

Single- and Multi-Point Aerodynamic Shape Optimization Using A Parallel Newton-Krylov Approach

Timothy M. Leung* and David W. Zingg†

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

A Newton-Krylov algorithm for aerodynamic shape optimization in three dimensions is presented for both single-point and multi-point optimization. An inexact-Newton method is used to solve the Euler equations, a discrete-adjoint method to compute the gradient, and a quasi-Newton method to find the optimum. 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 a B-spline surface, and a fast algebraic algorithm is used for grid movement at each iteration. For multi-point optimization, a composite objective function is used. Optimization results are presented to demonstrate the capabilities and efficiency of the approach.

I. Introduction

In the aerodynamic design of aircraft, there have been two major breakthroughs in the design process. The first breakthrough is the development of computational fluid dynamics (CFD). In the last decade, CFD has emerged as the primary design tool for aircraft aerodynamics, complementing, and sometimes replacing wind tunnel testing. Improvements in computer hardware, especially in parallel architectures, allow engineers to solve larger and more complex CFD problems. We can now accurately compute two-dimensional flow solutions for viscous turbulent flows in less than one minute on a desktop computer, while three-dimensional flows over an entire aircraft can be solved in a matter of hours using affordable parallel computers. 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 had 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 major 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 the “wing-fuselage” configuration; they can also serve as an inexpensive numerical testbed for unconventional configurations. Beginning with the early works of Hicks *et al.*¹ and Hicks and Henne,² aerodynamic shape optimization is becoming 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 design optimization.

Many examples in wing design optimization consider a only single operating condition, such as a fixed Mach number or 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 number and lift coefficients. Trade-offs and compromises must be made between different operating conditions to achieve an overall optimum. Nemec and Zingg¹⁵ studied multi-point optimization of airfoils using a discrete-adjoint method. Zingg and Elias²² investigated automatic selection of sampling points and weights. The work is extended in Zingg and Billing²³ to include

*PhD Candidate, AIAA Student Member

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics, Associate Fellow AIAA

multiple cruise and dive conditions. Studies in multi-point wing design include those by Reuther *et al.*,^{7,8} Cliff *et al.*,²⁴ Leoviriyakit and Jameson.¹⁰ Jameson *et al.*²⁵ studied multi-point aero-structural optimization of wings.

In this paper, we present three-dimensional single-point and multi-point optimization results using a Newton-Krylov approach. Our algorithm is based on Nemec and Zingg^{14,15} and Leung and Zingg.²⁶ The steady-state flow solution is computed using the inexact-Newton method, with the flexible generalized minimal residual method (FGMRES)^{27,28} solving the linear system at each outer iteration. The optimizer is based on the quasi-Newton optimizer BFGS (Broyden-Fletcher-Goldfarb-Shanno),²⁹ and the adjoint equation is again solved by FGMRES. The surface geometry is parameterized by B-spline control surfaces, which provide the design variables. Examples are presented to demonstrate the effectiveness of the approach.

II. Problem Formulation

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

$$\min_X \mathcal{J}(X, Q) \quad (1)$$

where Q represents the flow solution. To ensure that the optimizer yields a physically realistic shape, a set of inequality constraints may be imposed. We consider only geometric constraints that are functions of design variables:

$$C_j(X) \leq 0 \quad (2)$$

In addition, the discrete flow equation must also be satisfied:

$$R(X, Q) = 0 \quad (3)$$

A. Objective Functions

For aerodynamic shape optimization problems, the objective functions are based on performance measures, such as lift and drag coefficients (C_L , C_D). We consider lift-constrained drag minimization as our objective function:

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

Target lift and drag coefficients (C_L^* , C_D^*) are supplied by the user, while weights ω_L , ω_D , also user-supplied, specify the relative importance of maintaining lift vs. drag reduction. If C_L^* is attainable and C_D^* is not, this objective function represents lift-constrained drag minimization, with the lift constraint appearing as a penalty.

For multi-point optimization with N_p operating conditions, the objective function is the weighted sum of all operating points:

$$\mathcal{J}_T = \sum_{i=1}^{N_p} \omega_i \mathcal{J}_i \quad (5)$$

where the weights are specified by the user to reflect priorities or to approximate a weighted integral, for example:

$$\mathcal{J}_T \approx \int_{M_1}^{M_2} \omega(M) \mathcal{J}(M) dM \quad (6)$$

III. Geometry Parameterization and Design Variables

We parameterize the surface of the wing using B-spline control surfaces based on Fudge *et al.*³⁰ In this method, the k -th order B-spline representation of a surface in 3D space using $N \times M$ control points and basis functions is given by:

$$\mathbf{a}(s, t) = \sum_{i=1}^M \sum_{j=1}^N \mathbf{B}_{i,j} \mathcal{M}_{i,k}(s) \mathcal{N}_{j,k}(t) \quad (7)$$

where \mathbf{a} is the position vector along the curve at parametric distances s and t from the origin, \mathbf{B}_i are the locations of the B-spline control points, and $\mathcal{M}_{j,k}(t)$ and $\mathcal{N}_{i,k}(t)$ are the basis functions of order k , defined by the Cox-deBoor relationships:³¹

$$\mathcal{N}_{i,1}(t) = \begin{cases} 1 & \text{if } d_i \leq t < d_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$\mathcal{N}_{i,k}(t) = \left[\frac{t - d_i}{d_{i+k-1} - d_i} \right] \mathcal{N}_{i,k-1}(t) + \left[\frac{d_{i+k} - t}{d_{i+k} - d_{i+1}} \right] \mathcal{N}_{i+1,k-1}(t) \quad (9)$$

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

$$d_i = \begin{cases} 0 & 1 \leq i \leq k \\ i - k & k + 1 \leq i \leq N \\ N - k + 1 & N + 1 \leq i \leq N + k \end{cases} \quad (10)$$

Before 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:

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\mathbf{D} \\ \mathbf{D} &= \mathbf{X}\mathbf{V} \end{aligned} \quad (11)$$

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

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} x_{11} & \cdots & x_{1J} \\ \vdots & & \vdots \\ x_{I1} & \cdots & x_{IJ} \end{bmatrix} & \mathbf{U} &= \begin{bmatrix} \mathcal{N}_1(s_1) & \cdots & \mathcal{N}_M(s_1) \\ \vdots & & \vdots \\ \mathcal{N}_1(s_I) & \cdots & \mathcal{N}_M(s_I) \end{bmatrix} \\ \mathbf{X} &= \begin{bmatrix} B_{11} & \cdots & B_{1N} \\ \vdots & & \vdots \\ B_{M1} & \cdots & B_{MN} \end{bmatrix} & \mathbf{V} &= \begin{bmatrix} \mathcal{M}_1(t_1) & \cdots & \mathcal{M}_1(t_J) \\ \vdots & & \vdots \\ \mathcal{M}_N(t_1) & \cdots & \mathcal{M}_N(t_J) \end{bmatrix} \end{aligned} \quad (12)$$

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

$$\begin{aligned} s_i &= \frac{i-1}{I-1}(m-k+2) \\ t_j &= \frac{j-1}{J-1}(n-k+2) \end{aligned} \quad (13)$$

The control point locations are found by first solving for \mathbf{D} , and then \mathbf{X} in the least-squares problems in (11). To generate a new surface grid, the intermediate matrix \mathbf{D} in (11) is first generated based on the new control point locations \mathbf{X} , and then the new surface grid \mathbf{A} is generated.

Fig. 1 shows the parameterization of an ONERA M6 wing. In this example, the wing is represented using cubic B-spline control points, with nine control points in the spanwise direction, and 17 in the chordwise direction. Any control point can translate in the x , y or z direction to change the surface geometry, but we use only the z -coordinates as design variables. Moving individual control points is necessary to change the wing cross-section. Increasing the number of control points improves the flexibility of the parameterization, and allows the designer more control over changes in the geometry. Furthermore, the control points can be grouped together so that they can be described by planform variables, such as semi-span ($b/2$), chord (c), sweep (Λ), dihedral (Γ) and twist (Ω).

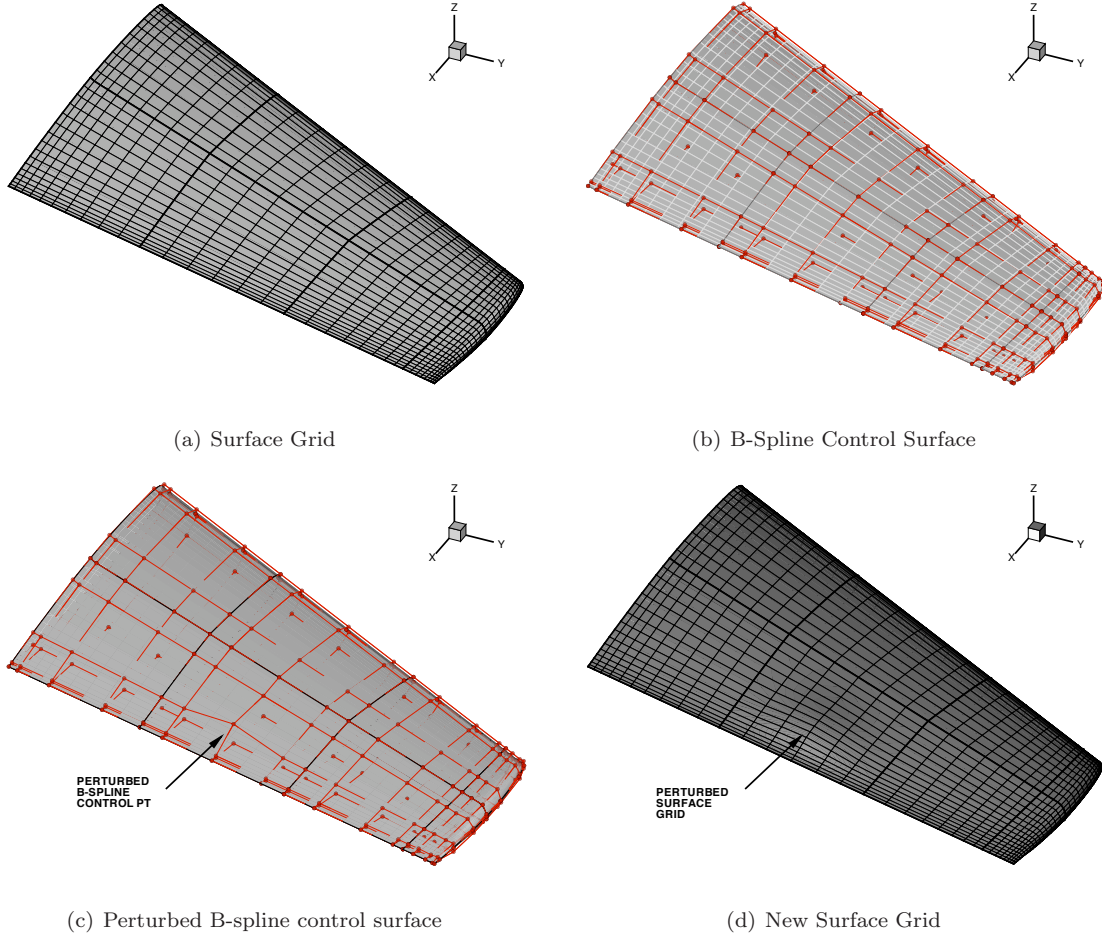


Figure 1. B-spline surface parameterization of an ONERA M6 wing with a perturbation

IV. Geometric Constraints

For geometric constraints, we have implemented a volume constraint and a thickness constraint. A volume constraint is imposed to limit the change in the volume enclosed by the wing. The constraint is expressed as a penalty term in the objective function:

$$\mathcal{J} = \mathcal{J}_0 + \mathcal{J}_{p,V} + \mathcal{J}_{p,T} \quad (14)$$

The penalty term is added when the volume V deviates from the initial volume V_0 by more than a prescribed factor v_f :

$$|V - V_0| > v_f V_0 \quad (15)$$

The volume penalty term is specified as a quadratic function:

$$\mathcal{J}_{p,V} = \frac{1}{2} \omega_V \left[\frac{|V - V_0| - v_f V_0}{V_0} \right]^2 \quad (16)$$

The penalty weight ω_V is user-supplied. For thickness constraints, we specify a minimum thickness at a fixed relative position $(x/c, 2y/b)$ on the wing. The i -th thickness constraint is expressed as:

$$t_i^* - t_i \leq 0 \quad (17)$$

where t_i is the thickness, and t_i^* is the target minimum. A penalty term is added only if t_i is lower than t_i^* :

$$\mathcal{J}_{p,i} = \left(1 - \frac{t_i}{t_i^*} \right)^2 \quad \text{if } t_i < t_i^* \quad (18)$$

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

$$\mathcal{J}_{p,T} = \omega_T \sum_i \mathcal{J}_{p,i} \quad (19)$$

By casting the constraints as penalty terms, our original optimization problem (1) can be solved using an algorithm for unconstrained optimization.

V. Flow Analysis

The governing equations for the optimization are the Euler equations. In curvilinear coordinates (ξ_1, ξ_2, ξ_3) , they can be expressed in the form:

$$\partial_t Q + \partial_{\xi_i} E_i = 0 \quad (20)$$

where

$$Q = \frac{1}{J} \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ e \end{pmatrix}, \quad \text{and} \quad E_i = \frac{1}{J} \begin{pmatrix} \rho U_i \\ \rho u_1 U_i + p \partial_x \xi_i \\ \rho u_2 U_i + p \partial_y \xi_i \\ \rho u_3 U_i + p \partial_z \xi_i \\ (e + p) U_i \end{pmatrix}.$$

The scalar J denotes the Jacobian of the mapping from physical space to computational space, and $U_i = u_j \partial_{x_j} \xi_i$ are the contravariant velocity components.

The flow equations are discretized on multi-block structured grids. In our parallel strategy, each block in the grid and the corresponding component of Q is distributed to a separate processor. The discretization of (20) is done in parallel in each block. Second-order centered differencing is used for interior nodes, and first-order one-sided differencing is used at block boundaries. For numerical stability, we use the JST scalar dissipation model,^{32,33} with second-difference dissipation near shocks and fourth-difference dissipation elsewhere. Boundary conditions and the coupling between blocks at the interfaces are done using simultaneous approximation terms (SATs). Details of the implementation of SATs can be found in Hicken and Zingg.³⁴

Discretization of the Euler equations produces a set of nonlinear algebraic equations. At steady-state, they can be represented by the equation

$$R(X, Q) = 0 \quad (21)$$

which is the same flow constraint equation in (3). In order to find this steady solution, we start with an initial guess $Q^{(0)}$ based on freestream properties and apply the Newton method, solving a linear system in the form:

$$\mathcal{A}^{(n)} \Delta Q^{(n)} = -R^{(n)} \quad (22)$$

where $\mathcal{A} = \partial R / \partial Q$ is the flow Jacobian and $R^{(n)} = R(Q^{(n)})$. The flow vector Q is updated, and the linear system is solved again until the L_2 -norm of $R(Q)$ is reduced to machine zero. Since adjoint gradient computation requires the flow Jacobian from a well-converged solution, reduction of 10 orders of magnitude or more is necessary. The linear system (22) is solved using the flexible Krylov subspace iterative method FGMRES.²⁸ When a Krylov method is used to solve the linear system, only a matrix-vector product with the flow Jacobian is required, and this can be approximated by a Frechet derivative (one-sided differencing):

$$\mathcal{A}v \approx \frac{R(Q + \epsilon v) - R(Q)}{\epsilon} \quad (23)$$

leading to a Jacobian-free approach for the flow solver. To improve the convergence of FGMRES, the linear system in (22) is right-preconditioned. The preconditioner uses a first-order Jacobian matrix (denoted by \mathcal{A}_1) that combines the second- and fourth-difference dissipation terms:

$$\hat{\kappa}_2 = \kappa_2 + \sigma \kappa_4; \quad \hat{\kappa}_4 = 0.0 \quad (24)$$

A value of $\sigma = 4.0$ is used here based on previous work.³⁵ The matrix \mathcal{A}_1 is obtained by hand linearization. The linearization of the artificial dissipation is performed by freezing the coefficients. The complex step method^{36,37} is used to linearize the SATs at block boundaries. Our approach is similar to Nielsen and Kleb,³⁸ but our application of the complex step method is limited to the SATs. An approximate-Schur

preconditioner based on Saad and Sosonkina³⁹ is applied to \mathcal{A}_1 to form the preconditioner. Details of the implementation can be found in Hicken and Zingg.³⁴

The Newton method converges quadratically when Q is sufficiently close to the solution. However, during the start-up phase, when the iterate is far from the solution, convergence may not be possible. Therefore, for stability during the start-up period, the flow solver instead uses an approximate-Newton method; the first-order Jacobian replaces the full Jacobian in (22). A pseudo-transient time step is also added during the start-up phase for globalization of the Newton method:

$$\Delta t_i^{(n)} = \frac{\Delta t_{\text{ref}}^{(n)}}{J_i(1 + \sqrt[3]{J_i})} \quad (25)$$

where the reference time step for iteration n is defined as

$$\Delta t_{\text{ref}}^{(n)} = A(B)^n \quad (26)$$

Values of $A = 0.1$ and $B = 1.5$ are used. In summary, during the approximate-Newton start-up phase, the linear system solved at each iteration n is given by:

$$\left[\mathcal{T}^{(n)} + \mathcal{A}_1^{(n)} \right] \Delta Q^{(n)} = R^{(n)} \quad (27)$$

where $\mathcal{T}^{(n)}$ is a diagonal matrix containing the reciprocal of the local time steps. The flow solver then switches to the Newton method when the normalized residual has dropped below a threshold τ , i.e.:

$$\frac{\|R^{(n)}\|_2}{\|R^{(0)}\|_2} \equiv \mathcal{R}^{(n)} < \tau \quad (28)$$

where $\tau = 0.1$ is used. During the Newton phase, the reference time step is based on Mulder and van Leer:⁴⁰

$$\Delta t_{\text{ref}}^{(n)} = \max \left[\alpha \left(\mathcal{R}^{(n)} \right)^{-\beta}, \Delta t_{\text{ref}}^{(n-1)} \right] \quad (29)$$

We find $\alpha = 1.0$ and $\beta = 1.75$ found to be satisfactory values for a wide range of problems. As the residual decreases, the time step approaches infinity, and the full Newton step is recovered.

VI. Optimizer

When the geometric constraints are implemented as quadratic penalty terms, we can now consider the optimization as an unconstrained problem. A quasi-Newton optimizer BFGS is used. In BFGS, the inverse of the Hessian matrix is approximated to get the search direction. BFGS guarantees that the search direction is a descent direction.

A. 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:

$$\mathcal{G} = \nabla \mathcal{J} = \frac{\partial \mathcal{J}}{\partial X} - \psi^T \frac{\partial R}{\partial X} \quad (30)$$

where the vector ψ is the adjoint variable, which is obtained by solving the adjoint equation:

$$\mathcal{A}^T \psi = \frac{\partial \mathcal{J}^T}{\partial Q} \quad (31)$$

It should be noted that the size of (31) is independent of the number of design variables. We solve the adjoint system by adopting the same strategy used for the flow solution. The adjoint system is solved using FGMRES as the iterative solver. The system is right-preconditioned using the approximate-Schur preconditioner. We specify a linear tolerance of 10^{-8} for the adjoint system. Note that the left-hand side

of the adjoint equation differs from (22) by a transpose. As a result, the Jacobian-free approach used in the flow solver cannot be used, and the matrix must be formed explicitly. The right-hand-side term $\partial\mathcal{J}/\partial Q$ is evaluated analytically for each objective function. Finally, the partial derivatives with respect to design variables $\partial\mathcal{J}/\partial X$, $\partial R/\partial X$ are evaluated using second-order centered differencing. For the k -th design variable:

$$\begin{aligned}\frac{\partial R}{\partial X_k} &\approx \frac{R[X + h\hat{e}_k, Q] - R[X - h\hat{e}_k, Q]}{2h} \\ \frac{\partial \mathcal{J}}{\partial X_k} &\approx \frac{\mathcal{J}[X + h\hat{e}_k, Q] - \mathcal{J}[X - h\hat{e}_k, Q]}{2h}\end{aligned}\quad (32)$$

where \hat{e}_k is the k -th unit vector, and the step size is given by:

$$h = \max(\epsilon \cdot |X_k|, 1 \times 10^{-4}) \quad (33)$$

B. Line Search

A line-search algorithm with backtracking is used to satisfy the strong Wolfe conditions. In the line-search algorithm, a cubic interpolant is constructed for the function

$$\phi(\beta) \equiv \mathcal{J}(X_n + \beta p_n) \quad (34)$$

where p_n is the search direction, and β is the step size along that direction. The minimum of the interpolant is either at its endpoint or in the interior, which can be found by the line-search iteration until the strong Wolfe conditions²⁹ are satisfied:

$$\beta_{k+1} = \beta_k - (\beta_k - \beta_{k-1}) \left[\frac{\phi'(\beta_k) + r_2 - r_1}{\phi'(\beta_k) - \phi'(\beta_{k-1}) + 2r_2} \right] \quad (35)$$

where

$$\begin{aligned}r_1 &= \phi'(\beta_{k-1}) + \phi'(\beta_k) - 3 \frac{\phi(\beta_{k-1}) - \phi(\beta_k)}{\beta_{k-1} - \beta_k} \\ r_2 &= \sqrt{r_1^2 - \phi'(\beta_{k-1})\phi'(\beta_k)}\end{aligned}\quad (36)$$

The line-search algorithm is considered stalled if a satisfactory step size cannot be found in 15 iterations, and the optimizer is re-started from the steepest descent direction $-\mathcal{G}$. The optimizer is considered converged to a local minimum when $\|\mathcal{G}\|_2$ falls below a user-defined tolerance. However, for many applications, it is often impractical to reduce the gradient by more than one or two orders of magnitude.

C. 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 other variables. This problem arises when we mix B-spline design variables with planform variables. In our experience, ΔX_i from initial to optimized geometry ranges from 10^{-5} for B-spline variables, to 10^0 for some planform variables (e.g. change in sweep angle, measured in radians). To help the convergence of the optimizer, we scale the design variables by the square root of their initial values:⁴¹

$$X_s = L^{-1}X \quad (37)$$

where the entries in X_s are the scaled design variables, those in X are the unscaled design variables, and the diagonal matrix L contains the square root of the initial unscaled design variables.

D. Grid Movement Strategy

A high-quality computational grid is necessary to compute the flow solution at each design iteration, and also to evaluate the partial derivatives $\partial\mathcal{J}/\partial X$, $\partial R/\partial X$ in (30). For aerodynamic shape optimization using body-fitted grids, expensive grid re-generation 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, a fast algebraic

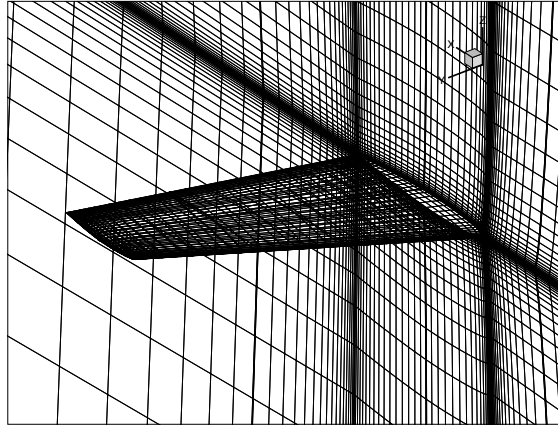


Figure 2. H-H grid over the ONERA M6 wing.

grid movement method is used. The movement of the nodes $k = 2$ to k_{max} along a grid line is determined by the algebraic equation:

$$\mathbf{x}_k^{\text{new}} = \mathbf{x}_k^{\text{old}} + \frac{\Delta \mathbf{x}_1}{2} [1 + \cos(\pi S_k)] \quad (38)$$

where $\Delta \mathbf{x}_1$ is the displacement of the surface node, and

$$S_k = \frac{\sum_{i=2}^k |\mathbf{x}_i - \mathbf{x}_{i-1}|}{\sum_{i=2}^{k_{max}} |\mathbf{x}_i - \mathbf{x}_{i-1}|} \quad (39)$$

is the normalized arclength distance along the grid line.

VII. Results and Discussion

We obtain flow solver and optimization results using a distributed-memory Beowulf-class cluster. The cluster uses Intel Itanium 2 processors with a CPU speed of 1.5GHz. Each node consists of 4 processors, with 8GB of shared memory per node. The computational nodes are connected by a high-bandwidth low-latency Myrinet network. Communication between processors is done using the message passing library MPICH.

A. Single-Point Wing Optimization

We present results from a single-point optimization of a wing at a transonic speed. The goal of this optimization case is to minimize drag at $M = 0.90$ while maintaining the lift coefficient of the original geometry. For this case, a 431,000-node grid over an ONERA M6 wing, shown in Fig. 2, is used. The grid has an H-H topology with 48 blocks. The ONERA M6 wing initially operates at $\alpha = 2.50^\circ$. At this operating condition, the lift and drag coefficients are $C_L = 0.307$ and $C_D = 0.0281$ respectively. The lift-constrained drag minimization objective function (4) is used, with targets and weights in lift and drag:

$$\begin{aligned} C_L^* &= 0.308 & \omega_L &= 100.0 \\ C_D^* &= 0.0075 & \omega_D &= 1.0 \end{aligned}$$

The high value of ω_L is found to be necessary for the optimizer to maintain the lift coefficient. Our design variables are the z -coordinates of 165 B-spline control points, which includes all control points except near the leading edge, trailing edge and wing tip, angle of attack (α), the changes in leading-edge sweep angle ($\Delta \Lambda_{LE}$) at the wing root and at the 14th and 15th B-spline section near the wing tip, as well as the twist angle (Ω), for a total of 170 design variables. The change in sweep angle is measured as the change from the original ONERA M6 wing. As Λ_{LE} changes, the wing is sheared along the chordwise direction, thus the planform area remains constant throughout the optimization cycle. A volume constraint (16) with $v_f = 0.0$ and a penalty weight of $\omega_V = 50.0$ was added to maintain the wing's volume.

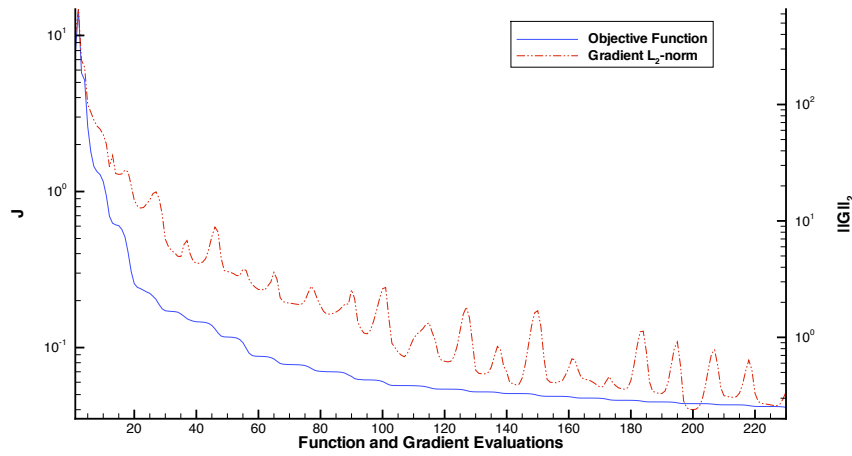


Figure 3. Convergence history for the single-point optimization case

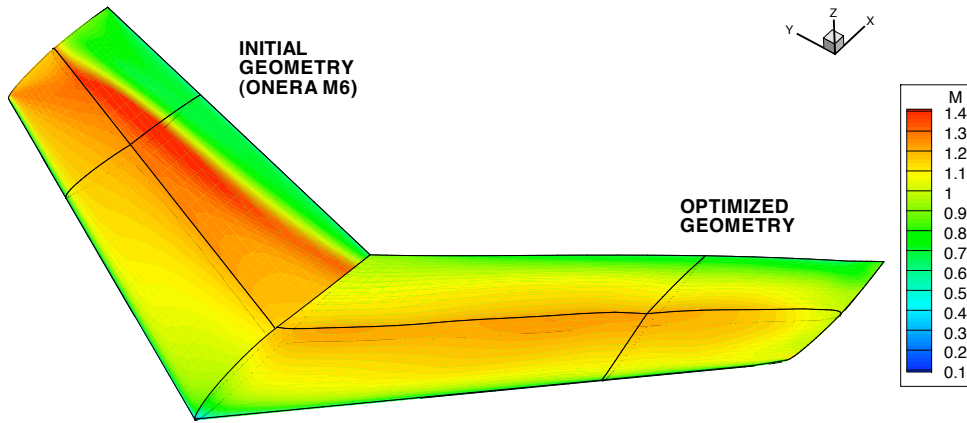


Figure 4. Comparison of Mach contours for single-point optimization case

The optimization convergence history is shown in Fig. 3. After 231 iterations, the objective function and the gradient L_2 -norm are reduced by about three orders of magnitude. At this point, the lift and drag values are:

$$C_L = 0.307$$

$$C_D = 0.00894$$

This represents a 68% reduction in drag. In the final geometry, the leading-edge sweep angle Λ_{LE} is increased from 30.0° to 43.7° . The angle of attack increases to $\alpha = 2.62^\circ$. A twist (washout) angle of $\Omega = -6.07^\circ$ was added. Surface Mach number contours of the optimized wing are compared to the ONERA M6 wing in Fig. 4, and pressure coefficients and wing sections of the optimized wing is plotted against the ONERA M6 wing at six spanwise stations in Fig. 5. Both Figs. 4 and 5 show that the optimizer has successfully eliminated the wave drag by removing the shock.

During the optimization cycle, a flow solve requires an average of 8.2 minutes to reduce the residual by 10 orders of magnitude, using 48 processors. The adjoint solver takes an average of 2.2 minutes to reduce the residual by eight orders of magnitude. That the flow solution takes about four times longer to solve than the adjoint solution is consistent with previous experience with our 2D adjoint solver. The entire optimization cycle (231 iterations) took 44 hours to complete. Table 1 shows that most of the drag reduction occurs in the first 20 iterations, which took about four hours.

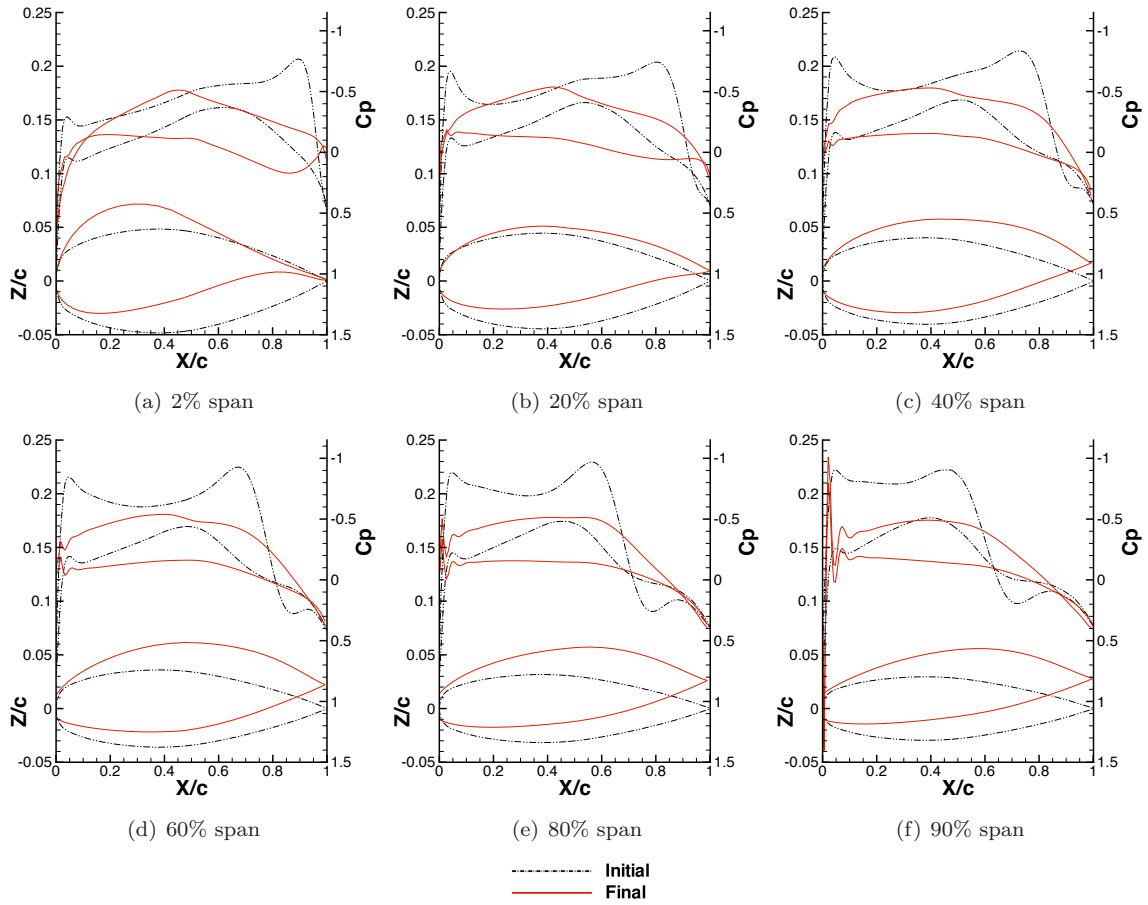


Figure 5. Comparison of pressure coefficients and wing sections at various spanwise stations for single-point optimization case

B. Multi-Point Wing Optimization

A well-known issue with single-point optimization is that the optimized wing will typically perform well only at the specified operating condition. In the single-point example presented in above, the optimizer was able to significantly decrease the drag at the specified cruise condition at $M = 0.90$. Compared to the original ONERA M6 wing, the performance improvement is significant over all transonic speeds, as shown in the drag divergence plots in Fig. 6a. However, when the performance of the optimized wing is examined more closely, in Fig. 6b, it is noted that drag increases rapidly at speeds both lower and higher than $M = 0.90$, which could be undesirable. While at the specified cruise condition the wing is essentially shock free, shocks develop rapidly away from the optimized condition, when the Mach number is above or below 0.90. In multi-point optimization, we seek to reduce the drag of the wing over a wider range of Mach numbers, in this case, between $0.70 < M < 0.90$, at a fixed lift coefficient C_L . We consider a three-point optimization case using (5). The selected operating points are: $M_1 = 0.70$, $M_2 = 0.88$ and $M_3 = 0.90$. We have arbitrarily weighted each operating point equally ($\omega_i = 1.0$), but the weighting may change at the discretion of the designer. The lift-constrained drag minimization objective function is used for each operating point, and the targets (C_L^* , C_D^*) and weights (ω_L , ω_D) are re-used from the single-point optimization case. The multi-point optimization begins with the final geometry from the single-point case.

The convergence history for the three-point optimization case is shown in Figure 7 after 78 iterations, which took 52 hours using 48 processors. The number of iterations required for this multi-point case is fewer than the single-point case because the initial geometry is aerodynamically more efficient than the ONERA M6 wing. Note that each iteration involves solving one flow solution and one adjoint solution *for each operating point*. The composite objective function and gradient L_2 -norm have both decreased by about three orders of magnitude. In the final geometry, the leading-edge sweep angle is slightly increased

Iteration	C_D	Drag Reduction
1	0.02804	—
20	0.01121	60%
50	0.01002	64%
100	0.00928	67%

Table 1. Drag reduction vs. iteration count for single-point optimization

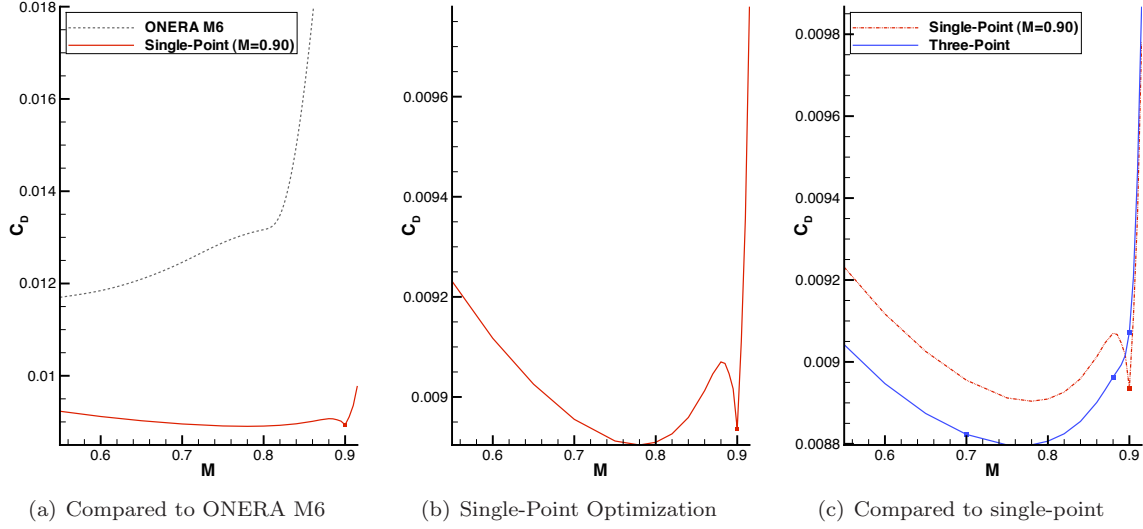


Figure 6. Drag divergence plot for the optimized wing at $C_L = 0.307$

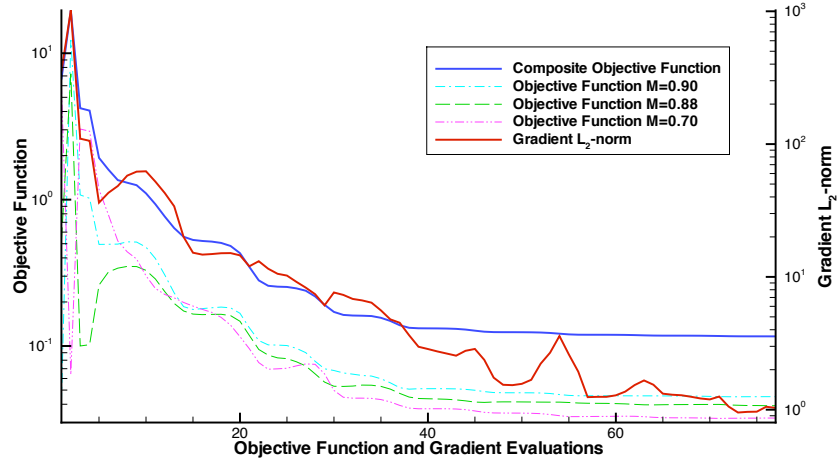


Figure 7. Convergence history for the three-point optimization case

to $\Lambda_{LE} = 44.3^\circ$, compared to 43.7° in the single-point case. The final twist angle is also slightly reduced to $\Omega = -5.32^\circ$. The operating angles of attack are $\alpha = 1.37^\circ$, 1.38° and 2.01° for the operating Mach numbers of $M = 0.70$, 0.88 and 0.90 respectively. The wing cross sections at six spanwise locations are compared to the single-point results in Figure 8. The final shape is very similar to the single-point results. The final geometry is shock-free at both $M = 0.70$ and $M = 0.88$, but a weak shock is present at $M = 0.90$.

Drag coefficients of the optimized wing over a range of Mach numbers at fixed $C_L = 0.307$ are shown in Fig 6c. As expected, drag coefficients at $M = 0.70$ and 0.88 have been decreased, while that at $M = 0.90$

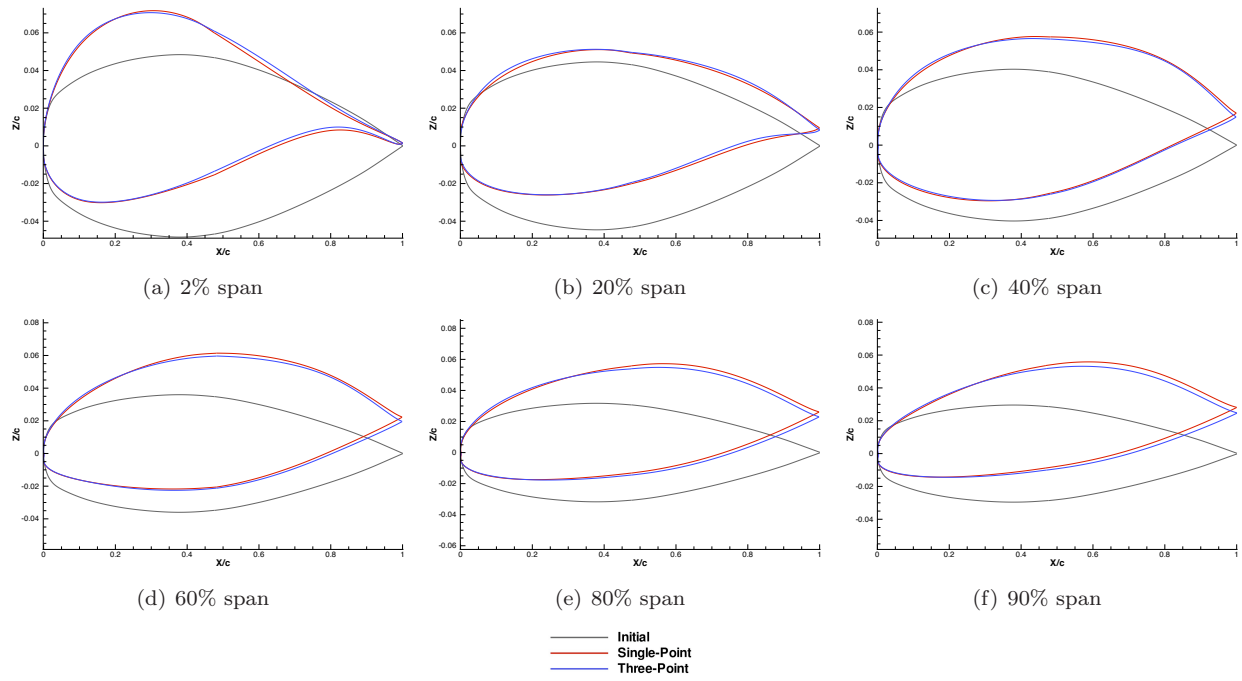


Figure 8. Comparison of wing section shapes

has increased. The figure clearly shows that the mean drag over the range of Mach numbers 0.70 to 0.90 has substantially decreased.

C. Effect of the Initial Geometry

The final design example studies whether multiple local optima exist. Hicken and Zingg⁴² have shown in the design of wingtip devices that multiple local optima can exist. The focus of this design example is on the sweep angle of the wing at transonic speeds.

The goal of this optimization is to find the shape which minimizes drag at an operating Mach number of $M = 0.76$ and a lift coefficient of $C_L = 0.40$. We use the following targets and weight on lift and drag in (4):

$$\begin{aligned} C_L^* &= 0.402 & \omega_L &= 50.0 \\ C_D^* &= 0.00610 & \omega_D &= 1.0 \end{aligned}$$

We perform two optimization cases using two initial geometries. The first is a tapered wing with the following planform parameters:

$$\begin{aligned} \text{Root chord } c_{\text{root}} &= 0.667 & \text{Planform area } S &= 0.967 \\ \text{Semi-span } b/2 &= 2.00 & \text{LE sweep angle } \Lambda_{LE} &= 5.2^\circ \\ \text{Taper ratio } \gamma &= 0.55 & \text{TE sweep angle } \Lambda_{TE} &= -5.2^\circ \\ \text{Aspect ratio } AR &= 8.27 \end{aligned}$$

The original wing has an cross section shape that is constant from wing root to wing tip. The second geometry has the same wing cross section as the first case, but the wing has an initial forward sweep angle of $\Lambda_{LE} = -6.3^\circ$ at the leading edge.

The volume grids around the two wings are 12-block H-H topology grids with 320,292 nodes. On the top and bottom surfaces, there are 31 and 41 nodes on the chordwise and spanwise directions respectively. The two initial geometries and their respective grids are shown in Fig. 9. Both the top and bottom surfaces of each wing are parameterized with a cubic B-spline surface with 12 control points in the spanwise direction and 11 in the chordwise direction. Every B-spline control point is a design variable, except at the leading and trailing edges, where the control points are fixed. There are 230 B-spline control point design variables

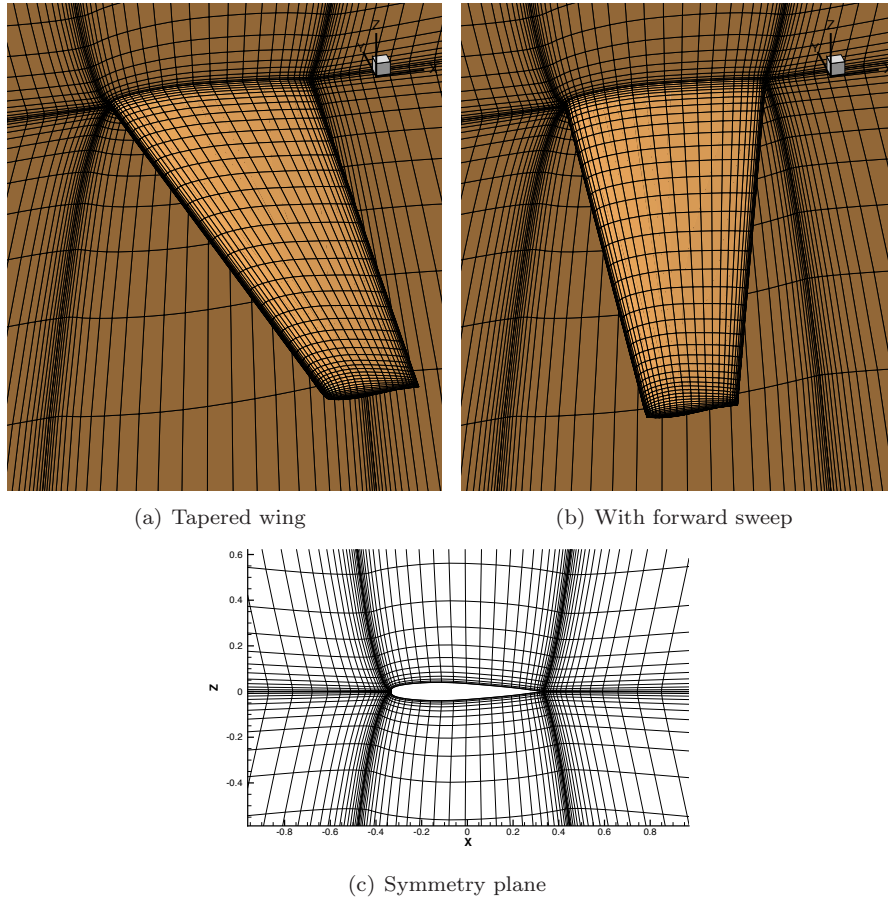


Figure 9. 12-block grid around tapered wings used for the design examples

Initial Geometry	Final C_L	Final C_D	Final Λ	Final Ω
Tapered wing	0.400	0.00671	28.6°	-3.60°
With forward sweep	0.399	0.00902	-29.4°	-0.60°

Table 2. Final geometries and aerodynamic performances

in total. In addition, the change in the wing's leading-edge sweep angle, twist and angle of attack are also design variables.

A volume constraint is used to maintain the volume enclosed by the wings, with the parameter $v_f = 0.0$ and penalty weight $\omega_V = 50.0$ used. Thickness constraints are also imposed near the leading edge and trailing edge to prevent grid cross-over:

$$\begin{aligned}
 (t/c)^* &= 0.04 & \text{at } (x/c) &= 0.02 \\
 (t/c)^* &= 0.002 & \text{at } (x/c) &= 0.97
 \end{aligned}$$

A penalty weight of $\omega_T = 50.0$ is used.

For both optimization cases, the objective function and gradient L_2 -norm have decreased by more than three orders of magnitude. In the final geometries, both volume and thickness constraints are active. We believe that both cases have converged to a local optimum. However, their final geometries and their performance are very different, as shown in Table 2. For the tapered wing, a swept-back configuration is obtained by the optimizer with $\Lambda = 28.6^\circ$, while for the forward-swept wing, the optimizer swept the wing further forward to $\Lambda = -29.4^\circ$. The final wings from both cases are compared to their respective initial geometries in Fig. 10. They are shown with Mach contours on the wing surface. In both cases, the shock on

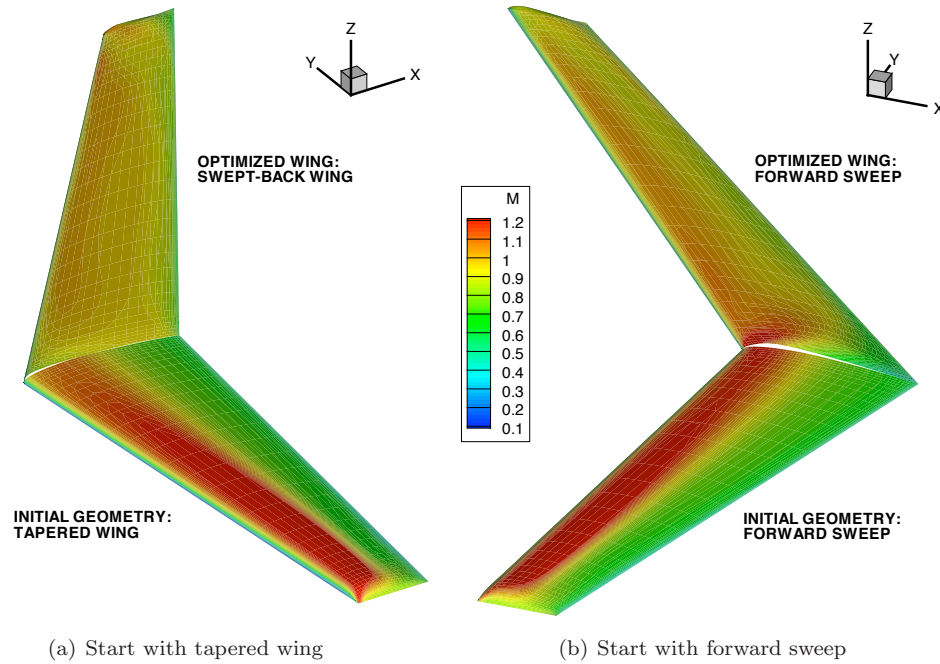


Figure 10. Initial and final geometries for the last case

the top surfaces has been eliminated in the optimized geometries. This can be seen clearly in the pressure coefficient plots in Fig. 11.

VIII. Conclusion

An efficient parallel Newton-Krylov algorithm is presented for single- and multi-point optimization of wings in three-dimensions. We have demonstrated that the algorithm is able to efficiently optimize both the planform and cross section of a wing with 170 design variables. In our multi-point optimization results, we have shown that we can improve the mean drag over a range of operating Mach numbers. Future work will involve extension to viscous and turbulent flows, as well as optimization of wing-body configurations.

Acknowledgments

This research was supported with funding through the University of Toronto, Kenneth M. Molson Fellowship, Ontario Graduate Scholarship for Science and Technology (OGSST) from the Government of Ontario, MITACS, Bombardier Aerospace and the Canada Research Chairs Program. The authors gratefully acknowledge Jason Hicken for his assistance and discussions.

References

- ¹Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An assessment of airfoil design by numerical optimization," NASA TM X-3092, NASA, July 1974.
- ²Hicks, R. M. and Henne, P. A., "Wing design by numerical optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.
- ³Pironneau, O., *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, 1983.
- ⁴Jameson, A., "Aerodynamic design via control theory," *J. of Sci. Comp.*, Vol. 3, 1988, pp. 233–260.
- ⁵Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation," AIAA Paper 97–643, 1997.
- ⁶Jameson, A., Martinelli, L., and Pierce, N., "Optimum aerodynamic design using the Navier-Stokes equations," *Theoretical Fluid Dynamics*, Vol. 10, 1998, pp. 213–237.
- ⁷Reuther, 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.

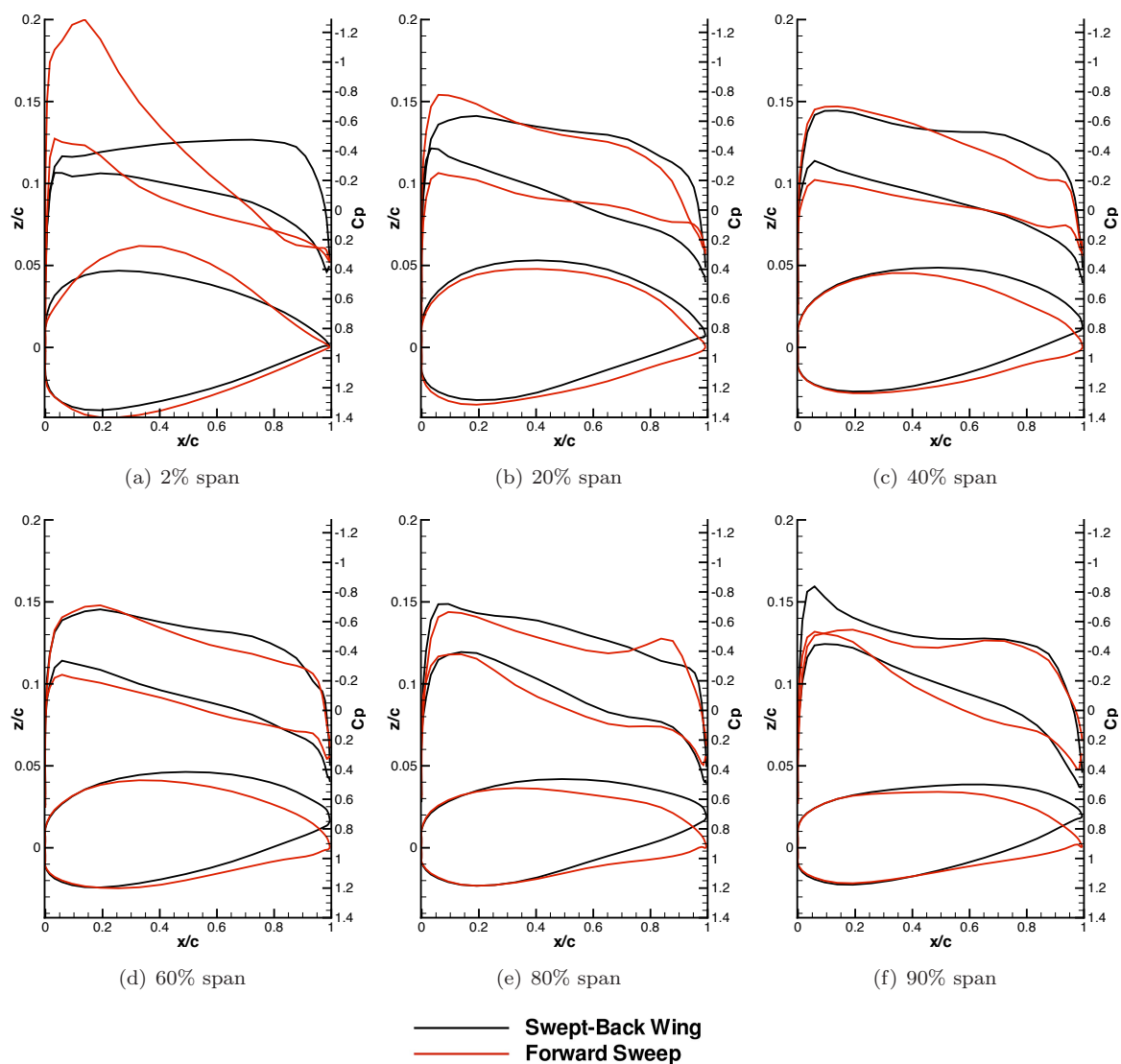


Figure 11. Pressure coefficients and wing sections for both initial geometries

⁸Reuther, 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.

⁹Jameson, A., Sririam, Martinelli, L., and Haimes, B., “Aerodynamic shape optimization of complete aircraft configurations using an unstructured grids,” AIAA Paper 2004–533, 2004.

¹⁰Leoviriakit, K. and Jameson, A., “Multi-point wing planform optimization via control theory,” AIAA Paper 2005–450, Reno, NV, 2005.

¹¹Nielsen, E. J. and Anderson, W. K., “Recent improvements in aerodynamic design optimization on unstructured meshes,” *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1155–1163.

¹²Nadarajah, 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.

¹³Nadarajah, S. K., *The Discrete Adjoint Approach to Aerodynamic Shape Optimization*, Ph.D. thesis, Stanford University, 2003.

¹⁴Nemec, 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.

¹⁵Nemec, 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.

¹⁶Nemec, M. and Aftosmis, M., “Adjoint algorithm for CAD-Based shape optimization using a Cartesian method,” AIAA Paper 2005–4987, June 2005.

- ¹⁷Mavriplis, D., "Formulation and multigrid solution of the discrete adjoint for optimization problems on instructed meshes," AIAA Paper 2005-319, 2005.
- ¹⁸Mavriplis, D., "A discrete adjoint for optimization problems on three-dimensional unstructured meshes," AIAA Paper 2006-50, 2006.
- ¹⁹Carpentieri, G., van Tooren, M. J. L., and Koren, B., "Improving the efficiency of aerodynamic shape optimization on unstructured meshes," AIAA Paper 2006-298, Reno, NV, 2006.
- ²⁰Nadarajah, S. and Jameson, A., "A Comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization," AIAA Paper 2000-667, 2000.
- ²¹Giles, M. B. and Pierce, N. A., "An introduction to the adjoint approach design," *Flow, Turbulence and Combustion*, Vol. 65, 2000, pp. 393-415.
- ²²Zingg, D. W. and Elias, S., "Aerodynamic Optimization Under a Range of Operating Conditions," *AIAA Journal*, Vol. 44, No. 11, 2006, pp. 2787-2792.
- ²³Zingg, D. W. and Billing, L., "Toward Practical Aerodynamic Design Through Numerical Optimization," AIAA Paper 2007-3950, 2007.
- ²⁴Cliff, 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.
- ²⁵Jameson, A., Leoviriyakit, K., and Shankaran, S., "Multi-Point Aero-Structural Optimization of Wings Including Planform Variations," AIAA Paper 2007-764, 2007.
- ²⁶Leung, T. M. and Zingg, D. W., "A Newton-Krylov Approach for Aerodynamic Shape Optimization of Wings," AIAA Paper 2008-5806, American Institute of Aeronautics and Astronautics, September 2008.
- ²⁷Saad, Y. and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear problems," *SIAM J. Sci. Stat. Comp.*, Vol. 7, 1986, pp. 856-869.
- ²⁸Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific Computing*, Vol. 14, No. 2, 1993, pp. 461-469.
- ²⁹Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer-Verlag, 1999.
- ³⁰Fudge, 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, January 2005.
- ³¹Rogers, D. F. and Adams, J. A., *Mathematical Elements for Computer Graphics*, McGraw-Hill, 2nd ed., 1990.
- ³²Jameson, 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, 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA, 1981.
- ³³Pulliam, T. H., "Efficient solution methods for the Navier-Stokes equations," Lecture notes for the von Kármán inst. for fluid dynamics lecture series: Numerical techniques for viscous flow computation in turbomachinery bladings, Brussels, Belgium, Jan. 1986.
- ³⁴Hicken, 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.
- ³⁵Nichols, J. and Zingg, D. W., "A three-dimensional multi-block Newton-Krylov flow solver for the Euler equations," AIAA Paper 2005-5230, 2005.
- ³⁶Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivative of Real Functions," *SIAM Review*, Vol. 40, No. 1, March 1998.
- ³⁷Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen, E. J., "Sensitivity Analysis for the Navier-Stokes Equations on Unstructured Meshes Using Complex Variables," AIAA Paper 99-3294, 1999.
- ³⁸Nielsen, E. J. and Kleb, B., "Efficient construction of discrete adjoint operators on unstructured grids by using complex variables," AIAA Paper 2005-324, Reno, NV, 2005.
- ³⁹Saad, Y. and Sosonkina, M., "Distributed Schur complement techniques for general sparse linear systems," *SIAM Journal of Scientific Computing*, Vol. 21, No. 4, 1999, pp. 1337-1357.
- ⁴⁰Mulder, W. A. and van Leer, B., "Experiments with implicit upwind methods for the Euler equations," *Journal of Computational Physics*, Vol. 59, 1985, pp. 232-246.
- ⁴¹Zingg, 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, American Institute of Aeronautics and Astronautics, September 2005.
- ⁴²Hicken, J. E. and Zingg, D. W., "An Investigation of Induced Drag Minimization Using a Newton-Krylov Algorithm," AIAA Report 2008-5807, American Institute for Aeronautics and Astronautics, Victoria, B.C., 2008.