearing as a penalty term. The target drag C_D^* should always be a value that is physically unattainable to ensure that the final drag is minimized, but better convergence is obtained if the target value is not too low. Values of $\omega_L = 100.0$ and $\omega_D = 1.0$ are suitable to maintain the final C_L to within 0.5% of C_L^* for a wide range of drag-minimization problems [27].

B. Multipoint Optimization

For multipoint optimization, we are interested in improving aerodynamic performance over a range of operating conditions. In this case, the quantity that we wish to minimize is the weighted integral I of the objective function over a range of Mach numbers:

$$I = \int_{M_l}^{M_h} P(M)\mathcal{J}[\mathcal{X}, \mathbf{Q}(M)]\,\mathrm{d}M \tag{5}$$

The user-specified weighting function $P(M)$ reflects the relative importance attached by the designer to each Mach number in this range. Rather than minimizing Eq. (5) directly, we instead apply Newton–Cotes quadrature rules to approximate it using a weighted sum of N_p discrete operating points:

$$\mathcal{J}_0 = \sum_{i=1}^{N_p} \underbrace{\omega_i' P(M_i)}_{\omega_i} \mathcal{J}_i \approx I \tag{6}$$

The weights ω_i combine both the weighting function $P(M_i)$ as well as the weights ω_i', which are based on the specific quadrature rule used. In this current study, we assume that $P(M) = 1$; in other words, all operating points are of equal importance to the designer. If we select equally spaced operating points between M_l and M_h, with the first and last operating point at M_l and M_h, respectively, and apply the trapezoidal rule, we obtain the following weights:

$$\omega_i = \begin{cases} 0.5 & i = 1 \\ 1.0 & i = 2\ldots N_p - 1 \\ 0.5 & i = N_p \end{cases} \tag{7}$$

Note that ω_i must be adjusted for other weighting functions $P(M)$ or if the operating points are not equally spaced.

C. Geometric Constraints

We have implemented two geometric constraints: a volume constraint and a thickness constraint. These constraints are expressed as quadratic penalty terms in the objective function:

$$\mathcal{J} = \mathcal{J}_0 + \mathcal{J}_{p,V} + \mathcal{J}_{p,T} \tag{8}$$

The volume constraint is imposed to limit the change in the volume enclosed by the wing. A penalty term is added when the volume V deviates from the initial volume V_0. Written in the form of Eq. (2), the volume constraint is expressed as:

$$|V - V_0| = 0 \tag{9}$$

If the constraint is violated, a penalty term is added to the objective function:

$$\mathcal{J}_{p,V} = \omega_V\left(1 - \frac{V}{V_0}\right)^2 \tag{10}$$

The penalty weight ω_V is user-supplied. A value of $\omega_V = 50.0$ is suitable for a wide range of drag-minimization problems. The volume constraint gives the optimizer more flexibility than applying a thickness constraint near midchord [28]. Therefore, we only apply thickness constraints near leading and trailing edges to ensure that the optimized geometry is physically possible, i.e., to prevent crossover. For thickness constraints, we specify minimum thicknesses at fixed relative positions $[x/c$ and $y/(b/2)]$ on the wing. The ith thickness constraint is expressed as

$$t_i^* - t_i \le 0 \tag{11}$$

where t_i is the thickness at the current optimization iteration, and t_i^* is the minimum required thickness. A penalty term is added if t_i is below t_i^*:

$$\mathcal{J}_{p,i} = \left(1 - \frac{t_i}{t_i^*}\right)^2 \quad \text{if } t_i < t_i^* \tag{12}$$

The contributions from all thickness constraints are summed and multiplied by a user-supplied weight ω_T:

$$\mathcal{J}_{p,T} = \omega_T \sum_i \mathcal{J}_{p,i} \tag{13}$$

We use a penalty weight of $\omega_T = 50.0$. This value is also suitable for a wide range of optimization problems. By casting the constraints as penalty terms, our original optimization problem [Eq. (1)] can be solved using an algorithm for unconstrained optimization.

III. Geometry Parameterization and Design Variables

We use B-spline control surfaces based on Fudge et al. [29] to parameterize the geometry of the wing. In this method, the kth order

B-spline representation of a surface in 3-D space using $M \times N$ control points and basis functions is given by

$$\mathbf{a}\,(s,t) = \sum_{j=1}^{N} \sum_{i=1}^{M} (\mathcal{X}_B)_{i,j} \mathcal{M}_{i,k}(s) \mathcal{N}_{j,k}(t) \tag{14}$$

where $\mathbf{a}$ is the position vector along the curve at parametric distances s and t from the origin, $(\mathcal{X}_B)_{i,j}$ are the locations of the control points, and $\mathcal{M}_{i,k}(s)$ and $\mathcal{N}_{j,k}(t)$ are the basis functions of order k, defined by the Cox–de Boor relationships [30]:

$$\mathcal{M}_{i,1}(t) = \begin{cases} 1 & \text{if } d_i \leq t < d_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$\mathcal{M}_{i,k}(t) = \left[\frac{t - d_i}{d_{i+k-1} - d_i}\right] \mathcal{M}_{i,k-1}(t) + \left[\frac{d_{i+k} - t}{d_{i+k} - d_{i+1}}\right] \mathcal{M}_{i+1,k-1}(t) \tag{16}$$

where d_i represents the elements of a uniform open knot vector given by

$$d_i = \begin{cases} 0 & 1 \leq i \leq k \\ i - k & k+1 \leq i \leq M \\ M - k + 1 & M + 1 \leq i \leq M + k \end{cases} \tag{17}$$

$\mathcal{N}$ is similarly defined. At the start of the optimization cycle, the surface grid is first parameterized using B-spline control surfaces. For a structured surface grid with I and J nodes in the parametric directions s and t, the B-spline surface can be described in discrete matrix form as

$$\mathbf{G}_S = \mathcal{U}\mathcal{D} \qquad \mathcal{D} = \mathcal{X}_B \mathcal{V} \tag{18}$$

where $\mathbf{G}_S$ contains either the x, y, or z coordinates for each surface grid node (j,k); $\mathcal{U}$ and $\mathcal{V}$ store the basis function values at parametric distances s and t from the grid origin; $\mathcal{D}$ is an intermediate matrix of size $M \times J$; and $\mathcal{X}_B$ is a matrix containing the x, y, or z coordinates of the control points:

$$\mathbf{G}_S = \begin{bmatrix} x_{11} & \cdots & x_{1J} \\ \vdots & & \vdots \\ x_{I1} & \cdots & x_{IJ} \end{bmatrix} \qquad \mathcal{U} = \begin{bmatrix} \mathcal{N}_1(s_1) & \cdots & \mathcal{N}_M(s_1) \\ \vdots & & \vdots \\ \mathcal{N}_1(s_I) & \cdots & \mathcal{N}_M(s_I) \end{bmatrix}$$

$$\mathcal{X}_B = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix} \qquad \mathcal{V} = \begin{bmatrix} \mathcal{M}_1(t_1) & \cdots & \mathcal{M}_1(t_J) \\ \vdots & & \vdots \\ \mathcal{M}_N(t_1) & \cdots & \mathcal{M}_N(t_J) \end{bmatrix} \tag{19}$$

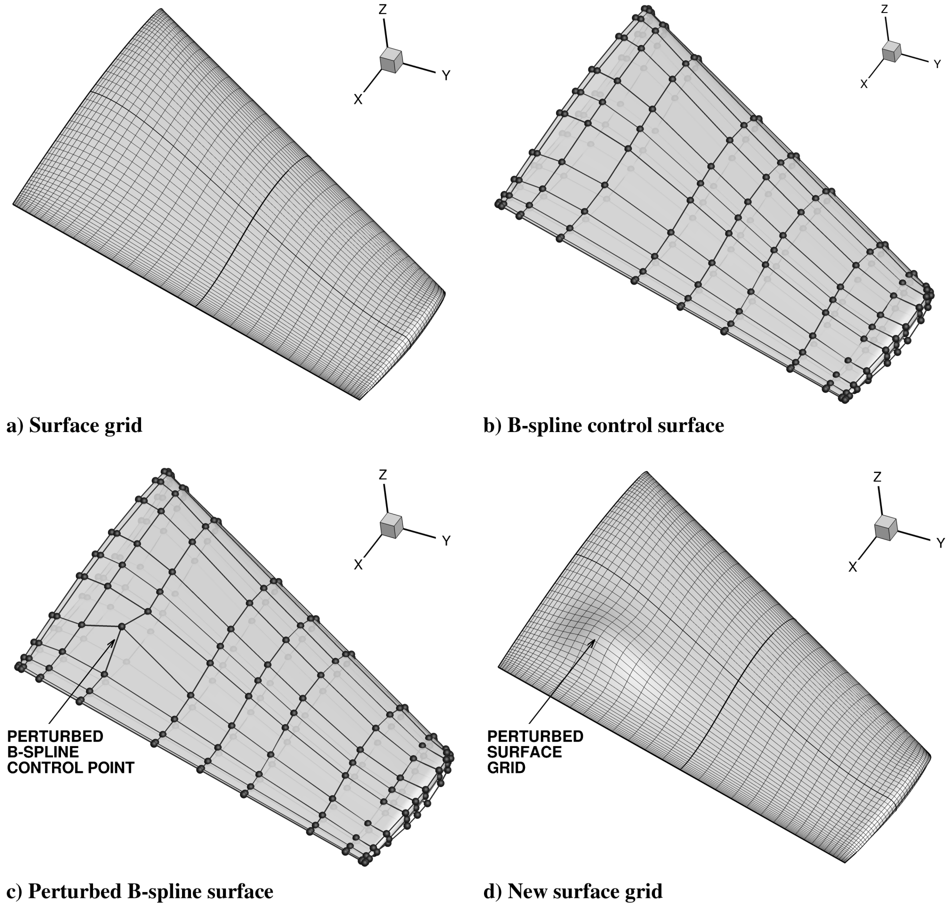
The distances s and t are calculated based on the nodal indices:

$$s_i = \frac{i-1}{I-1}(m - k + 2) \qquad t_j = \frac{j-1}{J-1}(n - k + 2) \tag{20}$$

The control point locations are found by first solving for $\mathcal{D}$ and then $\mathcal{X}_B$ in the least-squares problems in Eq. (18). This process is repeated for each of the three coordinates. Figures 1a and 1b show the parameterization of the ONERA M6 wing. In this example, the wing is represented using cubic B-spline control points, with 13 control points in both the spanwise and chordwise directions on each of the top and bottom surfaces.

To generate a new surface grid in response to changes in the location of the control points, the intermediate matrix $\mathcal{D}$ in Eq. (18) is first generated based on the new control point locations $\mathcal{X}_B$, and then the new surface grid $\mathbf{G}_S$ is generated. An example is shown in Figs. 1c and 1d.

Using this parameterization strategy, we can use two levels of design variables. At the planform level, control points are grouped



a) Surface grid

b) B-spline control surface

c) Perturbed B-spline surface

d) New surface grid

Fig. 1 B-spline surface parameterization of an ONERA M6 wing with a perturbation.

into a reduced set of planform variables, such as semispan ($b/2$), chord (c), leading-edge and trailing-edge sweep (Λ_{LE}, Λ_{TE}), dihedral (Γ) and geometric twist (Ω) angles. The twist angle is defined as a single variable, i.e., it varies linearly from wing root to tip. Other planform parameters such as taper ratio λ and aspect ratio Æ are extracted from the above planform variables. At the wing section level, each individual control point may independently move vertically to adjust the wing section shape; thus, the design variables are the z coordinates of the control points. Increasing the number of control points improves the flexibility of the parameterization and allows more control over changes in the geometry. However, the increased degrees of freedom will also slow the convergence of the optimizer, and therefore the selection of the number of design variables is a compromise between geometric flexibility and optimization convergence.

IV. Numerical Method

A. Flow Analysis

The governing equations for the optimization are the Euler equations, discretized on multiblock structured grids. In our parallel strategy, each block in the grid and the corresponding component of Q is distributed to a separate processor. Thus the discretization of Q is done in parallel in each block. Second-order centered differencing is used at interior nodes, whereas first-order one-sided differencing is used at boundaries and block interfaces. For numerical stability, we use a scalar dissipation model based on the JST model [31,32], with second-difference dissipation near shocks and fourth-difference dissipation everywhere else. Boundary conditions and the coupling between blocks at the interfaces are done using simultaneous approximation terms (SATs). The use of SATs reduces the number of ghost cells necessary and thus reduces the need for interprocessor communications. Details of the implementation of SATs can be found in Hicken and Zingg [25]. Communication between processors is done using the message passing library MPICH.

Spatial discretization of the Euler equations produces a set of nonlinear algebraic equations. These are solved using a parallel Newton–Krylov method. Because the adjoint gradient computation requires the flow-Jacobian matrix ($\partial\mathbf{R}/\partial\mathbf{Q}$) from a well-converged solution, residual reduction of 10 orders of magnitude is used. The linear system is solved using the Krylov subspace iterative method called the flexible generalized minimal residual (FGMRES) method [33]. To improve the convergence of FGMRES, we right-precondition the linear system. An approximate-Schur preconditioner based on Saad and Sosonkina [34] is applied to a first-order approximation of the Jacobian to form the preconditioner. Details of the implementation can be found in Hicken and Zingg [25].

The Newton method converges quadratically when $\mathbf{Q}$ is sufficiently close to the solution. However, during startup, when the iterate is far from the solution, convergence may not be possible. Therefore, for stability during the startup period, the flow solver instead uses an approximate Newton method, where the first-order Jacobian replaces the full Jacobian. A pseudotime step is added for globalization of the Newton method.

B. Optimizer

When the lift and geometric constraints are implemented as quadratic penalty terms, we can consider the optimization as an unconstrained problem. Applying the Newton method in the optimizer, the search direction is given by the gradient $\mathcal{G}$ and the inverse of the Hessian matrix $\mathcal{H}$:

$$\mathcal{P} = -\mathcal{H}^{-1}\mathcal{G} \tag{21}$$

In this current work, we use the BFGS method [35] to approximate the Hessian matrix, whereas the gradient is computed using the discrete adjoint method. For many aerodynamic shape-optimization applications, we can generally reduce the gradient L_2 norm by two to three orders of magnitude.

C. Adjoint Solver

At the heart of any gradient-based optimization is the fast and accurate evaluation of the objective function gradient $\mathcal{G}$. The gradient can be expressed as

$$\mathcal{G} = \frac{\partial\mathcal{J}}{\partial\mathcal{X}} - \mathbf{\Psi}^T\frac{\partial\mathbf{R}}{\partial\mathcal{X}} \tag{22}$$

where $\mathbf{\Psi}$ is the adjoint variable, obtained by solving the adjoint equation:

$$\left(\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\right)^T\mathbf{\Psi} = \left(\frac{\partial\mathcal{J}}{\partial\mathbf{Q}}\right)^T \tag{23}$$

We solve the adjoint system by adopting the same strategy used for the flow solution. The system is solved using FGMRES as the iterative solver, with the approximate Schur preconditioner to right-precondition the system. We specify a tolerance of 10^{-8} for the adjoint system to obtain an accurate gradient. The right-hand-side term $\partial\mathcal{J}/\partial\mathbf{Q}$ is differentiated by hand for each objective function. Finally, the partial derivatives with respect to design variables $\partial\mathcal{J}/\partial\mathcal{X}$ and $\partial\mathbf{R}/\partial\mathcal{X}$ in Eq. (22) are evaluated using second-order centered differencing. For the kth design variable,

$$\begin{aligned}\frac{\partial\mathbf{R}}{\partial\mathcal{X}_k} &\approx \frac{\mathbf{R}[\mathcal{X}+h\hat{\epsilon}_k,\mathbf{Q}] - \mathbf{R}[\mathcal{X}-h\hat{\epsilon}_k,\mathbf{Q}]}{2h}\\ \frac{\partial\mathcal{J}}{\partial\mathcal{X}_k} &\approx \frac{\mathcal{J}[\mathcal{X}+h\hat{\epsilon}_k,\mathbf{Q}] - \mathcal{J}[\mathcal{X}-h\hat{\epsilon}_k,\mathbf{Q}]}{2h}\end{aligned} \tag{24}$$

where $\hat{\epsilon}_k$ is the kth unit vector, and the step size h is given by

$$h = \max(\epsilon\cdot|X_k|, 10^{-6}) \tag{25}$$

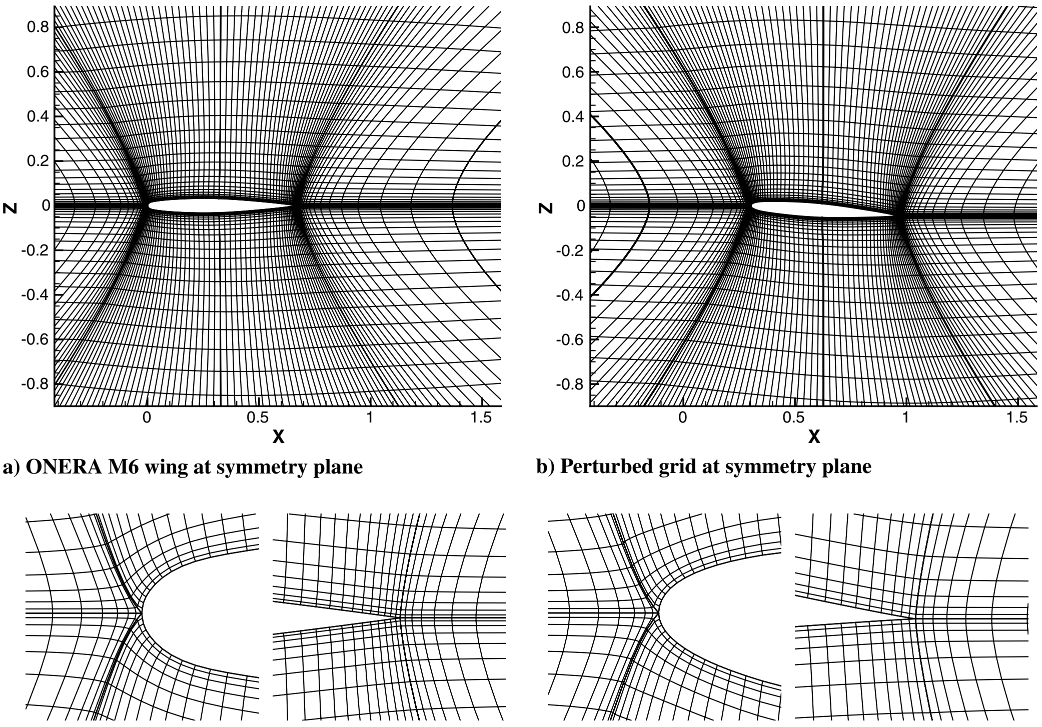
where $10^{-4} > \epsilon > 10^{-6}$. Note that the evaluation of Eq. (24) does not require additional flow solves. The cost of using finite differencing to evaluate Eq. (24) scales with the number of design variables. However, when a fast and inexpensive grid movement algorithm is used, the cost of evaluating these partial derivatives using finite differences is an order of magnitude lower than the cost of solving the adjoint equation. This is the case for the algebraic algorithm used for the current work. If a more expensive grid movement algorithm is used, such as the linear elasticity method [36], then it is more efficient to solve a mesh adjoint system [26,36,37].

D. Line Search

A line-search algorithm with a backtracking feature is used to find a suitable step length β along the search direction $\mathcal{P}_n$ that satisfies the strong Wolfe conditions [35]. To construct a cubic interpolant during a line search, both the objective function and the gradient are computed at every trial point. The line search is considered stalled if a satisfactory step size cannot be found in 15 line-search iterations. In that case, the optimizer is restarted from the steepest descent direction $-\mathcal{G}$, and the approximate Hessian inverse is reset to the identity matrix.

E. Design Variable Scaling

The scaling of the design variables is crucial to the performance of the optimizer. A problem is considered poorly scaled if changes in one variable produce much larger variations in the value of the objective function than changes of comparable magnitude in other variables. This problem arises when we mix B-spline design variables with planform variables. In our experience, $\Delta\mathcal{X}_i$, from initial to optimized geometry, ranges from 10^{-5} to 10^{-3} for B-spline variables, and to 10^0 for angle-of-attack and planform variables (e.g., change in sweep angle). To improve the convergence of the optimizer, we scale the z coordinates of the control points by the square root of their initial values, as discussed in Zingg et al. [28]. This scaling method assumes that both the leading and trailing edges of the wing are at $z = 0.0$ and that the B-spline control points at the leading and trailing edges are not used as design variables. If this is not the case, the coordinates have to be translated before scaling is



a) ONERA M6 wing at symmetry plane

b) Perturbed grid at symmetry plane

c) ONERA M6 wing close up

d) Perturbed grid close up

Fig. 2 Example of the algebraic grid movement algorithm applied to an ONERA M6 wing.

applied. To avoid any divisions by zero, any control points at $z = 0.0$ are scaled by the average scaling of the other design variables. Angle of attack and planform design variables are not scaled.

F. Grid Movement Strategy

A high-quality computational grid is necessary to compute the flow solution at each design iteration. For aerodynamic shape optimization using body-fitted grids, expensive grid regeneration can be avoided by employing a suitable grid movement algorithm. Each time the wing surface changes, the grid is adjusted accordingly. For the current work, we use a fast and robust algebraic grid movement method. In this method, the movement of the nodes along a grid line is determined by the algebraic equation,

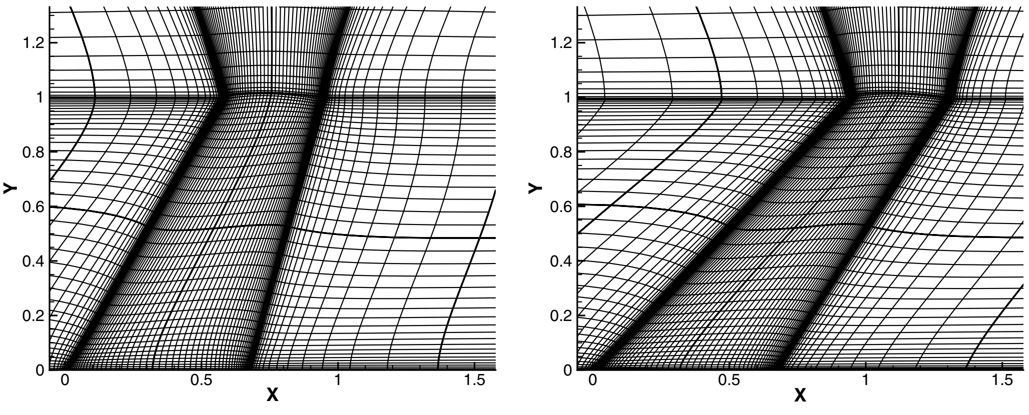
$$\mathbf{x}_k^{\text{new}} = \mathbf{x}_k^{\text{old}} + \frac{\Delta \mathbf{x}_1}{2}[1 + \cos(\pi S_k)] \quad \text{for } k = 2 \ldots k_{\max} \qquad (26)$$

where $\Delta \mathbf{x}_1$ is the displacement of the surface node, $k_{\max}$ is the index of the node on the grid line at the outer boundary, and

$$S_k = \frac{\sum_{i=2}^{k} |\mathbf{x}_i - \mathbf{x}_{i-1}|}{\sum_{i=2}^{k_{\max}} |\mathbf{x}_i - \mathbf{x}_{i-1}|} \qquad (27)$$

is the normalized arclength distance along the grid line. This algebraic grid movement algorithm provides sufficient grid quality for many optimization applications, as long as the changes in the geometry are relatively small and the block boundaries are sufficiently far from the body surface. Examples are shown in Figs. 2 and 3.

In the first example (Fig. 2), the root location of the ONERA M6 wing is displaced by $0.3c_{\text{root}}$, and the wing is rotated by 5 deg with respect to the far-field boundaries. The grid at the symmetry plane is shown. This amount of movement in the grid is typically found near the wing tip for optimization cases where both the sweep angle and twist angles change. In the second example (Fig. 3), a 10-deg sweep is added. The grid lines extending from the wing surface are plotted. In both examples, the grid quality of the perturbed grid remains high.



a) ONERA M6 wing at symmetry plane

b) Perturbed grid at symmetry plane

Fig. 3 Example of the algebraic grid movement algorithm applied to an ONERA M6 wing.

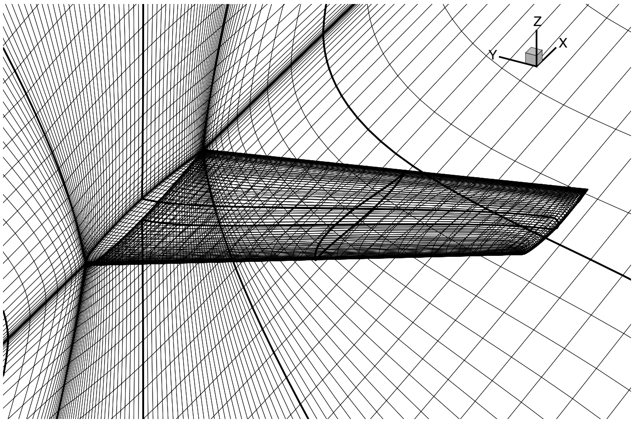


Fig. 4 Grid over the ONERA M6 wing.

V. Results and Discussion

Optimization results are obtained using a Beowulf-class cluster. The cluster uses Intel Xeon 5500 (Nehalem) processors with a CPU speed of 2.53 GHz, with 16 GB of shared memory per computational node (8 processors). The computational nodes are connected by a nonblocking 4x-DDR Infiniband network.

A. Single-Point Wing Optimization

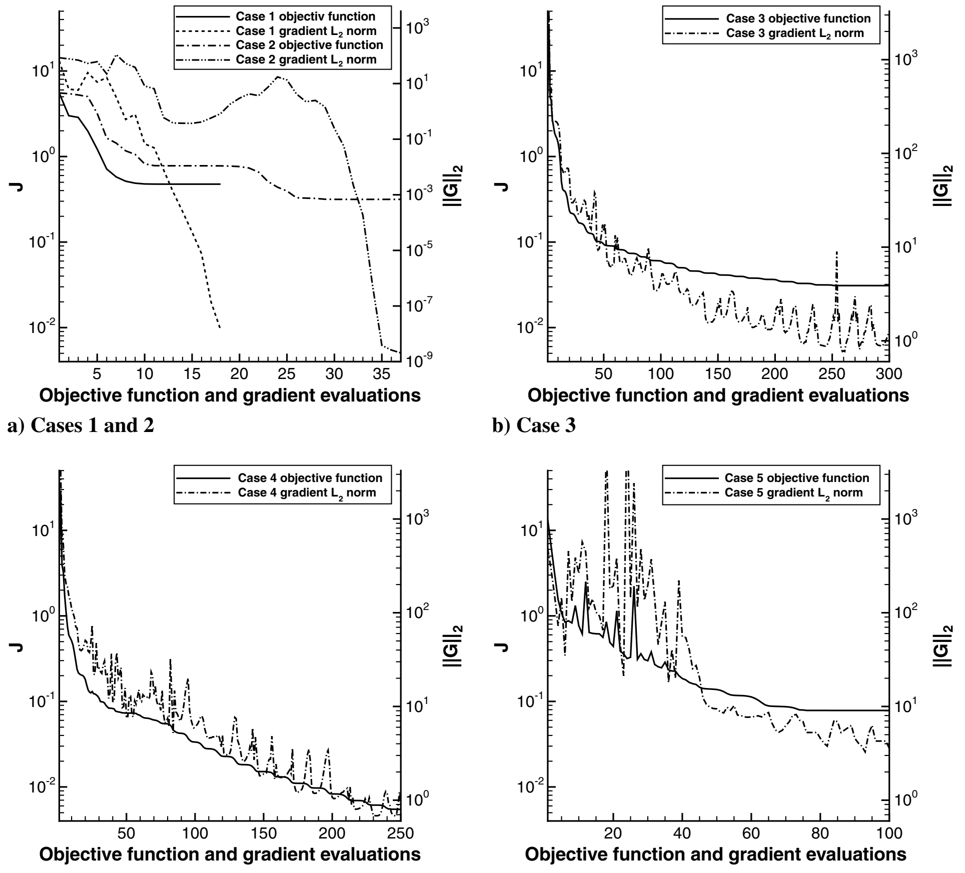
We present single-point optimization results at a transonic speed. The goal of this optimization is to minimize drag at $M = 0.90$ while maintaining a lift coefficient of $C_L = 0.30$. The initial geometry is the ONERA M6 wing. The computational grid around the wing has an H-H topology with 48 blocks and 1,063,000 nodes, as shown in Fig. 4. Each block is distributed to a different processor. The ONERA M6 wing initially generates the required lift at an angle of attack of $\alpha = 2.69$ deg. At this operating condition, the wing has a drag coefficient of $C_D = 0.0265$.

We use the lift-constrained drag-minimization objective function [Eq. (4)], with the following targets and weights in lift and drag:

$$C_L^* = 0.302 \qquad \omega_L = 100.0 \qquad C_D^* = 0.0075 \qquad \omega_D = 1.0$$

We run single-point optimization cases using the following combinations of geometric design variables: 1) leading-edge sweep ($\Delta\Lambda_{\mathrm{LE}}$) only; 2) leading-edge sweep and linear geometric twist (Ω); 3) leading-edge sweep, linear geometric twist, and z coordinates of 225 B-spline control points controlling wing section shapes; 4) leading-edge sweep, linear geometric twist, and z coordinates of 119 B-spline control points controlling wing section shapes (using fewer control points to parameterize the geometry); and 5) z coordinates of 119 B-spline control points controlling wing section shapes.

In Cases 1–3, we parameterize the top and bottom surfaces of the wing using a 13×13 B-spline surface. However, only planform variables are used in Cases 1 and 2. In Cases 4 and 5, we reduce the number of control points in the spanwise direction to seven, thereby reducing the number of design variables by almost a factor of two. In Cases 3–5, where the wing section geometry is also optimized, the z coordinates of all B-spline control points except near the leading and trailing edges are used as design variables. Note that, as sweep (Λ_{LE}) changes, the wing is sheared along the chordwise direction, and thus the planform (projected) area remains constant throughout the optimization cycle. Similarly, twist (Ω) is obtained by translating the control points vertically; thus changing Ω also does not affect the planform area. In Case 5, no planform design variables are used. In Cases 3–5, a volume constraint [Eq. (10)] with a penalty weight of $\omega_V = 50.0$ is used to maintain the wing's volume. A minimum thickness of $(t/c)^* = 0.001$ is also applied at $(x/c) = 0.99$ to prevent grid crossover. The penalty weight for the thickness



a) Cases 1 and 2

b) Case 3

c) Case 4

d) Case 5

Fig. 5 Convergence histories for single-point optimization cases.

Table 1 Final C_D at $C_L = 0.30$ for single-point optimization cases

Case	Final C_D
1	0.0128
2	0.0118
3	0.00826
4	0.00803
5	0.00892

constraint is $\omega_T = 50.0$. In addition to the geometric design variables, the angle of attack (α) is a design variable in all cases.

Convergence histories for all five single-point optimization cases are shown in Fig. 5. In the cases using planform variables alone (Cases 1 and 2), the gradient L_2 norm ($\|\mathcal{G}\|_2$) converges to below 10^{-7} and 10^{-8}, respectively. The optimizer increased Λ_{LE} significantly to reduce wave drag, to $\Lambda_{\mathrm{LE}} = 55$ and 53 deg, respectively, but the final flow fields are not shock free. The final drag coefficients from both cases are much higher than for Cases 3–5, where the optimizer has the freedom to modify the section shapes as well, as shown in Table 1. The addition of the twist design variable in Case 2 further reduces the induced drag somewhat. To maintain the required

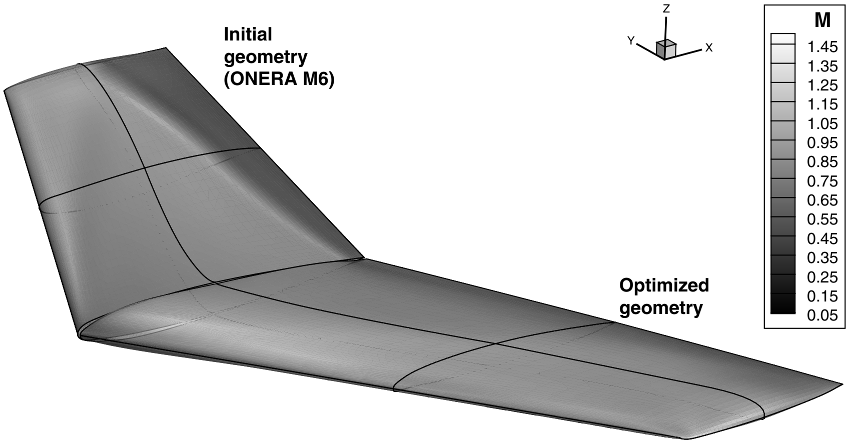
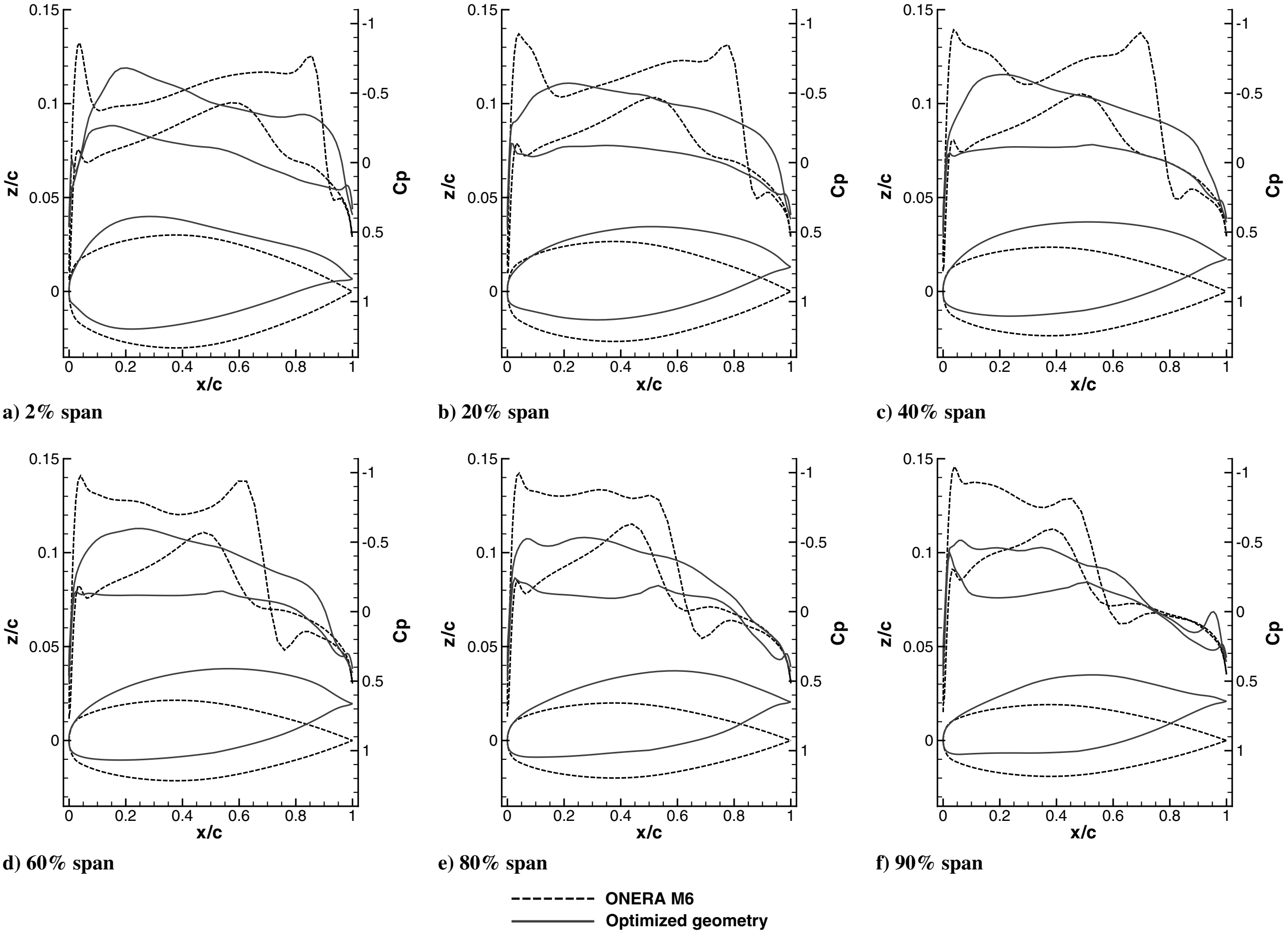


Fig. 6 Comparison of Mach contours for single-point optimization (Case 4).



a) 2% span

b) 20% span

c) 40% span

d) 60% span

e) 80% span

f) 90% span

Fig. 7 Comparison of C_p and wing sections for single-point optimization (Case 4).

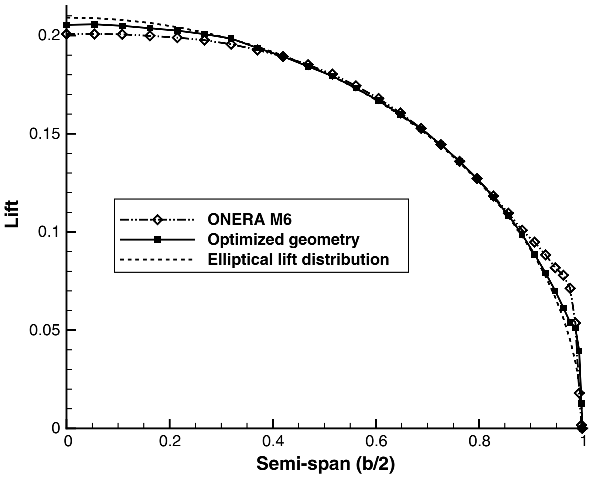


Fig. 8 Lift distribution for the optimized wing (Case 4).

C_L at the higher sweep angles, the angles of attack were increased to 3.90 and 6.22 deg, respectively. In Case 2, a twist of $\Omega = -5.6$ deg (washout) is added at the tip.

In cases where individual B-spline control points are used as design variables (Cases 3–5), both $\mathcal{J}$ and $\|\mathcal{G}\|_2$ are reduced by more than two orders of magnitude. Cases 3 and 4 show further drag reductions, compared with Case 5, by adding sweep angle and twist as design variables. In particular, Case 4, with a reduced number of spanwise sections, converged the fastest. The convergence comparison between Cases 3 and 4 indicates that, with a careful choice of geometry parameterization, it is possible to substantially reduce the number of design variables while obtaining better optimization results and speeding up the optimizer. In general, using seven to eight spanwise sections allows for both speed and geometric flexibility.

Table 2 Cost breakdown of gradient computation compared to flow solution

	Time, s	Relative cost
Flow solution	516	100%
Adjoint solution	253	49%
$\partial\mathbf{R}/\partial\mathcal{X}$ and $\partial\mathcal{J}/\partial\mathcal{X}$	21	4.1%

Table 3 Drag reduction vs iteration count for single-point optimization (Case 4)

Iteration	C_D	Drag reduction
1	0.0265	
20	0.0107	59.8%
50	0.0100	62.1%
100	0.00890	66.3%
250	0.00803	70.0%

In Case 4, after 250 iterations, the lift and drag coefficients are

$$C_L = 0.30 \qquad C_D = 0.00803$$

This represents a 70% drag reduction compared to the ONERA M6 wing. In the optimized geometry from Case 4, Λ_{LE} increased from 30.0 deg to 35.2 deg, and a twist angle of $\Omega = -3.27$ deg was added. The angle of attack decreased slightly to $\alpha = 2.55$ deg. Both volume and thickness constraints are active at the end of the optimization cycle, with the final volume within 0.1% of the initial volume and the thickness within 0.08% of the target minimum thickness.

Surface Mach number contours of the optimized wing are compared to the ONERA M6 wing in Fig. 6. Pressure coefficients and wing sections of the ONERA M6 wing and the optimized wing are plotted at six spanwise stations in Fig. 7. Both figures show that the optimizer has successfully eliminated the wave drag by removing the shocks. The spanwise lift distributions for the ONERA M6 wing and the optimized geometry are compared to the elliptical lift distribution in Fig. 8. It shows that the lift distribution of the optimized geometry matches the elliptical lift distribution more closely, indicating that the induced drag has also been minimized.

During the optimization cycle, a flow solve requires an average of 8.6 min to reduce the residual by 10 orders of magnitude, using 48 processors. The adjoint solver takes an average of 4.2 min to reduce the linear residual by eight orders of magnitude. The evaluation of $\partial\mathbf{R}/\partial\mathcal{X}$ and $\partial\mathcal{J}/\partial\mathcal{X}$ require two calls to the grid movement algorithm per design variable; each grid movement call plus residual evaluation takes only 0.08 s to complete. The cost breakdown of a gradient evaluation is compared to a flow solution in Table 2. The timing shown in the table is averaged over all design iterations.

The entire optimization cycle (250 iterations) for Case 4 took 56 h to complete. Drag coefficients at $C_L = 0.30$ are shown in Table 3 after 20, 50, and 100 iterations. Optimization time can be further shortened by using more processors, because the flow solver is found to scale well up to 1000 processors [38].

B. Multipoint Wing Optimization

Even with a single-point optimization, the improvement in performance is significant over all transonic speeds, compared with the ONERA M6 wing, as shown in Fig. 9. However, a degree of point optimization is visible. Performance degrades rapidly on either side of $M = 0.9$. Multipoint optimization is useful when a wing is to be designed to operate under a range of operating conditions and can eliminate the point optimization phenomenon as well.

The goal of our multipoint optimization is to reduce the drag coefficient at a fixed lift coefficient of $C_L = 0.30$ between $M_l = 0.80$

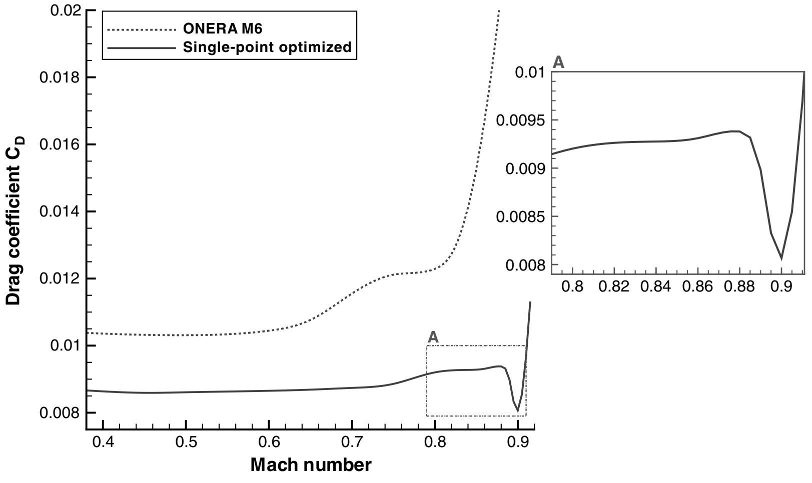
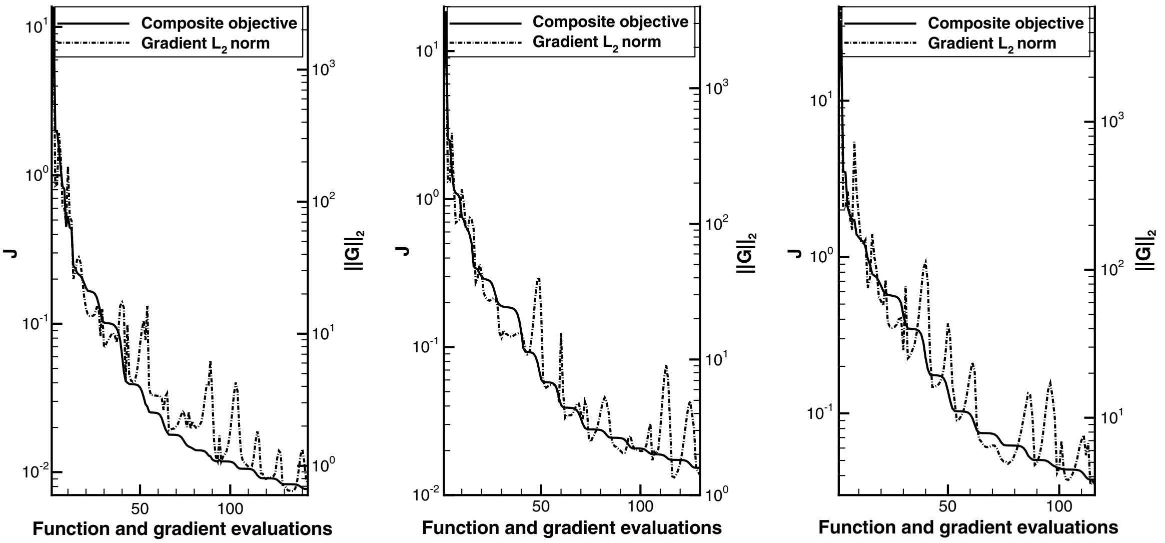


Fig. 9 C_D vs M at $C_L = 0.30$ plot for the single-point optimization (Case 4).



a) Two-point

b) Three-point

c) Five-point

Fig. 10 Convergence histories for multipoint optimization cases.

Table 4 Accuracy of approximation of integral

Case	I	Composite $\mathcal{J}$	Error
Single-point	9.21×10^{-4}		
Two-point	8.46×10^{-4}	8.35×10^{-4}	1.20%
Three-point	8.30×10^{-4}	8.27×10^{-4}	0.63%
Five-point	8.31×10^{-4}	8.30×10^{-4}	0.12%

and $M_h = 0.90$. Note that we did not specify the same target lift over the range of Mach numbers, which scales with the square of the Mach number. This would mean specifying the same $M^2 C_L^*$ over all operating points. Instead, we assume that, as the aircraft increases M, it climbs to a higher altitude (lower density) such that the same lift is generated with a constant C_L. Note that we can only be confident that the weighted integral I [Eq. (5)] has been minimized if the composite objective function $\mathcal{J}$ [Eq. (6)] is a good approximation to I.

In the cases presented, the lift-constrained drag-minimization objective function [Eq. (4)] is used at each operating point; targets and weights in lift and drag (C_L^*, C_D^*, ω_L, and ω_D) are the same as the single-point case. Geometry parameterization, geometric design variables, and geometric constraints are identical to single-point Case 4, and the angles of attack at the additional operating points are also design variables. We initiate the multipoint optimization cases with the final optimized geometry from Case 4 of the single-point cases. Our multipoint optimization cases are performed using the trapezoidal rule for numerical integration with two, three, and five operating points. All operating points are equally spaced, with the first and last operating points at $M = 0.80$ and $M = 0.90$, respectively. Weights ω_i for each operating point are given in Eq. (7).

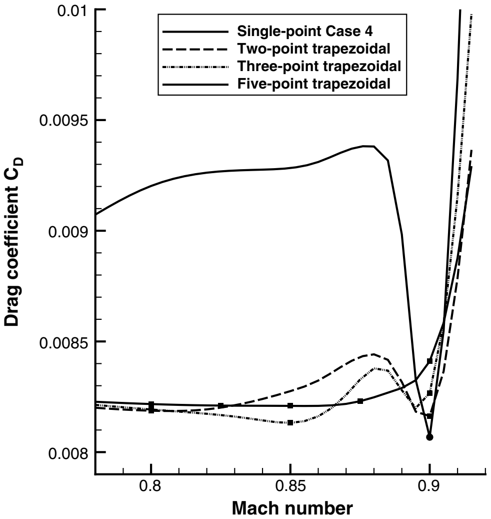


Fig. 11 C_D vs M at $C_L = 0.3$ plots for two-, three-, and five-point optimized wings.

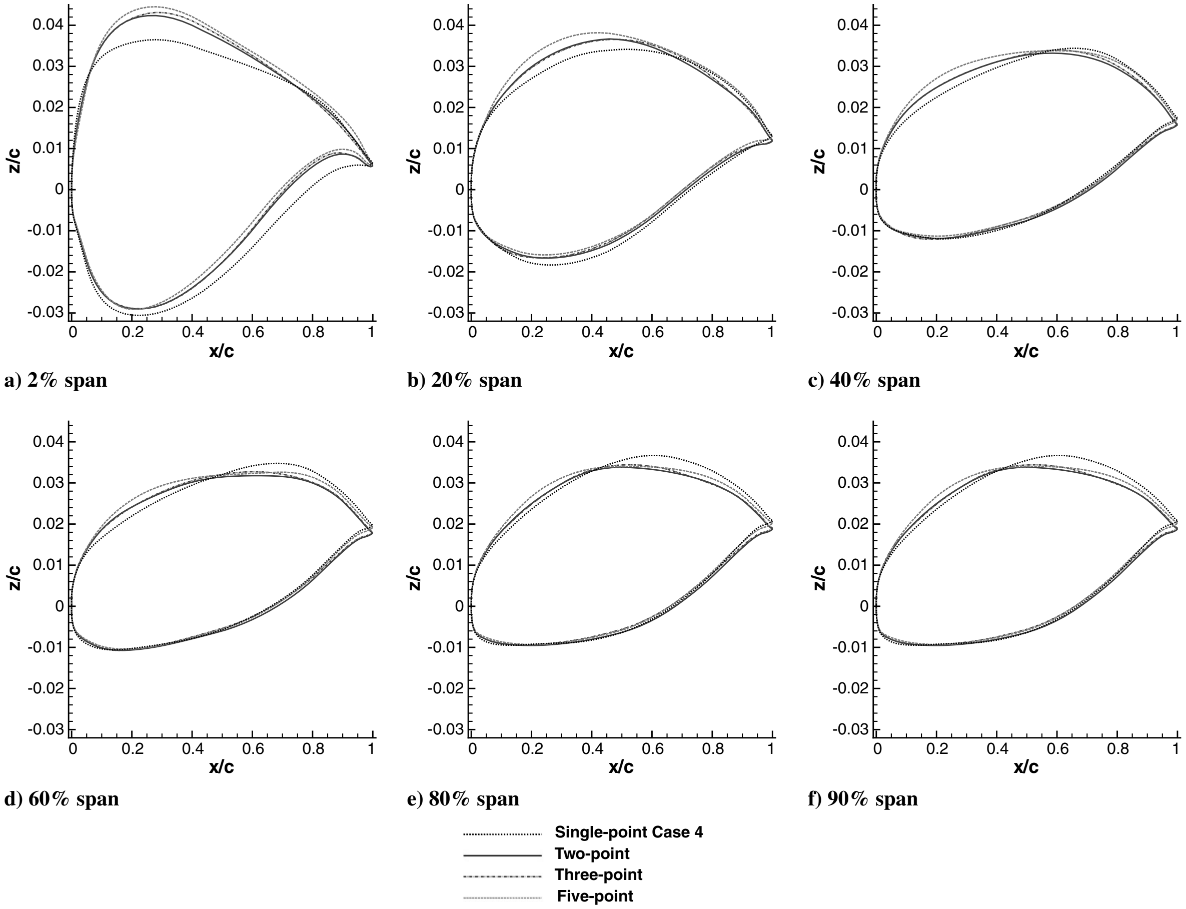
The convergence histories for the three cases are shown in Fig. 10. Note that, in each iteration, the flow solution and gradient must be computed for each operating point. In each of these three cases, the gradient L_2 norm is reduced by more than two orders of magnitude. The accuracy of the composite objective function compared to the integral I is shown in Table 4. The I values shown are obtained by numerically integrating the objective function values over the entire Mach number range with a fine interval.

Drag coefficients for the optimized wings at $C_L = 0.30$ are compared to the single-point Case 4 results in Fig. 11. With two operating conditions, the drag is substantially reduced over most of the operating range, with a slight penalty in C_D at $M = 0.90$. The composite objective function $\mathcal{J}$ differs from the integral I by 1.2%, indicating that I is not well approximated. Significant point optimization remains, as seen in Fig. 11. The three-point case continues this trend; the integral is better approximated (0.63% error), but some point optimization is still seen. Finally, with five operating points, the integral is well approximated (0.12% error), and there is no point optimization. Relative to the single-point case, the wing optimized with a five-point approximation to the integral has a drag coefficient at $M = 0.90$ that is 4.3% higher. If this is of concern to the designer, then this can be addressed by choosing a different weighting function $P(M)$. However, any increase in the weighting function at $M = 0.90$ will penalize the drag at other Mach numbers. If the wing will operate throughout the prescribed range of Mach numbers with equal probability, then the constant weighting function $P(M) = 1$ used here is appropriate.

The wing sections from the two-, three-, and five-point cases are shown in Fig. 12; the final sweep and twist angles are compared in Table 5. For practical wing design, a wider range of operating conditions, including off-design conditions, must be considered, as discussed by Buckley et al. [22]. However, the present example demonstrates that the algorithm presented is effective for multipoint optimization and well suited to practical wing design.

VI. Conclusions

An efficient parallel Newton–Krylov algorithm is presented for aerodynamic shape optimization of wings. An effective strategy for enabling simultaneous optimization of planform variables and



a) 2% span

b) 20% span

c) 40% span

d) 60% span

e) 80% span

f) 90% span

Fig. 12 Comparison of wing section shapes.

Table 5 Comparison of wing planforms

Wing	Λ_{LE}, deg	Ω, deg
ONERA M6	30.0	0.0
Single-point	35.2	−3.27
Two-point	39.4	−2.94
Three-point	39.7	−2.98
Five-point	43.4	−3.16

section shapes is shown to converge well for problems parameterized with over 200 design variables, reducing both wave and induced drag. Multipoint optimization results demonstrate that a drag integral can be minimized over a range of Mach numbers while avoiding point optimization, thus producing an efficient and robust design. Future work will involve extension to turbulent flows as well as aerostructural optimization.

Acknowledgments

This research was supported with funding through the University of Toronto, Government of Ontario, MITACS, Bombardier Aerospace, and the Canada Research Chairs Program.

References

[1] Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An Assessment of Airfoil Design by Numerical Optimization," NASA Rept. TM X-3092, July 1974.

[2] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412. doi:10.2514/3.58379

[3] Pironneau, O., *Optimal Shape Design for Elliptic Systems*, Springer–Verlag, Berlin, 1983.

[4] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285

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

[6] Jameson, A., Martinelli, L., and Pierce, N., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, Nos. 1–4, 1998, pp. 213–237. doi:10.1007/s001620050060

[7] 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. doi:10.2514/2.2413

[8] 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 2," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 61–74. doi:10.2514/2.2414

[9] Jameson, A., Sriram, S., Martinelli, L., and Haimes, B., "Aerodynamic Shape Optimization of Complete Aircraft Configurations Using Unstructured Grids," AIAA Paper 2004-0533, 2004.

[10] Leoviriakit, K., and Jameson, A., "Multipoint Wing Planform Optimization via Control Theory," AIAA Paper 2005-0450, 2005.

[11] Nielsen, E., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1155–1163. doi:10.2514/2.1765

[12] Nadarajah, S. K., Jameson, A., and Alonso, J. J., "Sonic Boom Reduction Using an Adjoint Method for Wing-Body Configurations in Supersonic Flow," AIAA Paper 2002-5547, 2002.

[13] Nadarajah, S. K., "The Discrete Adjoint Approach to Aerodynamic Shape Optimization," Ph.D. Thesis, Stanford University, Palo Alto, CA, 2003.

[14] Nemec, M., and Zingg, D. W., "Newton–Krylov Algorithm for Aerodynamic Design Using the Navier- Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1146–1154. doi:10.2514/2.1764

[15] Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.
doi:10.2514/1.10415

[16] Nemec, M., and Aftosmis, M. J., "Adjoint Algorithm for CAD-Based Shape Optimization Using a Cartesian Method," AIAA Paper 2005-4987, June 2005.

[17] Mavriplis, D., "Formulation and Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes," AIAA Paper 2005-0319, 2005.

[18] Mavriplis, D., "A Discrete Adjoint for Optimization Problems on Three-Dimensional Unstructured Meshes," AIAA Paper 2006-50, 2006.

[19] Carpentieri, G., van Tooren, M. J. L., and Koren, B., "Improving the Efficiency of Aerodynamic Shape Optimization on Unstructured Meshes," AIAA Paper 2006-298, 2006.

[20] Zingg, D. W., and Elias, S., "Aerodynamic Optimization Under a Range of Operating Conditions," *AIAA Journal*, Vol. 44, No. 11, 2006, pp. 2787–2792.
doi:10.2514/1.23658

[21] Zingg, D. W., and Billing, L., "Toward Practical Aerodynamic Design Through Numerical Optimization," AIAA Paper 2007–3950, 2007.

[22] Buckley, H., Zhou, B., and Zingg, D. W., "Airfoil Optimization Using Practical Aerodynamic Design Requirements," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1707–1719.
doi:10.2514/1.C000256

[23] Cliff, S. E., Reuther, J. J., Saunders, D. A., and Hicks, R. M., "Single-Point and Multipoint Aerodynamic Shape Optimization of High-Speed Civil Transport," *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 997–1005.
doi:10.2514/2.2886

[24] Jameson, A., Leoviriyakit, K., and Shankaran, S., "Multi-Point Aero-Structural Optimization of Wings Including Planform Variations," AIAA Paper 2007-764, 2007.

[25] Hicken, J. E., and Zingg, D. W., "Parallel Newton–Krylov Solver for the Euler Equations Discretized Using Simultaneous-Approximation Terms," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2773–2786.
doi:10.2514/1.34810

[26] Hicken, J., and Zingg, D., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 400–413.
doi:10.2514/1.44033

[27] Leung, T. M., and Zingg, D. W., "Design of Low-Sweep Wings for Maximum Range," AIAA Paper 2011-3178, June 2011.

[28] Zingg, D. W., Leung, T. M., Diosady, L., Truong, A., Elias, S., and Nemec, M., "Improvements to a Newton–Krylov Adjoint Algorithm for Aerodynamic Shape Optimization," AIAA Paper 2005-4857, Sept. 2005.

[29] Fudge, D., Zingg, D. W., and Haimes, R., "A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimization," AIAA Paper 2005-0451, Jan. 2005.

[30] Rogers, D. F., and Adams, J. A., *Mathematical Elements for Computer Graphics*, 2nd ed., McGraw–Hill, New York, 1990.

[31] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81–1259, 1981.

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

[33] Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific Computing*, Vol. 14, No. 2, 1993, pp. 461–469.
doi:10.1137/0914028

[34] Saad, Y., and Sosonkina, M., "Distributed Schur Complement Preconditioning," *SIAM Journal on Scientific Computing*, Vol. 21, No. 4, 1999, pp. 1337–1356.
doi:10.1137/S1064827597328996

[35] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer–Verlag, Berlin, 1999.

[36] Truong, A. H., Oldfield, C. A., and Zingg, D. W., "Mesh Movement for a Discrete-Adjoint Newton–Krylov Algorithm for Aerodynamic Optimization," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1695–1704.
doi:10.2514/1.33836

[37] Nielsen, E., and Park, M., "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953.
doi:10.2514/1.16052

[38] Hicken, J. E., Osusky, M., and Zingg, D. W., "Comparison of Parallel Preconditioners for a Newton–Krylov Flow Solver," *Computational Fluid Dynamics 2010: Proceedings of the Sixth International Conference on Computational Fluid Dynamics*, Springer, Berlin, 2010.

W. Anderson
Associate Editor