Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation

W. Kyle Anderson
NASA Langley Research Center
Hampton, Virginia, 23681-0001

V. Venkatakrishnan*
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Abstract

A continuous adjoint approach for obtaining sensitivity derivatives on unstructured grids is developed and analyzed. The derivation of the costate equations is presented, and a second-order accurate discretization method is described. The relationship between the continuous formulation and a discrete formulation is explored for inviscid, as well as for viscous flow. Several limitations in a strict adherence to the continuous approach are uncovered, and an approach that circumvents these difficulties is presented. The issue of grid sensitivities, which do not arise naturally in the continuous formulation, is investigated and is observed to be of importance when dealing with geometric singularities. A method is described for modifying inviscid and viscous meshes during the design cycle to accommodate changes in the surface shape. The accuracy of the sensitivity derivatives is established by comparing with finite-difference gradients and several design examples are presented.

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), Mail stop 403, 6 North Dryden Street, NASA Langley Research Center, Hampton, Va. 23681-0001

Introduction

Aerodynamic design optimization has been an important area of research for many years. Although some of the early work in this area has been limited in applicability because of a lack of computational tools, advances in computational algorithms and computer hardware have recently fostered intense efforts aimed at aerodynamic and multidisciplinary optimization. Among the methods currently used are gradient-based optimizers in which a specified objective function is minimized. The gradients of the objective function with respect to the design variables are used to update the design variables in order to systematically reduce the cost function to arrive at a local minimum. An important step in this process is the determination of these gradients, which are also referred to as sensitivity derivatives.

Several techniques have been investigated for evaluating the sensitivities for aerodynamic applications. A description of these techniques can be found in Refs. 5, 15, 16, and in the references contained therein. Of particular interest in the present context are adjoint methods. In these methods, the objective function is augmented with the flow equations enforced as constraints through the use of Lagrange multipliers. These methods are particularly suited to aerodynamic design optimization for which the number of design variables is large in relation to the number of aerodynamic constraints or to the number of objective functions in a multipoint design. This is because the derivatives with respect to all design variables for each objective function or aerodynamic constraint, can be obtained with a computational effort roughly equivalent to that for a single solution of the flow equations.

Adjoint methods can generally be divided into discrete and continuous adjoint methods. In the discrete adjoint approach, the augmented cost function is discretized before variations are taken. For the continuous adjoint formulation, the process is reversed: variations are performed first, followed by the discretization. Note that the operations of differentiation and discretization do not commute in general. Hence, derivatives obtained by using the two approaches may not be identical and would differ according to the level of truncation error. A comparison of these two approaches for a quasi-one-dimensional problem is given in Ref. 37.

Much of the pioneering theoretical work in adjoint methodology has been presented in Refs. 19, 25, 29, 30, and 31. Although optimality conditions for aerodynamic applications have been derived from a continuous approach in Refs. 3 and 6, the computer implementations have generally followed the discrete approach. One of the advantages of the discrete adjoint approach is that, because the equations are discretely adjoint to the flow equations, the derivatives obtained are consistent with finite-difference gradients independent of the mesh size. A disadvantage of this approach is that it requires the transpose of the matrix that represents the linearization of the discrete residual with respect to the flow variables. For higher order accurate schemes, where the residual has a complex dependence on grid points, an exact implementation of this approach may be difficult to realize. For this reason, previous implementations of the discrete adjoint approach, such as those in Refs. 6, 7, 26, and 27, have used a discretization of the adjoint equations that is consistent with a first-order accurate discretization of the flow equations. Second-order accurate implementations of the discrete adjoint approach have been carried out on structured grids

in Refs. 10 and 23. On unstructured grids, a discrete adjoint approach for the Euler equations that is consistent with a second-order discretization of the flow equations has recently been implemented.[13]

In Ref. 19, Jameson developed a control theory framework for optimization using both the full potential and Euler equations for compressible flows. Computational results based on this approach were first presented in Ref. 20. This approach has been further developed and implemented for both two- and three-dimensional applications.[21,22] In these references, the continuous adjoint approach is pursued in both the derivation and the implementation on structured grids. In Refs. 32 and 33, the technique has been applied on complex configurations with a multiblock algorithm.

The continuous adjoint approach has also been considered by Iollo et al.[17] and Iollo and Salas[18] for both one-dimensional flow and two-dimensional flows over simple geometries. Kuruvila, Ta'asan, and Salas[24] and Ta'asan and Kuruvila[38] have investigated an efficient "one-shot" approach in which the design variables are updated in a hierarchical manner. Cabuk and Modi[11] and Cabuk et al.[12] have also used an adjoint formulation to design an optimal diffuser shape using the incompressible Navier-Stokes equations.

In this paper, the problem of aerodynamic optimization on unstructured grids via a continuous adjoint approach is developed and analyzed for inviscid and viscous flows. A detailed discretization of the adjoint equations is presented, and the relationship with the discrete adjoint approach is investigated. The accuracy of the resulting derivatives is assessed by comparison with finite-difference gradients. In addition, a mesh movement scheme is presented for restructuring the grid in response to changes in the surface geometry. The resulting methodology is then used to design several airfoils for inviscid compressible flow, as well as for incompressible laminar flow.

Adjoint Variable Approach for Sensitivity Derivatives

Considering first steady inviscid compressible flow, the governing equations are given by:

$$\frac{\partial}{\partial x}\mathbf{F}(\mathbf{Q}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{Q}) = 0 \tag{1}$$

where $\mathbf{Q}$ is the set of dependent variables for the Euler equations $(\rho, \rho u, \rho v, E)$, $\mathbf{F}$ and $\mathbf{G}$ represent the flux vectors of mass, momentum, and energy, and x and y are Cartesian coordinates.

In the adjoint approach for design optimization, a cost function is defined and augmented with the flow equations as constraints:

$$I(\mathbf{Q}, \mathbf{D}, \Psi) = I_c(\mathbf{Q}, \mathbf{D}) + \int_\Omega (\Psi, \mathbf{R})d\Omega = I_c(\mathbf{Q}, \mathbf{D}) + I_R(\mathbf{Q}, \mathbf{D}, \Psi) \tag{2}$$

where $\mathbf{R}$ represents the steady-state flow equations, $\mathbf{D}$ is the vector of design variables, and Ψ are the Lagrange multipliers (also referred to as the costate or adjoint variables). In Eq. (2), $I_c(\mathbf{Q}, \mathbf{D})$ represents the cost that is to be minimized, and $\int_\Omega (\Psi, \mathbf{R})d\Omega$ is the inner product of the costate variables with the residual. Examples of suitable cost func-

tions include drag minimization and matching a specified pressure distribution, for which $I_c(\mathbf{Q}, \mathbf{D})$ can be written as

$$I_c(\mathbf{Q}, \mathbf{D}) = \oint_\Gamma (c_p k_x \cos\alpha - c_p k_y \sin\alpha) ds \quad \text{Drag minimization} \tag{3a}$$

$$I_c(\mathbf{Q}, \mathbf{D}) = \frac{1}{2}\oint_\Gamma (c_p - c_p^*)^2 ds \quad \text{Specified pressure distribution} \tag{3b}$$

where c_p is the pressure coefficient, k_x and k_y are x and y components of a unit normal to the surface, and α is the angle of attack. The cost function can also involve field integrals, such as viscous dissipation, although these are not considered in this paper. It is assumed that the cost functions are differentiable although this assumption may not be valid for flows with shock waves or other singularities. A smoothing procedure as suggested in Refs. 19 and 22 may be employed to place the derivation on firmer theoretical ground. However, in numerical implementations, dissipation typically smears discontinuities over a finite number of mesh points, thus mitigating the effects of non-differentiability. Therefore, smoothing of the cost function is not performed in this paper with no apparent consequences. This step is consistent with discrete approaches where the lack of differentiability is also not explicitly taken into account.

The derivation of the adjoint equations closely follows classical techniques from calculus of variations, as outlined in Ref. 36. In shape optimization, calculation of the first variation of functionals, such as those in Eqs. (3a) and (3b), requires that the integral on the modified surface be expressed in terms of quantities on the original surface. For example, cost functions such as drag minimization are composed of terms that involve products of both geometric and nongeometric quantities:

$$I_c(\mathbf{Q}, \mathbf{D}) = \oint_\Gamma (g(\mathbf{Q}(\mathbf{D}))k(\mathbf{D})) ds \tag{4}$$

Here, g is an arbitrary function of the flow variables, and k represents either k_x or k_y. For cost functions such as Eq. (3b), k assumes a value of unity. A general form for the first variation can be written as

$$\delta I_c(\mathbf{Q}, \mathbf{D}) = \oint_{\Gamma'} (g_{new} k_{new}) ds' - \oint_\Gamma (g_{old} k_{old}) ds \tag{5}$$

where Γ and Γ' represent the old and the new surface of the geometry, respectively, and the subscripts old and new denote quantities on these surfaces. Evaluation of these integrals is addressed after a discussion on obtaining variations of I_R.

The method for obtaining the variations of the volume integral in Eq. (2) involving the residual $\mathbf{R}$ follows closely that of Pironneau.[31] Denoting this volume integral as

$$I_R(\mathbf{Q}, \mathbf{D}, \Psi) = \int_\Omega (\Psi, \mathbf{R}) d\Omega \tag{6}$$

the variation that properly accounts for volume changes, as well as for changes in the flow field, is given by

$$\delta I_R = -\int_\Omega \tilde{\mathbf{Q}}^T \left(A^T \frac{\partial \Psi}{\partial x} + B^T \frac{\partial \Psi}{\partial y} \right) d\Omega + \int_\Gamma \tilde{\mathbf{Q}}^T (A^T k_x + B^T k_y) \Psi ds \tag{7}$$

where $\mathbf{A}^T$ and $\mathbf{B}^T$ are the transposes of the inviscid flux Jacobian matrices and the surface integral is over the solid walls as well as the far-field. In deriving Eq. (7), it is tacitly assumed that the fluxes and the costate variables are differentiable; similar assumptions have been discussed earlier regarding the cost function. The variation of the augmented cost function in Eq. (2) is formed by combining Eq. (7) with the variations in $I_c(\mathbf{Q}, \mathbf{D})$. Because $\tilde{\mathbf{Q}}$ is arbitrary, the volume integrals present in the variation of the augmented cost function can be eliminated by requiring that Ψ satisfy the following adjoint (costate) equation:

$$-A^T \frac{\partial \Psi}{\partial x} - B^T \frac{\partial \Psi}{\partial y} = 0 \tag{8}$$

The surface integral in Eq. (7) is used together with the variations in the cost function $I_c(\mathbf{Q}, \mathbf{D})$ to determine both the boundary conditions and the sensitivity derivatives. The boundary conditions for Ψ are chosen to eliminate the terms that multiply $\tilde{\mathbf{Q}}$ on the boundaries. The surface integral can be rewritten as

$$\int_\Gamma \tilde{\mathbf{Q}}^T (A^T k_x + B^T k_y) \Psi d\Gamma = \int_\Gamma (\bar{A}^T \Psi, \tilde{\mathbf{Q}}) d\Gamma \tag{9}$$

where $\bar{A}^T = A^T k_x + B^T k_y$. In the far field, this term can be rewritten by using a locally one-dimensional characteristic decomposition at the boundary to yield

$$\int_\Gamma (\bar{A}^T \Psi, \tilde{\mathbf{Q}}) d\Gamma = \int_\Gamma (\Psi, T \bar{\Lambda} \tilde{W}) d\Gamma \tag{10}$$

where $\tilde{\mathbf{W}} = \mathbf{T}^{-1} \tilde{\mathbf{Q}}$, $\mathbf{T}^{-1}$ is the matrix of left eigenvectors of $\bar{\mathbf{A}}$, and $\bar{\Lambda}$ are the corresponding eigenvalues. Boundary conditions for the costate variables in the far field are obtained using characteristic-type boundary conditions on the field variables, where the propagation of information is based on the signs of the eigenvalues. For shape optimization, variations in $\mathbf{W}$ associated with free-stream quantities are zero, so that the corresponding costate variables on the boundary can be extrapolated from the interior of the domain. The other costate variables on the boundary are obtained by requiring the remaining terms in Eq. (10) to vanish. When Mach number or angle of attack are design vari-

ables, variations in $\mathbf{W}$ reflect the appropriate changes in free-stream conditions and are used in obtaining the derivatives with respect to these variables.

On solid walls, the boundary condition that there is no flow normal to the new surface is written as

$$\left(Q_2 + \frac{\partial Q_2}{\partial x}\tilde{x} + \frac{\partial Q_2}{\partial y}\tilde{y} + \tilde{Q}_2\right)(k_x + \tilde{k}_x) + \left(Q_3 + \frac{\partial Q_3}{\partial x}\tilde{x} + \frac{\partial Q_3}{\partial y}\tilde{y} + \tilde{Q}_3\right)(k_y + \tilde{k}_y) = 0 \tag{11}$$

Using Eq. (11), the surface integral in Eq. (9) can be expressed as

$$\begin{aligned}\int_\Gamma (\bar{A}^T\Psi, \tilde{\mathbf{Q}})ds = \int_\Gamma (&\tilde{Q}_1[(k_x\psi_2 + k_y\psi_3)\phi] + \\ &\tilde{Q}_2[(k_x\psi_2 + k_y\psi_3)(1-\gamma)u] + \\ &\tilde{Q}_3[(k_x\psi_2 + k_y\psi_3)(1-\gamma)v] + \\ &\tilde{Q}_4[(k_x\psi_2 + k_y\psi_3)(\gamma-1)] + R(\psi_1 + \psi_2 u + \psi_3 v + \psi_4 H))ds\end{aligned} \tag{12a}$$

where H is the total enthalpy and

$$R = -(Q_2\tilde{k}_x + Q_3\tilde{k}_y) - \left(k_x\frac{\partial Q_2}{\partial x} + k_y\frac{\partial Q_3}{\partial x}\right)\tilde{x} + \left(k_x\frac{\partial Q_2}{\partial y} + k_y\frac{\partial Q_3}{\partial y}\right)\tilde{y} \tag{12b}$$

In order to compute the variation in Eq. (5), the integrand for the first integral is expanded as follows:

$$g_{new}k_{new} = (g + g_x\tilde{x} + g_y\tilde{y} + \tilde{g})(k_{new} + \tilde{k}_{new}) \tag{13}$$

In Eq. (13), the derivatives g_x and g_y account for spatial changes and $\tilde{g}$ reflects the variation due to the fact that the solution of the governing equations has changed in response to the changing surface. Note that for structured grids, which employ a mapping to a fixed computational domain, these spatial derivatives do not arise because the variations in the generalized coordinates are zero. However, variations in the mapping function need to be considered which naturally provides a mechanism to account for grid sensitivities in a continuous framework.[32]

The boundary conditions for the costate variables are derived by combining the boundary terms from the variation in the cost function with those from Eq. (12a) and then eliminating terms that involve variations in Q. Because g is a function of Q, $\tilde{g}$ is given by

$$\tilde{g} = \sum_{i=1}^{4} \frac{\partial g}{\partial Q_i} \tilde{Q}_i \tag{14}$$

The boundary terms that multiply $\tilde{Q}$ are eliminated by requiring that

$$(k_x \psi_2 + k_y \psi_3) \begin{bmatrix} \phi \\ (1-\gamma)u \\ (1-\gamma)v \\ (\gamma - 1) \end{bmatrix} + k \left[\frac{\partial g}{\partial Q_1}, \frac{\partial g}{\partial Q_2}, \frac{\partial g}{\partial Q_3}, \frac{\partial g}{\partial Q_4} \right]^T = 0 \tag{15}$$

Note that the column vector that multiplies $(k_x \psi_2 + k_y \psi_3)$ corresponds to the derivatives of pressure with respect to the dependent variables. In order to obtain a unique boundary condition for $(k_x \psi_2 + k_y \psi_3)$, the second column must be a scalar multiple of the first. Therefore, g can only be a function of pressure, $h(p)$, which yields the following boundary condition:

$$k_x \psi_2 + k_y \psi_3 + k \frac{d}{dp} h(p) = 0 \tag{16}$$

Thus, cost functions such as specification of a velocity distribution or minimization of surface entropy are inadmissible, except in special cases where they can be expressed solely in terms of pressure.

As an example of an allowable cost function, consider the drag coefficient given by

$$c_d = \frac{2}{\gamma M_\infty^2} \int_\Gamma \left(\frac{p}{p_\infty} - 1 \right) (k_x \cos\alpha + k_y \sin\alpha) ds \tag{17}$$

The appropriate boundary condition for this case is given by

$$k_x \psi_2 + k_y \psi_3 + \frac{2}{\gamma M_\infty^2 p_\infty} (k_x \cos\alpha + k_y \sin\alpha) = 0 \tag{18}$$

Surface Parameterization

In shape design, the best representation of the surface for design problems remains an open issue. In the current study, the geometries are modeled with B splines, which offer great flexibility in the definition of the surfaces. By varying the polynomial degree and the number of control points, a wide range in the number of design variables and surface fidelity can be obtained. On one hand, the design variables can be made to correspond to the

individual grid points on the surface by choosing a linear polynomial and an appropriate number of control points. Conversely, a single polynomial curve of degree n (known as a Bezier curve) can be used to describe the geometry by choosing the number of control points to be $n + 1$. In addition, through the knot sequence associated with the spline, sharp breaks in the surface such as those that occur in cove regions and blunt trailing edges can still be represented in a single curve.

In a B-spline representation, the x- and y-coordinates of the surfaces are written in a parametric form as[14]

$$x(t) = \sum_{i=1}^{n+1} X_i N_{i,k}(t) \tag{19a}$$

$$y(t) = \sum_{i=1}^{n+1} Y_i N_{i,k}(t) \tag{19b}$$

where (x, y) are the Cartesian coordinates of the surface, $N_{i,k}$ is the B-spline basis function of order k, (X_i, Y_i) are the coordinates of the B-spline control polygon, and $n + 1$ is the total number of control points. Notice that the surface description with Eqs. (19a) and (19b) is still continuous.

In Fig. 1, a point on the old surface is assumed to move to the new surface while remaining at a fixed value of t. Consequently, variations in the basis functions need not be considered. In addition, generality is maintained for the surface geometry as variations are not restricted to being strictly normal to the existing surface.

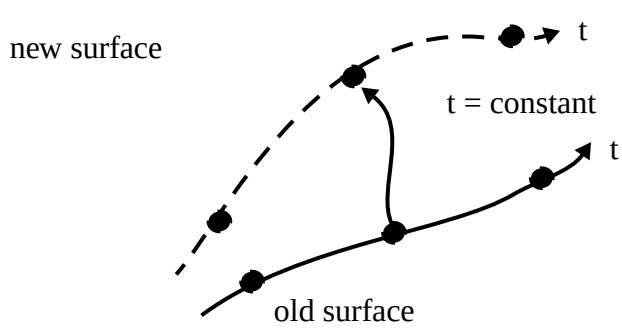


Figure 1. Movement of point on surface.

For small variations, an incremental length on the new surface can be written as

$$ds' = (1 + \Delta)ds \tag{20}$$

where

$$\Delta = \frac{C_x\dot{x} + C_y\dot{y}}{\dot{x}^2 + \dot{y}^2} \tag{21}$$

and

$$C_x = \sum_{i=1}^{n+1} \tilde{X}_i \cdot \frac{dN_{i,k}}{dt} \tag{22a}$$

$$C_y = \sum_{i=1}^{n+1} \tilde{Y}_i \cdot \frac{dN_{i,k}}{dt} \tag{22b}$$

Here, $\tilde{X}_i$ and $\tilde{Y}_i$ are variations in the position of the B-spline control points, and $\dot{x}$ and $\dot{y}$ are derivatives with respect to t. Because a given point on both the old and new surfaces is at a fixed value of t, the coordinates on the new surface can be written as

$$x_{new} = \sum_{i=1}^{n+1} X_i^{new} N_{i,k} = \sum_{i=1}^{n+1} (X_i^{old} + \tilde{X}_i) N_{i,k} \tag{23a}$$

$$y_{new} = \sum_{i=1}^{n+1} Y_i^{new} N_{i,k} = \sum_{i=1}^{n+1} (Y_i^{old} + \tilde{Y}_i) N_{i,k} \tag{23b}$$

Therefore, the variation of a point on the surface is given by

$$\tilde{x} = \sum_{i=1}^{n+1} \tilde{X}_i N_{i,k} \tag{24a}$$

$$\tilde{y} = \sum_{i=1}^{n+1} \tilde{Y}_i N_{i,k} \tag{24b}$$

Since the components of the surface normal can be expressed as

$$k_x = \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \tag{25a}$$

$$k_y = \frac{-\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \tag{25b}$$

the variations in the surface normals can be derived as

$$\tilde{k}_x = \frac{(1 - k_x^2)}{\sqrt{\dot{x}^2 + \dot{y}^2}} \sum_{i=1}^{n+1} \tilde{Y}_i \frac{dN_{i,k}}{dt} + \frac{k_x k_y}{\sqrt{\dot{x}^2 + \dot{y}^2}} \sum_{i=1}^{n+1} \tilde{X}_i \frac{dN_{i,k}}{dt} \tag{26a}$$

$$\tilde{k}_y = -\frac{(1 - k_y^2)}{\sqrt{\dot{x}^2 + \dot{y}^2}} \sum_{i=1}^{n+1} \tilde{X}_i \frac{dN_{i,k}}{dt} - \frac{k_x k_y}{\sqrt{\dot{x}^2 + \dot{y}^2}} \sum_{i=1}^{n+1} \tilde{Y}_i \frac{dN_{i,k}}{dt} \tag{26b}$$

By using Eqs. (13), and (20)–(26b), variations of integrals that involve gk_y can be written as

$$\begin{aligned} \delta\left(\int_\Gamma g k_y ds\right) = & \int_\Gamma \tilde{g} k_y ds - \sum_{i=1}^{n+1} \tilde{X}_i \\ & \int_\Gamma \left(g \frac{dN_{i,k}}{dt} + \dot{x} \frac{\partial g}{\partial x} N_{i,k} \right) dt \\ & - \sum_{i=1}^{n+1} \tilde{Y}_i \int_\Gamma \left(\dot{x} \frac{\partial g}{\partial y} N_{i,k} \right) dt \end{aligned} \tag{27a}$$

Similarly, variations of integrals involving gk_x are given by

$$\begin{aligned} \delta\left(\int_\Gamma g k_x ds\right) = & \int_\Gamma \tilde{g} k_x ds + \sum_{i=1}^{n+1} \tilde{X}_i \int_\Gamma \left(\dot{y} \frac{\partial g}{\partial x} N_{i,k} \right) dt \\ & + \sum_{i=1}^{n+1} \tilde{Y}_i \int_\Gamma \left(\dot{y} \frac{\partial g}{\partial y} N_{i,k} + g \frac{\partial N_{i,k}}{\partial t} \right) dt \end{aligned} \tag{27b}$$

For cost functions such as (3b) that only involve flow quantities, a similar procedure yields

$$\delta\left(\int_\Gamma g\,ds\right) = \int_\Gamma \tilde{g}\,ds \tag{27c}$$

$$+ \sum \tilde{Y}_i \int_\Gamma \left(N_{i,k}\frac{\partial g}{\partial y} + \frac{g\dot{y}}{\dot{x}^2 + \dot{y}^2}\frac{dN_{i,k}}{dt} \right)\sqrt{\dot{x}^2 + \dot{y}^2}\,dt$$

$$+ \sum \tilde{X}_i \int_\Gamma \left(N_i\frac{\partial g}{\partial x} + \frac{g\dot{x}}{\dot{x}^2 + \dot{y}^2}\frac{dN_i}{dt} \right)\sqrt{\dot{x}^2 + \dot{y}^2}\,dt$$

The terms in Eqs. (27a), (27b), and (27c) involving $\tilde{g}$ are eliminated by the boundary conditions along with the terms involving $\tilde{Q}$ in Eq. (12a) as discussed earlier. The sensitivity derivatives are obtained by combining the remaining terms in Eq. (27a), (27b), or (27c) with the last term in Eq. (12a). For example, using Eqs. (24a)–(26b) to compute the variations in the coordinates and metric terms, the sensitivity derivatives of the drag coefficient with respect to each B-spline control point are given by

$$\frac{\partial c_d}{\partial \tilde{X}_i} = \frac{2}{\gamma M_\infty^{\,2}}\int_\Gamma \left(\left(\dot{y}\frac{\partial g_1}{\partial x} - \dot{x}\frac{\partial g_2}{\partial x} \right) N_{i,k} - g_2\frac{dN_{i,k}}{dt} \right) dt \tag{28a}$$

$$+ \int_\Gamma \left(Q_3\frac{dN_{i,k}}{dt} - \left(k_x\frac{\partial Q_2}{\partial x} + k_y\frac{\partial Q_3}{\partial x} \right) N_{i,k} \right)(\Psi_1 + u\Psi_2 + v\Psi_3 + H\Psi_4)\,dt$$

$$\frac{\partial c_d}{\partial \tilde{Y}_i} = \frac{2}{\gamma M_\infty^{\,2}}\int_\Gamma \left(\left(\dot{y}\frac{\partial g_1}{\partial y} - \dot{x}\frac{\partial g_2}{\partial y} \right) N_{i,k} - g_1\frac{dN_{i,k}}{dt} \right) dt \tag{28b}$$

$$+ \int_\Gamma \left(-Q_2\frac{dN_{i,k}}{dt} - \left(k_x\frac{\partial Q_2}{\partial y} + k_y\frac{\partial Q_3}{\partial y} \right) N_{i,k} \right)(\Psi_1 + u\Psi_2 + v\Psi_3 + H\Psi_4)\,dt$$

where $g_1 = (p/p_\infty - 1)\cos\alpha$ and $g_2 = (p/p_\infty - 1)\sin\alpha$. For cost functions such as lift or moment coefficients, a similar procedure is followed. When Mach number and angle of attack are considered as design variables, variations from surface integrals in the far field also contribute to the sensitivity derivatives.

Navier-Stokes

In this section, the adjoint equations with the associated boundary conditions and the expressions for the sensitivity derivatives are derived for viscous flows. Only steady incompressible viscous flows are considered in this paper to make the analysis more transparent.

The governing equations, with the artificial compressibility parameter β, are given by

$$\frac{\partial}{\partial x}\beta u + \frac{\partial}{\partial y}\beta v = 0$$

$$\frac{\partial}{\partial x}(u^2 + p) + \frac{\partial}{\partial y}(uv) = \frac{\mu}{Re}\left(\frac{\partial}{\partial x}\left(2\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) \tag{29}$$

$$\frac{\partial}{\partial x}(uv) + \frac{\partial}{\partial y}(v^2 + p) = \frac{\mu}{Re}\left(\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(2\frac{\partial v}{\partial y}\right)\right)$$

The cost function is augmented as in Eq. (2) with the flow equations as constraints through the Lagrange multipliers Ψ with $\mathbf{Q} = \{p, u, v\}^T$. The variation in I_R is split into the contributions from the inviscid and the viscous terms I_R^{inv} and I_R^{visc}, respectively. These can be derived as

$$\delta I_R^{inv} = -\int_\Omega \tilde{\mathbf{Q}}^T\left(A^T\frac{\partial \Psi}{\partial x} + B^T\frac{\partial \Psi}{\partial y}\right)d\Omega + \int_\Gamma \tilde{\mathbf{Q}}^T(\mathbf{A}^T k_x + \mathbf{B}^T k_y)\Psi ds \tag{30}$$

and

$$\begin{aligned}
\frac{Re}{\mu}\delta I_R^{visc} = &-\int_\Omega \tilde{u}\left(\frac{\partial}{\partial x}\left(2\frac{\partial \Psi_2}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right)\right)d\Omega \\
&-\int_\Omega \tilde{v}\left(\frac{\partial}{\partial y}\left(2\frac{\partial \Psi_3}{\partial y}\right) + \frac{\partial}{\partial x}\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right)\right)d\Omega \\
&-\int_\Gamma \Psi_2\left[k_x 2\frac{\partial}{\partial x}\tilde{u} + k_y\left(\frac{\partial}{\partial y}\tilde{u} + \frac{\partial}{\partial x}\tilde{v}\right)\right]ds \\
&-\int_\Gamma \Psi_3\left[k_y 2\frac{\partial}{\partial x}\tilde{v} + k_x\left(\frac{\partial}{\partial y}\tilde{u} + \frac{\partial}{\partial x}\tilde{v}\right)\right]ds \\
&+\int_\Gamma k_x\left[2\tilde{u}\frac{\partial \Psi_2}{\partial x} + \tilde{v}\frac{\partial \Psi_2}{\partial y} + \tilde{v}\frac{\partial \Psi_3}{\partial x}\right]ds \\
&+\int_\Gamma k_y\left[2\tilde{v}\frac{\partial \Psi_3}{\partial y} + \tilde{u}\frac{\partial \Psi_2}{\partial y} + \tilde{u}\frac{\partial \Psi_3}{\partial x}\right]ds
\end{aligned} \tag{31}$$

Combining the field integrals in Eqs. (30) and (31) and setting the integrands to zero yields the following adjoint system:

$$-A^T\frac{\partial \Psi}{\partial x} - B^T\frac{\partial \Psi}{\partial y} = \mathbf{T} \tag{32a}$$

where

$$\mathbf{T} = \frac{\mu}{Re}\begin{bmatrix} 0 \\ \frac{\partial}{\partial x}\left(2\frac{\partial \Psi_2}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right) \\ \frac{\partial}{\partial x}\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right) + \frac{\partial}{\partial y}\left(2\frac{\partial \Psi_3}{\partial y}\right) \end{bmatrix} \tag{32b}$$

For purposes of illustration, the boundary conditions for the adjoint system are derived with the assumption that the cost function is the drag coefficient:

$$\begin{aligned} c_d = \int_\Gamma \Big[& c_p k_x \cos\alpha + c_p k_y \sin\alpha \\ & -\frac{2\mu}{Re}\left(2k_x\frac{\partial u}{\partial x} + k_y\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right)\cos\alpha \\ & -\frac{2\mu}{Re}\left(2k_y\frac{\partial v}{\partial y} + k_x\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right)\sin\alpha \Big] ds \end{aligned} \tag{33}$$

Note that the expression for drag retains the components from the viscous stress tensor. It will be seen that this is a requirement for obtaining boundary conditions for the adjoint equations. Using Eqs. (27a) and (27b), the variation in the drag coefficient is given by

$$\begin{aligned} \delta c_d = \; & \delta c_d^{flow} + \sum \tilde{X}_i \int_\Gamma \left(\dot{Y} N_{i,k} \frac{\partial G_1}{\partial x} - \dot{X} N_{i,k} \frac{\partial G_2}{\partial x} - \frac{\partial N_{i,k}}{\partial t} G_2 \right) dt \\ & + \sum \tilde{Y}_i \int_\Gamma \left(\dot{Y} N_{i,k} \frac{\partial G_1}{\partial y} - \dot{X} N_{i,k} \frac{\partial G_2}{\partial y} - \frac{\partial N_{i,k}}{\partial t} G_1 \right) dt \\ & + \int_\Gamma \tilde{u}\left(\beta k_x \Psi_1 + \frac{\mu}{Re}\left(2k_x\frac{\partial \Psi_2}{\partial x} + k_y\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right)\right)\right)\sqrt{\dot{x}^2 + \dot{y}^2}\, dt \\ & + \int_\Gamma \tilde{v}\left(\beta k_y \Psi_1 + \frac{\mu}{Re}\left(2k_y\frac{\partial \Psi_3}{\partial y} + k_x\left(\frac{\partial \Psi_2}{\partial y} + \frac{\partial \Psi_3}{\partial x}\right)\right)\right)\sqrt{\dot{x}^2 + \dot{y}^2}\, dt \end{aligned} \tag{34a}$$

where

$$\delta c_d^{flow} = \int_\Gamma \Bigg(2\tilde{p}(k_x \cos\alpha + k_y \sin\alpha) - \frac{2\mu}{Re}\left(2k_x \frac{\partial}{\partial x}\tilde{u} + k_y\left(\frac{\partial}{\partial y}\tilde{u} + \frac{\partial}{\partial x}\tilde{v}\right)\right)\cos\alpha - \frac{2\mu}{Re}\left(2k_y \frac{\partial}{\partial y}\tilde{v} + k_x\left(\frac{\partial}{\partial y}\tilde{u} + \frac{\partial}{\partial x}\tilde{v}\right)\right)\sin\alpha \Bigg)\sqrt{\dot{x}^2 + \dot{y}^2}\,dt \tag{34b}$$

and

$$G_1 = 2p\cos\alpha - 4\frac{\mu}{Re}\frac{\partial u}{\partial x}\cos\alpha - 2\frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\sin\alpha \tag{34c}$$

$$G_2 = 2p\sin\alpha - 4\frac{\mu}{Re}\frac{\partial v}{\partial y}\sin\alpha - 2\frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\cos\alpha \tag{34d}$$

Expressing the velocities on the new surface in a Taylor series and noting that the velocities on the old and new surface are both zero, the variations in the velocity components can be written as

$$\tilde{u} = -\frac{\partial u}{\partial x}\tilde{x} - \frac{\partial u}{\partial y}\tilde{y} \tag{35a}$$

$$\tilde{v} = -\frac{\partial v}{\partial x}\tilde{x} - \frac{\partial v}{\partial y}\tilde{y} \tag{35b}$$

In order to derive the boundary conditions, Eqs. (30), (31), (32a), and (32b) are combined, and terms that involve the variations in the velocity gradients and $\tilde{p}$ are eliminated. This requires that the following relationships hold:

$$k_x\Psi_2 + k_y\Psi_3 + 2k_x\cos\alpha + 2k_y\sin\alpha = 0 \tag{36a}$$

$$-4k_x\cos\alpha - 2\Psi_2 k_x + 4k_y\sin\alpha + 2k_y\sin\alpha = 0 \tag{36b}$$

$$-2k_y\cos\alpha - 2k_x\sin\alpha - \Psi_2 k_y - \Psi_3 k_x = 0 \tag{36c}$$

This system is overdetermined and is satisfied by the choice

$$\Psi_2 = -2\cos\alpha \tag{37a}$$

$$\Psi_3 = -2\sin\alpha \tag{37b}$$

The variation in the drag coefficient can be obtained from Eq. (34a) by using these equations in conjunction with Eqs. (24a) and (24b).

Without the inclusion of the full stress tensor in the cost function, it is not possible to obtain a consistent set of boundary conditions for Ψ_2 and Ψ_3. Generally, suitable cost functions are composed of terms that will appropriately balance the boundary terms from the residuals. In particular, cost functions such as lift, drag, and pitching moment are admissible. It is not immediately obvious that the specification of a pressure distribution is allowable because of the absence of viscous terms in the cost function. However, a suitable cost function can be obtained by first replacing the pressure term in the stress tensor by the difference between the current and the desired pressure coefficient Δc_p. This is then premultiplied by the surface normal scaled by this difference in c_p and postmultiplied by the surface normal. In nondimensionalized variables, the resulting expression is given by

$$I_c = \int_\Gamma \{k_x \Delta c_p, k_y \Delta c_p\} \begin{bmatrix} \frac{\Delta c_p}{2} - \frac{2\mu}{Re}\frac{\partial u}{\partial x} & -\frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ -\frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{\Delta c_p}{2} - \frac{2\mu}{Re}\frac{\partial v}{\partial y} \end{bmatrix} \begin{Bmatrix} k_x \\ k_y \end{Bmatrix} ds \tag{38}$$

After expansion, Eq. (38) can be rewritten as

$$\begin{aligned} I_c = \int_\Gamma \Bigg(& \frac{1}{2}\Delta c_p{}^2 \\ & - \frac{\mu}{Re}\Delta c_p k_x\left(2k_x\frac{\partial u}{\partial x} + k_y\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) + k_y\left(2k_y\frac{\partial v}{\partial y} + k_y\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right)\Bigg) ds \end{aligned} \tag{39}$$

This equation can be recast in terms of the velocity gradient normal to the boundary as

$$I_c = \int_\Gamma \left(\frac{1}{2}\Delta c_p^2 - \frac{\mu \Delta c_p}{Re}\frac{\partial u_n}{\partial n}\right) ds \tag{40}$$

where u_n is the normal velocity component and n is the surface normal direction. The velocity gradient term in this equation is zero by the continuity equation, so that the cost function in Eq. (39) corresponds to specifying a pressure distribution. However, all the

terms in Eq. (39) are required for the derivation of the boundary conditions for the adjoint equations. The final boundary conditions on Ψ_2 and Ψ_3 for specifying a pressure distribution are given by

$$\Psi_2 = -2k_x(c_p - c_p^*) \tag{41a}$$

$$\Psi_3 = -2k_y(c_p - c_p^*) \tag{41b}$$

The continuous adjoint formulation for Navier-Stokes equations described in this section poses a problem in the evaluation of the sensitivity derivatives. The evaluation of these derivatives requires second derivatives of the velocity components because G_1 and G_2 involve velocity gradients that are further differentiated in Eq. (34a). Recall that these terms arise from expressing the cost function on the new surface in a Taylor series expansion about the old surface. In the present work, because the flow solver is only second-order accurate, pointwise second derivatives are inconsistent in general. An accurate evaluation of second derivatives would require the flow solver to be at least third-order accurate. If a mapping is employed, as is possible with structured grids, the surface remains at a constant coordinate line, and this problem does not occur.

In a discrete adjoint approach, the restriction on defining a suitable cost function and the need for second derivatives are eliminated, as will be shown. However, the full implication of designing for cost functions in a discrete framework for which boundary conditions are not obtainable in the continuous case is not clear at this time.

Discretization

Flow equations

The discretization of the flow equations is first addressed since it has implications for the discretization of the adjoint equations. The discretization of the compressible inviscid equations is given first; a similar procedure is used to discretize the inviscid contributions for the incompressible Navier-Stokes equations. The equations represent a system of conservation laws for a control volume that relates the rate of change of a vector of state variables $\mathbf{Q}$ to the flux through the volume surface. The equations are written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}\,d\Omega + \oint_{\Gamma}\mathbf{F}(\mathbf{Q},\mathbf{n})\,ds = 0 \tag{42}$$

where for compressible flows $\mathbf{Q} = [\rho, \rho u, \rho v, E]^T$ and $\mathbf{F}(\mathbf{Q},\mathbf{n})$ is the flux of mass, momentum, and energy through the control volume. In these equations, $\mathbf{n}$ is the vector normal to the boundary, ρ is the density, u and v are the Cartesian velocity components, and E is the total energy per unit volume. These equations are closed by the equation of state for a perfect gas.

In discretizing Eq. (42), the variables are stored at the vertices of a triangular mesh. The control volumes are defined by the median dual. The discrete form of Eq. (42) for vertex i, with an associated control volume Ω_i, is given by

$$\frac{\partial}{\partial t}\int_{\Omega_i} \mathbf{Q}\, d\Omega + \sum_{j \in N_i} \mathbf{F}_{ij} l_{ij} = 0 \tag{43}$$

where $\mathbf{F}_{ij}$ is the numerical flux that approximates the normal flux through the control-volume edge dual to the triangle edge that joins nodes i and j, l_{ij} is the length of the dual edge, and N_i is the set of vertex neighbors of i. The numerical fluxes are computed by using a Roe-type approximate Riemann solver:[34]

$$\mathbf{F}_{ij} = \frac{1}{2}[\mathbf{F}(\mathbf{Q_i};\mathbf{n}) + \mathbf{F}(\mathbf{Q}_j;\mathbf{n}) - \left|\tilde{\mathbf{A}}(\mathbf{Q_r}, \mathbf{Q_l};\mathbf{n})\right|(\mathbf{Q_r} - \mathbf{Q}_l)] \tag{44}$$

where $\tilde{\mathbf{A}}$ is the Jacobian matrix evaluated at the Roe state, and $\mathbf{Q}_r$ and $\mathbf{Q}_l$ are the dependent variables on the right and left boundaries of the control volume face which are obtained by extrapolation:

$$\mathbf{Q}_l = \mathbf{Q}_i + \frac{\varphi}{2}\nabla\mathbf{Q} \cdot (\mathbf{r}_j - \mathbf{r}_i) \tag{45a}$$

$$\mathbf{Q}_r = \mathbf{Q}_j + \frac{\varphi}{2}\nabla\mathbf{Q} \cdot (\mathbf{r}_i - \mathbf{r}_j) \tag{45b}$$

where $\varphi = 0$ for first-order discretization, $\varphi = 1$ for second-order discretization, and $\mathbf{r}_i$ and $\mathbf{r}_j$ are the position vectors of nodes i and j, respectively. Note that the definition of the fluxes in Eq. (44) is different from a standard Riemann solver in that the unsplit fluxes are evaluated by using data at the nodes $\mathbf{Q}_i$ and $\mathbf{Q}_j$ instead of data at the extrapolated states $\mathbf{Q}_r$ and $\mathbf{Q}_l$. This discretization remains second-order accurate and has the benefit that the only term that involves data other than at the immediate neighbors occurs through the dissipation. This enables a discretization of the continuous adjoint equations to be easily obtained that is identical to the discrete adjoint approach, except for small differences that arise from the higher order dissipation.

For computing the viscous contributions to the residual, a finite-volume scheme is used that is equivalent to a Galerkin discretization with linear basis functions. On triangular grids, this discretization only requires data at the immediate neighboring nodes.

Adjoint equations

The adjoint equations can, in principle, be discretized by any stable and consistent method. However, insufficient grid resolution may result in poor accuracy of the sensitivity derivatives in that they do not agree with those obtained by finite differences. Inaccurate sensitivity derivatives may lead to failure in the optimization process.[37] Sensitivity de-

rivatives that agree with finite-difference gradients can be obtained regardless of grid size by making the equations discretely adjoint to the discretized flow equations. However, achieving this for higher order discretizations can be an onerous task. In the present work, the discretization is been derived with strong guidance from a discrete adjoint formulation so that for first-order accuracy, a direct correspondence with a discrete adjoint approach is achieved. Higher order accuracy for the discretization of the inviscid terms is obtained through the use of extrapolation of the costate variables.

The discretization of the adjoint equation is performed by adding a time derivative to Eq. (8) and using a finite-volume type of method similar to that used for the flow solver. In this context, Eq. (8) is integrated over control volumes, where the matrices are taken outside the integrand and are evaluated using nodal point values of the dependent variables:

$$\frac{\partial}{\partial t}\int_{\Omega}\Psi d\Omega - \left(\mathbf{A}_i^T\int_{\Omega}\frac{\partial\Psi}{\partial x}d\Omega + \mathbf{B}_i^T\int_{\Omega}\frac{\partial\Psi}{\partial y}d\Omega\right) = 0 \tag{46}$$

The volume integrals are converted to surface integrals over each of the control volumes, and the values of the costate variables on the boundaries are obtained by using upwind type formulas:

$$\Psi_{ij} = \frac{1}{2}\left[(\Psi_i + \Psi_j) + \bar{\mathbf{A}}(\mathbf{Q}_i;\mathbf{n})^{-T}\left(\frac{\partial\Phi}{\partial\mathbf{Q}}\right)^T(\Psi_r - \Psi_l)\right] \tag{47}$$

where the extrapolated costate variables Ψ_r and Ψ_l are obtained by using formulas that are similar to Eqs. (45a) and (45b). The data used for evaluation of the matrices and the formulas used for obtaining the costate variables on the faces of the control volumes have been chosen so that a discrete adjoint formulation is obtained for first-order spatial accuracy. The resulting discretization of the inviscid contributions may be written as follows:

$$\frac{\partial}{\partial t}\int_{\Omega_i}\Psi d\Omega - \sum_{j\in N_i}\mathbf{G}_{ij}l_{ij} = 0 \tag{48}$$

The numerical flux, $\mathbf{G}_{ij}$, used in calculating the residual for the control volume that surrounds node i, is given by

$$\mathbf{G}_{ij} = \frac{1}{2}\left[\bar{\mathbf{A}}(\mathbf{Q}_i;\mathbf{n})^T(\Psi_i + \Psi_j) + \left(\frac{\partial\Phi}{\partial\mathbf{Q}_l}\right)^T(\Psi_r - \Psi_l)\right] \tag{49}$$

where $\Phi = \left|\tilde{\mathbf{A}}(\mathbf{Q_r}, \mathbf{Q_l};\mathbf{n})\right|(\mathbf{Q}_r - \mathbf{Q}_l)$. Note that $\mathbf{G}_{ij} \neq -\mathbf{G}_{ji}$.

On solid boundaries, the flux along the wall for closing off the surface integral around node i is given by

$$\mathbf{G}_i l_i = k_x\mathbf{A}_i^T\Psi + k_y\mathbf{B}_i^T\Psi = \bar{\mathbf{A}}_i^T\Psi \tag{50}$$

For enforcing the boundary conditions on the costate variables, a weak formulation is used in which the fluxes are modified appropriately to reflect the imposition of the boundary conditions. Numerically, the Jacobian matrix in Eq. (50) is evaluated without explicitly enforcing the boundary condition on the flow variables that no flow is allowed through the surface. In this way, the contributions from the fluxes in the interior in conjunction with the boundary flux in Eq. (50) combine so that the resulting discretization corresponds with that from a discrete adjoint approach.

Note that in Eq. (49) the linearization of Φ is somewhat cumbersome but has been previously derived (see for example Ref. 4). A simpler equation can be obtained by employing the approximate linearization of Φ as

$$\frac{\partial \Phi}{\partial \mathbf{Q}} = \left|\tilde{\mathbf{A}}\right| \tag{51}$$

This equation is less complicated than the full linearization and only differs from the exact linearization in proportion to $(\mathbf{Q}_r - \mathbf{Q}_l)$, but numerical experiments have indicated that on very coarse grids some of the sensitivity derivatives are of poor accuracy compared with finite-difference derivatives. Although these errors decrease as the grid resolution increases, the full linearization is used in the current work.

Since the viscous equations used in the current study are for incompressible flow, the corresponding terms in the adjoint equations (Eq. (32a)) have the same form and are therefore discretized in the same manner. The Dirichlet boundary conditions for ψ_2 and ψ_3 are strongly enforced with the same technique used to set the velocities to zero in the flow solver. In the implicit solver, this is achieved by zeroing the off-diagonal elements in the rows of the matrix that correspond to boundary nodes, as well as the appropriate terms on the right-hand side.

For viscous flows, a direct correspondence with a discrete adjoint formulation is not achieved near solid boundaries. This is easily seen by examining the resulting matrix structures from both approaches for a small mesh shown in Fig. 2, where it is assumed that nodes 1, 3, and 5 lie on a solid wall.

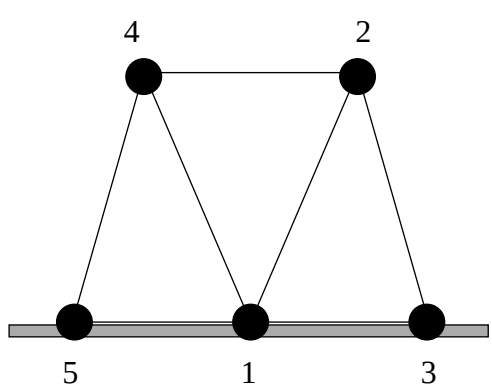


Figure 2. Sample mesh.

In the discrete adjoint approach, the augmented cost function is given by

$$I(\mathbf{Q}, \mathbf{D}, \Psi, \mathbf{X}(\mathbf{D})) = I_c(\mathbf{Q}, \mathbf{D}) + \Psi^T R(Q, D, X((D))) \tag{52}$$

where $\mathbf{R}$ is the vector of discrete residuals and, thus, depends explicitly on the grid-point locations X. Taking variations of Eq. (52) and regrouping terms yields the adjoint equation

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \{\Psi\} + \left\{\frac{\partial I_c}{\partial \mathbf{Q}}\right\} = 0 \tag{53}$$

The variation in the cost function is then given by

$$\delta I = \left(\frac{\partial I_c^T}{\partial D} + \Psi^T\left(\frac{\partial R}{\partial D} + \frac{\partial R}{\partial X}\frac{\partial X}{\partial D}\right)\right)\tilde{D} \tag{54}$$

In these equations, it is understood that the linearization of the residual includes the full effects of the boundary conditions. Here, $\partial X / \partial D$ represents the sensitivity of the interior grid points to changes in the design variables. In the continuous adjoint formulation described earlier, no counterpart to this term exists. The determination of grid sensitivities is dependent on the methodology used to restructure the mesh. Neglecting these terms is equivalent to freezing the interior grid points, regardless of changes in the surface geometry. Nevertheless, in a second-order-accurate scheme, the $\partial R / \partial D$ term in Eq. (54) accounts for changes in the residuals at the nodes immediately adjacent to the surface, as well as at the second nearest neighbors.

A diagram of the matrix structure associated with the configuration of nodes in Fig. 2 is shown in Fig. 3 for the discrete adjoint approach. The matrix structure for the continuous adjoint approach is shown in Fig. 4. In these figures, the solid circles represent the nonzero entries in the matrices. Note that in both figures, a first-order discretization of the inviscid terms is assumed so that the stencil only involves the nearest neighbors.

Comparing Figs. 3 and 4, it is seen that the matrix structures are significantly different. This is due to the strong enforcement of the no-slip condition in the flow solver, which leads to zeros in the columns of the adjoint system. For the continuous case, explicit enforcement of the boundary condition on Ψ_2 and Ψ_3 leads to zeros along rows. Of particular interest in the discrete adjoint case is that because of the zeros in the columns, the solution of the costate variables in the interior of the mesh does not depend on the values of Ψ_2 and Ψ_3 at the boundary. Furthermore, because the residual equation for the flow solver at these points is replaced by a Dirichlet condition on the velocities, the residual does not depend on the design variables so that $\partial R / \partial D = 0$. Therefore, there is no contribution to the sensitivity derivatives in Eq. (54) from these terms. The result is that in the discrete adjoint case the values of Ψ_2 and Ψ_3 on the boundary are completely arbitrary

and have no effect on the sensitivity derivatives. This has been verified by numerical experiments.

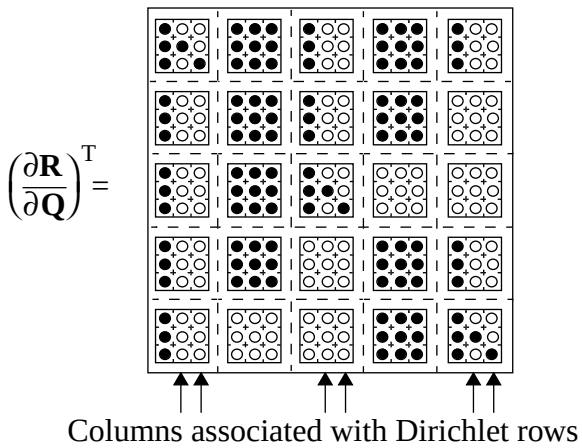


Figure 3. Matrix structure for discrete adjoint approach.

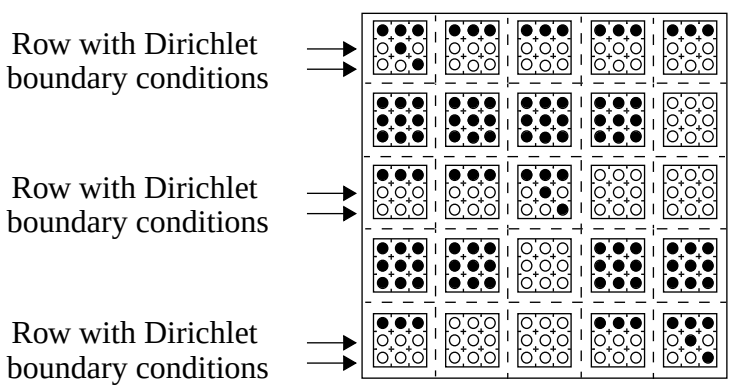


Figure 4. Matrix structure for continuous adjoint approach.

In light of the discussion above, it is of interest to compare the values of the costate variables that are obtained from both the continuous and the discrete adjoint formulations for a viscous flow. In Fig. 5, profiles of Ψ_2 as a function of the distance from the body are shown for a case in which the cost function is the drag of an airfoil and the location of the profile is taken to be at the midchord of the airfoil on the upper surface. In the figure, the values of Ψ_2 agree well away from the body. Near the boundary, however, the costate variables from the continuous and discrete formulations do not agree. As the mesh is refined, the distance from the surface of the airfoil in which these discrepancies occur de-

creases. Thus, one would expect that in the limit of vanishing mesh size, the two approaches would agree.

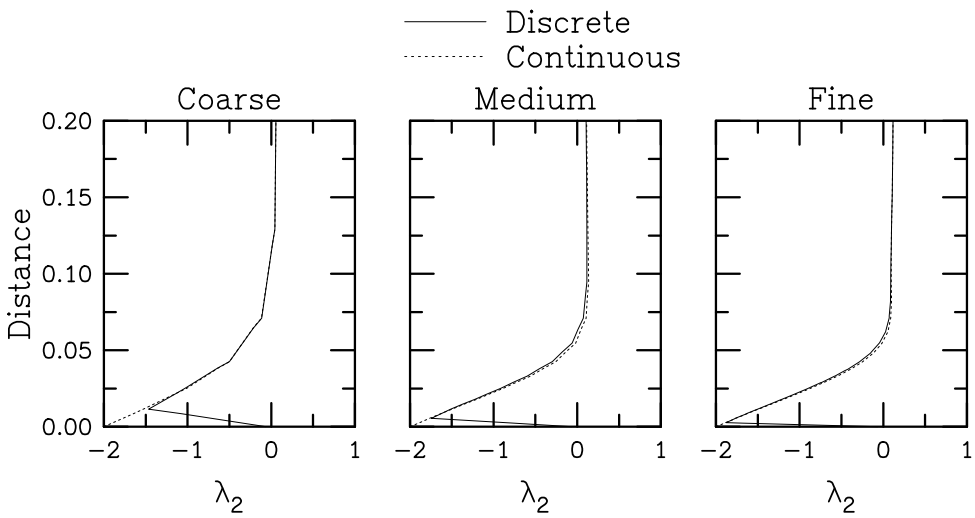


Figure 5. Profiles of Ψ_2 obtained from both discrete and continuous adjoint formulations.

In Eq. (53), Ψ can be determined provided that $\partial\mathbf{R}/\partial\mathbf{Q}$ is nonsingular irrespective of the cost function. Also, no difficulty is encountered in determining the sensitivity derivatives with Eq. (54). In particular, note that this equation does not require explicit calculation of second derivatives. Therefore, for viscous flows, a discrete approach is used in the current study, except that higher order accuracy for the inviscid terms is achieved by using the continuous approach described in the inviscid section. The implementation of this approach does not entail much additional effort because the inviscid terms are already discretely adjoint for first-order accuracy, and the viscous terms only involve the nearest neighbors. The accuracy of the derivatives using this approach is comparable to that obtained for inviscid flows. For first-order accuracy, the resulting method is identical to the standard discrete adjoint approach.

Solution procedures

For the flow equations, an implicit solution method with multigrid acceleration is used. Details may be found in Refs. 1, 2, and 9. The discretized equations for the costate variables in the absence of the time derivative represent a linear system that can be solved by using a technique such as preconditioned GMRES.[35] Alternatively, by retaining the time derivative, the equations can be solved to steady state by using a time-marching procedure. In the present work, the time term is included and a multigrid procedure is used with preconditioned GMRES as a smoother. The preconditioning is accomplished using an incomplete lower/upper (LU) decomposition with no fill-in. The motivation for retaining the time term is that this approach often converges in situations for which the GMRES

procedure might otherwise “stall.” Note that because the equations are linear, the matrix-vector products are easily formed by simply passing the vector to the residual routine in place of the costate variables. By forming the matrix-vector products in this way, the largest contribution to memory requirements is through the preconditioner so that the resulting scheme requires roughly the same amount of memory as the flow solver.

Grid generation and mesh movement

The unstructured meshes used in this work are generated using the software package described in Ref. 28. This employs an advancing front type of method that generates good quality grids for both inviscid and viscous calculations.

For shape optimization, the design is carried out in a domain that changes during the design cycle as the shape of the boundary changes. Therefore, the existing grid is modified in order to conform to the changing domain.

For inviscid flows, the strategy outlined in Ref. 42 is used to restructure the mesh in response to the changes in the surface shape. The tension-spring analogy is employed to allow the field grid points to respond to the displacements of the points on the surface. The following linear system of equations is solved with a Jacobi iteration strategy:

$$\sum_{j \in N_i} K_{ij}(\Delta \mathbf{x}_i - \Delta \mathbf{x}_j) = 0 \tag{55}$$

where $\Delta \mathbf{x}_i$ and $\Delta \mathbf{x}_j$ are the displacements from the initial position for nodes i and j. The spring stiffness K_{ij} is assumed to be l_{ij}^{-2}, where l_{ij} is the length of the edge that joins nodes i and j. Note that by using Eq. (55), the mesh remains unchanged when the surface is held fixed. When the boundary shape changes during the design cycle, this method does not guarantee that the grid lines will not cross. An improvement is to make the spring system nonlinear (i.e., the shape change is decomposed into smaller steps, and the procedure is repeated at each step). Also, in order to maintain good mesh quality throughout the design cycle, the edges are reconnected according to either a Delaunay criterion or by locally minimizing maximum angles (min-max).

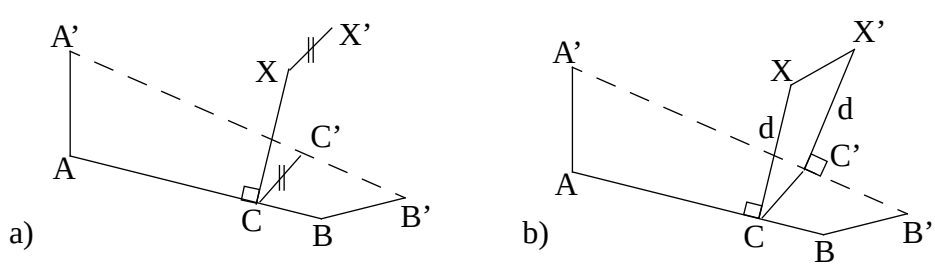
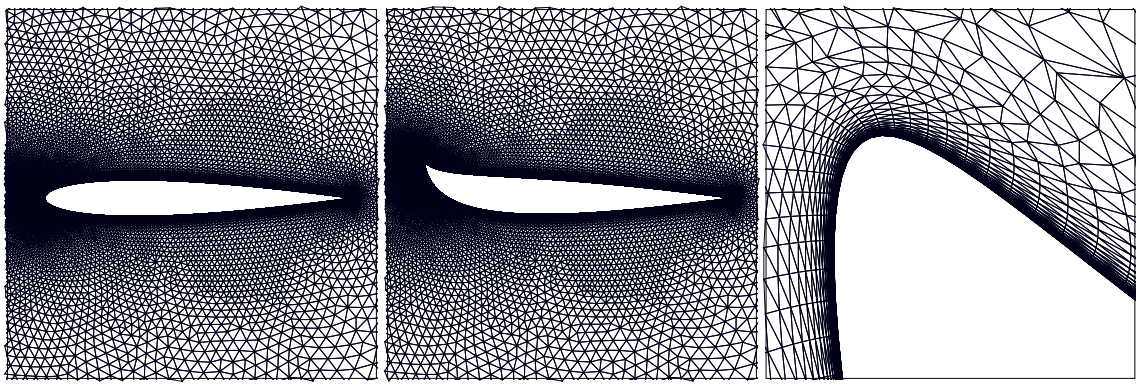


Figure 6. Methodology for mesh movement for viscous grids.

For grids with high-aspect-ratio cells, the inviscid strategy fails for a number of reasons. The spring analogy typically results in invalid grids with crossing of grid lines. In addition, both the Delaunay and min-max criteria often result in nodes with large connectivities. Therefore, the grid-movement scheme is modified to deal with Navier-Stokes grids. The Delaunay criterion is replaced by the min-max criterion where the swapping is only carried out if the maximum angle exceeds a specified angle (set to 150°). The distance to the wall for each node in the mesh is first computed. When the points on the surface are displaced, the field points move in response, as shown in Fig. 6(a). Here, AB is an edge on the surface of the body. Nodes A and B move to A' and B', respectively. For the field point X, the nearest point on edge AB is denoted by C. Given vectors AA' and BB', the vector CC' is obtained by linear interpolation. The field point X moves to X' such that XX' is equal and parallel to CC'. In order to contain the effect of grid movement to a specified region, XX' is multiplied by an exponential factor that decays from unity at the surface to nearly zero at a specified cut-off distance. This technique, in combination with edge swapping, allows for large changes in body shapes even when highly stretched grids are used. However, the grids tend to lose orthogonality near the surface when large changes occur in the surface geometry. To improve orthogonality near the surface, the method described above is replaced by the one shown in Fig. 6(b) within a specified distance to the wall. In this technique, CC' is obtained as before, but C'X' remains orthogonal to A'B' and the normal distance d is maintained. It is also desirable to revert to the inviscid algorithm in regions where the grid is not highly stretched. Therefore, outside another specified distance from the wall, the inviscid algorithm is employed. Thus, the final scheme is a blending of all three methods. This scheme has been found to be effective in dealing with Navier-Stokes grids, even for large-scale changes in surface shape, and is reasonably insensitive to the cutoff distances provided that the region in which orthogonality is maintained is restricted to the immediate vicinity of the wall. Unless the displacements of the surfaces are large, the last step can be skipped.

The technique described above is demonstrated in Fig. 7 for a Navier-Stokes grid about an airfoil. The grid contains 26949 nodes, and the spacing at the wall is 2×10^{-6} relative to the chord. In this figure, the nose of the geometry is distorted by moving one of the Bezier control points in this region. Although the geometry is significantly altered, a valid mesh results, which maintains good quality as well as orthogonality near the surface. It should be pointed out that for multielement configurations, the procedure described may fail for large relative displacements of the elements because the cutoff regions that may be initially distinct could "collide." Further work is necessary in this area.



A) Initial mesh. B) Distorted mesh. C) Close-up in nose region.

Figure 7. Example of mesh movement for viscous mesh.

Optimizer

The optimizer used in the current study is KSOPT,[40] which uses a quasi-Newton method to determine the search directions and a polynomial line search technique to determine the step length in the descent direction. This code has been chosen because it is capable of multipoint design and can handle both equality and inequality constraints. In addition, upper and lower bounds can be placed on design variables; this is the method that is currently used to enforce the geometric constraints necessary to maintain a viable geometry throughout the design cycle.

Results

Accuracy of derivatives

To assess the accuracy of derivatives, an isolated transonic airfoil and a subsonic multielement airfoil (where interaction between the elements occurs through the flow field) are studied. For the first test, a single 12th-order Bezier curve is used to approximate an NACA 0012 airfoil, with only 13 control points. In the experiment that follows, a grid with 4770 nodes is generated, with 128 grid points on the surface of the airfoil. The cost function is the lift coefficient, and derivatives with respect to the Bezier control points are obtained using the continuous adjoint method and are compared with those from finite differences. The Mach number for this case is 0.75, and the angle of attack is 1.25°. The resulting pressure distribution is shown in Fig. 8 and exhibits a shock on the upper surface of the airfoil.

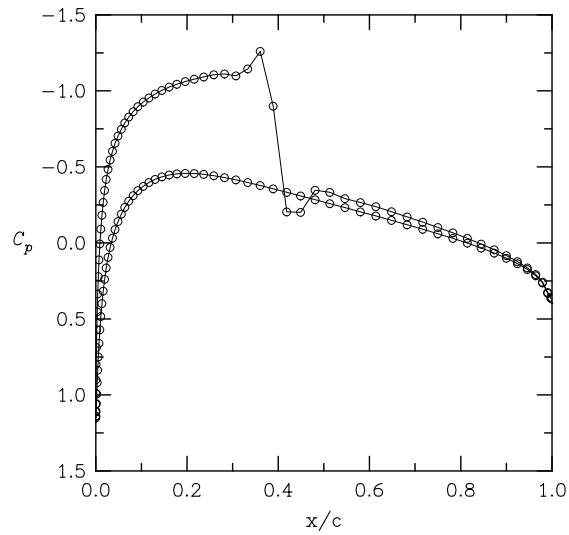


Figure 8. Pressure distribution for NACA 0012 with $M_\infty = 0.75$ and $\alpha = 1.25°$.

The sensitivity derivatives of the lift with respect to the y position of the individual control points are shown in Figs. 9 and 10 using the continuous adjoint approach. In these figures, the derivatives are obtained by using the second-order formulation for both the flow solver and the adjoint equations. The corresponding derivatives for first-order accuracy are not shown because the first-order scheme has been verified to be discretely adjoint to the flow solver in this case. In Fig. 9, the derivatives at the first and last control points (numbers 1 and 13) correspond to those at the trailing edge. Although the derivatives of the control points at the trailing edge are available from the adjoint approach, the corresponding finite-difference derivatives are not obtained because the geometry would "separate" at the trailing edge. Instead, the grid point at the trailing edge is perturbed, and the resulting derivative is compared with the sum of the derivatives at this location from the adjoint approach. A close-up view of the derivatives away from the trailing edge is shown in Fig. 10. The figures indicate that the derivatives are fairly accurate; the largest discrepancy between the adjoint and the finite-difference derivatives is less than 5 percent. Note that in this study specification of the costate variables as a boundary condition across discontinuities in the field, as suggested in Refs. 17 and 18, is not done with no apparent degradation in accuracy.

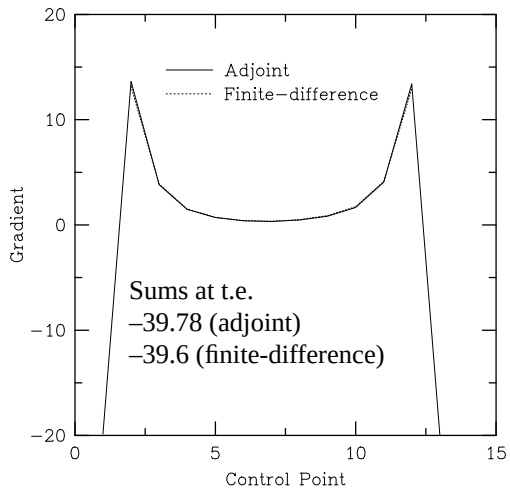


Figure 9. Comparison of derivatives obtained using adjoint approach with finite differences for NACA 0012 with $M_\infty = 0.75$ and $\alpha = 1.25°$.

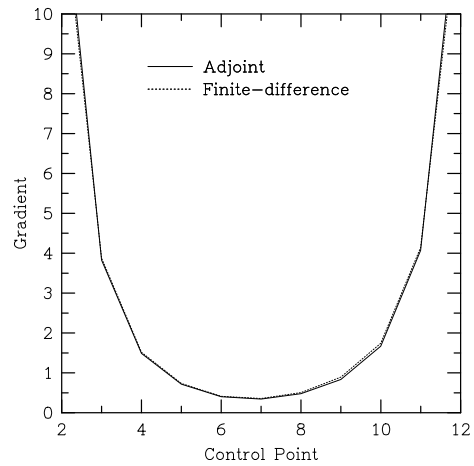


Figure 10. Close-up view of Fig. 9

As mentioned previously, unless the adjoint equations are discretized appropriately, the resulting derivatives may exhibit inaccuracies when compared with finite-difference gradients on coarse grids. To study this aspect more closely, a two-element airfoil is considered for which the surface of each element is represented with a third-order B-spline with 31 control points. The cost function is the lift coefficient, and the derivatives with respect to the design variables on the aft element are computed with both methods on a set of four sequentially finer grids. These grids, denoted as grids 4, 3, 2, and 1, consist of 1103, 3030, 9591, and 18,392 nodes, respectively; of these nodes, 88, 176, 352, and 704 lie on the air-

foil surfaces. Obtaining the sensitivity derivatives with central finite-difference formulas requires 58 flow-field computations for each grid. For the adjoint approach, all derivatives are obtained in one solution of the adjoint equations, which requires roughly the same amount of work as one solution of the flow field.

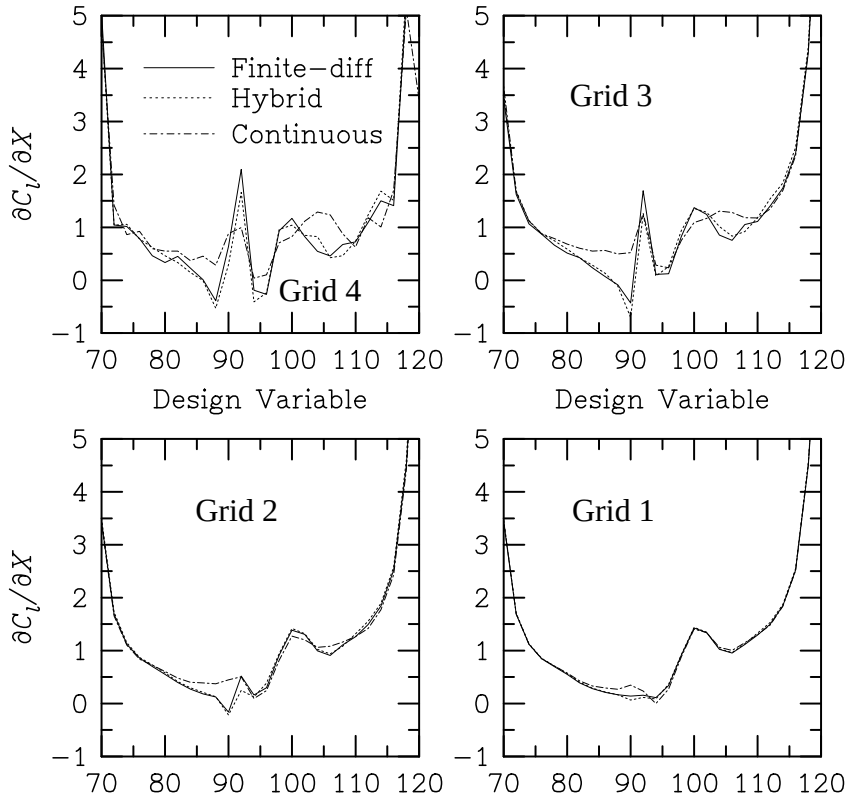


Figure 11. Accuracy of derivatives on aft element of a two-element airfoil; third-order B-spline with 31 control points

In Fig. 11, finite-difference derivatives are compared with those obtained using the formulas for the continuous adjoint approach. In this figure, the derivatives in the immediate vicinity of the trailing edge are not shown so that the derivatives over the bulk of the airfoil can be examined more closely. The importance of the derivatives near the trailing edge is discussed later in this section. In addition, derivatives are also shown from a "hybrid" approach in which the costate variables are obtained from the continuous adjoint approach and are subsequently used in a discrete adjoint framework to compute the sensitivity derivatives by using Eq. (54). In this approach, no approximations are used in Eq. (54), so that the only difference between the hybrid approach and a purely discrete adjoint approach stems from small differences in obtaining the costate variables for the second-order discretization. Recall that in the continuous adjoint case, no sensitivities that result from mesh movement appear in the equations. Therefore, for the hybrid approach, the

mesh sensitivities have been neglected in order to examine the effects. As discussed earlier, when the surface of the airfoil is perturbed, the residuals are affected at the nodes on the body as well as at their first and second nearest neighboring nodes (the second nearest nodes are affected through the gradient computation). Therefore, the residuals at these nodes contribute to the sensitivity-derivative calculation in Eq. (54). However, inclusion of these contributions does not account for any change in the interior residuals caused by the possible movement of interior mesh points. The situation is thus equivalent to the case for which the surface of the airfoil is modified but the interior of the mesh is held fixed.

As seen in the figure, the finite-difference derivatives are nonsmooth on the coarser grids and have several derivatives of negative sign. The derivatives obtained with the hybrid approach follow an almost identical pattern. The derivatives obtained from the continuous adjoint approach are smoother on all of the grids and remain positive over the entire interval shown. Although discrepancies result over parts of the airfoil, the derivatives calculated with all three methods agree as the grids are refined. A case could be made that the continuous adjoint derivatives are "better" because the signs of derivatives are always in "correct" agreement with those from the finest grid. However, when designing on the coarser grids, this could cause the optimizer to fail because the derivatives do not accurately represent the discrete derivatives.[37] Conversely, the hybrid approach may be considered to be "better" in that the derivatives agree more closely with those obtained from finite differences on all the meshes. Although this may lead to successful numerical optimization on all grids, the resulting geometry may be quite different from that obtained with a finer grid. In either case, a suitably refined grid must be employed in which case neither the continuous nor the discrete approach offers a significant advantage over the other.

In Fig. 11, the discrepancies in the derivatives on the coarse grids stem from three sources. These include the fact that the second-order scheme is not exactly discretely adjoint to the flow equations on all grids. Also, small errors in the finite-difference calculations may be present as a result of the choice of step size which was not optimized for accuracy for each of the 29 design variables although a reasonable effort was made to determine acceptable values. In addition, the derivatives obtained from finite differences include the effect of grid sensitivities because the interior mesh points are relaxed each time a design variable is perturbed using the techniques described earlier. As mentioned previously, neither the continuous adjoint nor the hybrid approach has included these effects because the continuous formulation assumes no dependence on a grid and the hybrid formulation has neglected these contributions for this test. The figure shows clearly that as the grids are refined the derivatives over the bulk of the airfoil approach the same value regardless of the methodology used to obtain them.

Although it is tempting to conclude from the above example that grid sensitivities do not play a major role as the grid is refined, this conclusion is not always valid. To demonstrate, a simple example is given in which the geometry and flow conditions are held fixed while the grid is allowed to change. More specifically, the relationship between the airfoil surface and the grid is changed. The role of the grid sensitivities is studied by considering the derivative of the lift of a single airfoil due to a vertical translation.

For this case, an NACA 0012 airfoil at a free-stream Mach number of 0.5 and an angle of attack of 2° is considered. A sequence of structured C-type grids is utilized in which

each grid represents a uniform refinement in each direction over the previous level. Two structured-grid codes[39,41] are used, in addition to the unstructured-grid flow solver. For the unstructured flow solver, the cells in the structured mesh are simply divided into triangles. The derivative of the lift with respect to translation of the airfoil surface in the y direction is computed with central differences. The airfoil surface is perturbed a small amount, and three different techniques are considered for modifying the interior mesh:

1. The airfoil surface and the entire mesh are shifted.
2. The airfoil surface is perturbed, and the rest of the mesh remains fixed.
3. The airfoil surface, as well as the mesh line that extends from the trailing edge of the airfoil to the downstream outer boundary are perturbed, and the rest of the grid remains fixed.

For the case in which the airfoil and the mesh are simultaneously perturbed, the lift does not change and the derivative is zero, independent of the mesh size. This case corresponds to simply a shifting of the origin of the coordinate system; therefore, no calculations are required. The importance of the second method for computing the finite-difference derivative is that this situation corresponds to the case in which grid sensitivities are ignored in a discrete formulation. This correspondence has been verified using the derivatives obtained from the first-order adjoint code, where the derivatives are obtained by using the hybrid methodology and the grid sensitivities are neglected. The third method is chosen simply for demonstration purposes. Note that in the numerical experiments that follow, all results are converged to machine zero and the step size for computing the finite-difference derivatives has been varied over a large range of values with no significant changes in the results. In all cases, the step size that is used is much smaller than the distance from the surface of the airfoil to the first grid line, so no crossing of grid lines occurs.

In an ideal situation, the lift of a single airfoil in an unbounded flow would be insensitive to a vertical change in the coordinates so that the derivative would be zero. Numerically, however, changes may occur because of the changing location of the airfoil relative to the outer boundary and because of possible changes in the grid. In the case where the entire grid is shifted, the derivative of lift due to a shift in the y location of the surface is zero. By shifting all grid lines except the one at the outer boundary, the derivatives have been found to remain very small ($O(10^{-5})$) which indicates that the derivative of lift due to the location of the outer boundary is small. In this case, the changes are not only attributable to the changing location of the outer boundary but also to some small grid effects at the outer boundary.

In Fig. 12, the sensitivity derivatives of the lift with respect to translation of the airfoil in the y direction are shown for methods 2 and 3 described above. As seen in the figure, the derivatives due to the translation of the airfoil surface depend greatly on the methodology used to modify the grid. More importantly, these derivatives do not tend to zero as the mesh is refined but actually increase in magnitude!

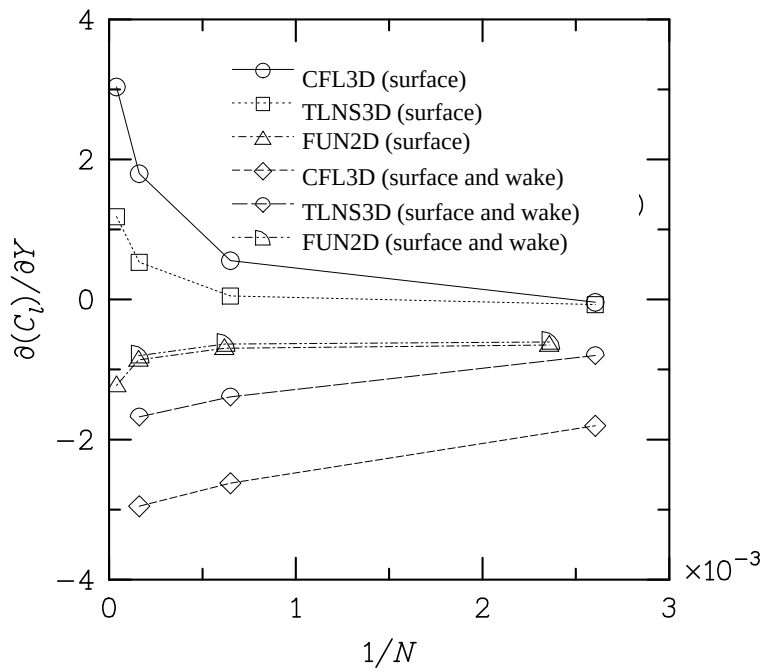


Figure 12. Finite-difference derivatives of lift with respect to vertical shift in airfoil position obtained with fixed computational grid.

Computing the derivative of lift with respect to a vertical translation corresponds to a simple summation of the derivatives of lift with respect to the y position of each of the design variables. For example, $\partial c_l/\partial Y$ for the aft element of the airfoil shown in Fig. 11 can be computed by a summation of the individual derivatives. Although the individual derivatives shown in Fig. 11 converge as the mesh is refined, the derivatives at the trailing edge do not. In Fig. 13, the individual sensitivity derivatives that are obtained with the hybrid approach are now plotted at a scale so that the derivatives at the trailing edge can be seen. Whereas the derivatives away from the trailing edge converge as the grid is refined (see Fig. 11), those at the trailing edge of the airfoil do not exhibit the same level of convergence and, in fact, continually change as the grid is refined. This behavior appears to be caused by the singularity at the trailing edge and is the source of the sensitivity of the derivative to the manner in which the grid is treated.

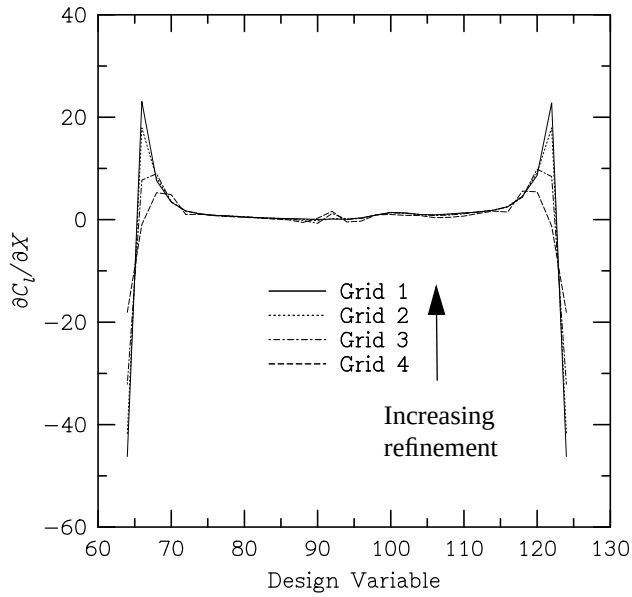


Figure 13. Sensitivity derivatives.

In Fig. 14, finite-difference derivatives similar to those shown in Fig. 12 are shown for a symmetric Joukowski airfoil at the same Mach number and angle of attack as before. For this airfoil, the slope of the upper and lower surfaces in the analytical definition are identical at the trailing edge, and the effect of the singularity should be reduced. These derivatives have been obtained by shifting only the surface of the grid, as in method 2. As is clearly seen in the figure, the derivatives are much smaller in magnitude than those for the NACA 0012 and do not increase in magnitude as the grid is refined.

From the foregoing discussion, it is apparent that the grid sensitivities near the trailing edge of the airfoil can play a major role in the computation of the derivatives necessary to position airfoils relative to one another. It should be emphasized that during an actual design the grid is generally "relaxed," so that the original relationships between the grid points are more or less intact, and that the effect of the grid would be much less pronounced than that shown above. The important point is that without inclusion of the grid sensitivities, the derivatives obtained would correspond to the case above in which the interior grid is held fixed. Because the derivatives clearly depend on the manner in which the mesh and the geometry interact, this factor must be accounted for in the computations when derivatives are needed in the immediate vicinity of the trailing edge. Furthermore, the errors caused by failure to properly account for these terms do not vanish as the mesh is refined. However, from the results in Fig. 11, it appears that grid sensitivities can be safely neglected in regions away from the trailing edge, provided that the grid is sufficiently refined.

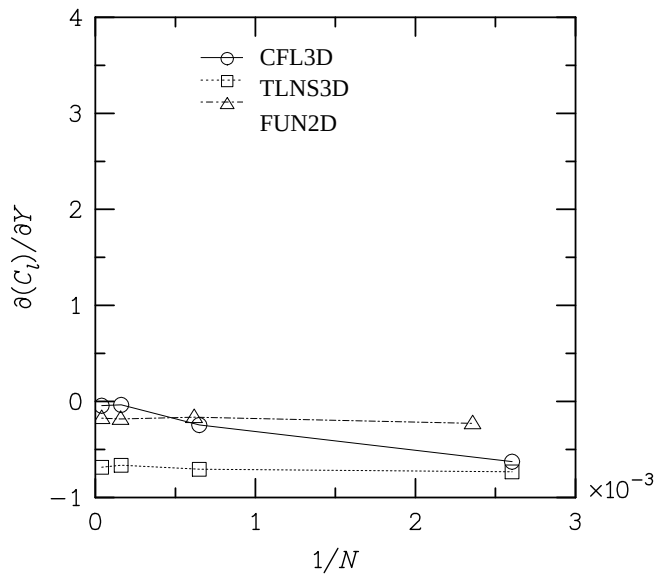


Figure 14. Translation for Joukowski airfoil.

Inviscid Design Examples

An example of shape optimization is shown below in which drag minimization has been performed for a single airfoil. The initial geometry is an NACA 0012 airfoil, described by a third-order B-spline with 50 control points. The grid consists of 4763 nodes with 128 nodes on the airfoil surface. The Mach number for this test is 0.75, and the initial angle of attack is 2°. For this case, the computed lift coefficient is 0.4229, with a corresponding drag coefficient of 0.0123. For this design, the cost associated with maintaining the current lift coefficient is combined with that for minimizing the drag:

$$I_c(\mathbf{Q}, \mathbf{D}) = \frac{1}{2}(c_l - c_l^*)^2 + 10 \times \frac{1}{2}(c_d - c_d^*)^2 \tag{56}$$

where c_l^* is the desired lift coefficient and c_d^* is zero. The factor of 10 associated with the cost function for drag is chosen so that the contribution from each cost function is of the same order of magnitude. The design variables are the y-coordinates of the control points that describe the airfoil, except those at the trailing edge, which remain fixed. The angle of attack is an additional design variable and is allowed to vary in order to maintain the lift. The total number of design variables for this case is 49. For this case, the continuous adjoint approach is used instead of the hybrid approach.

After 10 design iterations, the lift coefficient is 0.4225, which is in close agreement with the specified lift coefficient of 0.4229. The drag has been reduced from 0.0123 to 0.0016, and the final angle of attack is 1.747°. The objective function and the root mean

square (rms) of the sensitivity derivatives have each been reduced between 1 and 2 orders of magnitude. Note that these gradients are not the projected gradients and that several side constraints are active. The initial and final pressure distributions are shown in Fig. 15; the corresponding geometries are shown in Fig. 16.

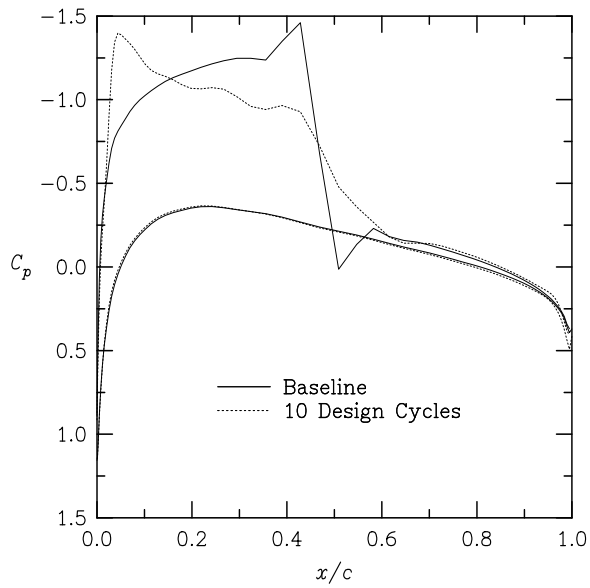


Figure 15. Initial and final pressure distribution for NACA 0012 design case.

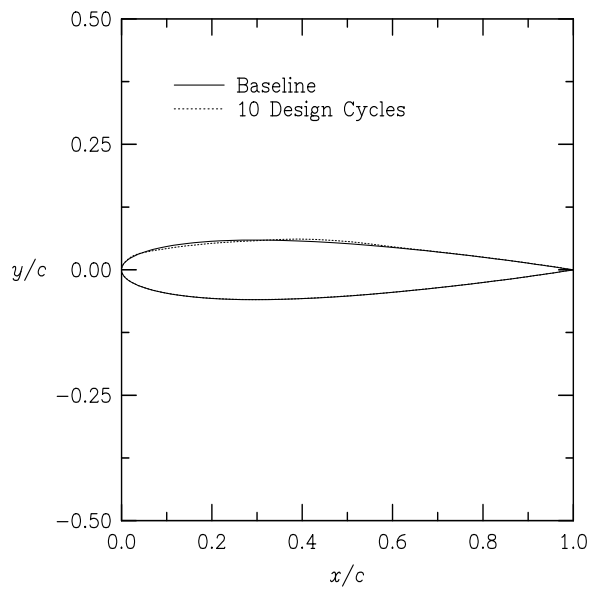


Figure 16. Initial and final airfoil for NACA 0012 design case.

The next case is that of a two-element airfoil configuration that consists of two airfoils in which the top airfoil is displaced from the other in the positive y direction by 0.5 chord and in the negative x direction by 0.5 chord. The free-stream Mach number is 0.60, and the angle of attack is 0°. A sequence of three grids for use with multigrid acceleration has been generated for this case. The finest grid consists of 7974 points and is shown in Fig. 17.

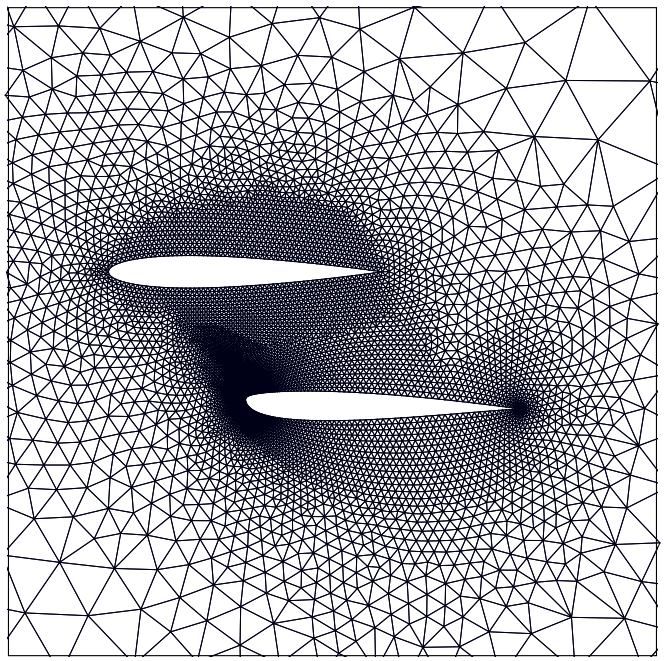


Figure 17. Initial configuration for two-element test case.

For this case, the objective is to modify the shape of the aft airfoil in order to achieve a desired pressure distribution on the front airfoil. The desired pressure distribution has been obtained from analysis of the initial configuration, with the shape of the aft airfoil modified. Although this test case is somewhat fabricated, it demonstrates flexibility that may be difficult to achieve with inverse methods in which the interaction between elements is not taken into account.

Pressure contours for the initial flow field are shown in Fig. 18(a); the corresponding contours of Ψ_2 are shown in Fig. 18(b). The pressure contours indicate the presence of a shock between the two airfoils, with a Mach number ahead of the shock on the lower airfoil of approximately 1.25. The costate variables shown in the accompanying figure, on the other hand, exhibit a shocklike structure in a location that corresponds to the sonic line in the flow field. However, in designing for other objective functions, the contours of the costate variables change and do not necessarily show such a clear correspondence with the flow field. For example, if the cost function is zero at the design point in an unconstrained optimization, the costate variables are all zero independent of the flow field, due to the homogeneity of the boundary conditions.

The initial and final pressure distributions on the surface of the airfoils are shown in Fig. 19; the pressure contours after three design cycles are shown in Fig. 20. As seen from Fig. 19, the pressure distribution obtained after three design iterations agrees closely with that desired. The cost function has been reduced over 3 orders of magnitude, and the rms of the sensitivity derivatives has also been reduced over 3 orders of magnitude after the second design cycle. The final pressure distribution on the aft element does not exhibit the strong shock that is initially present.

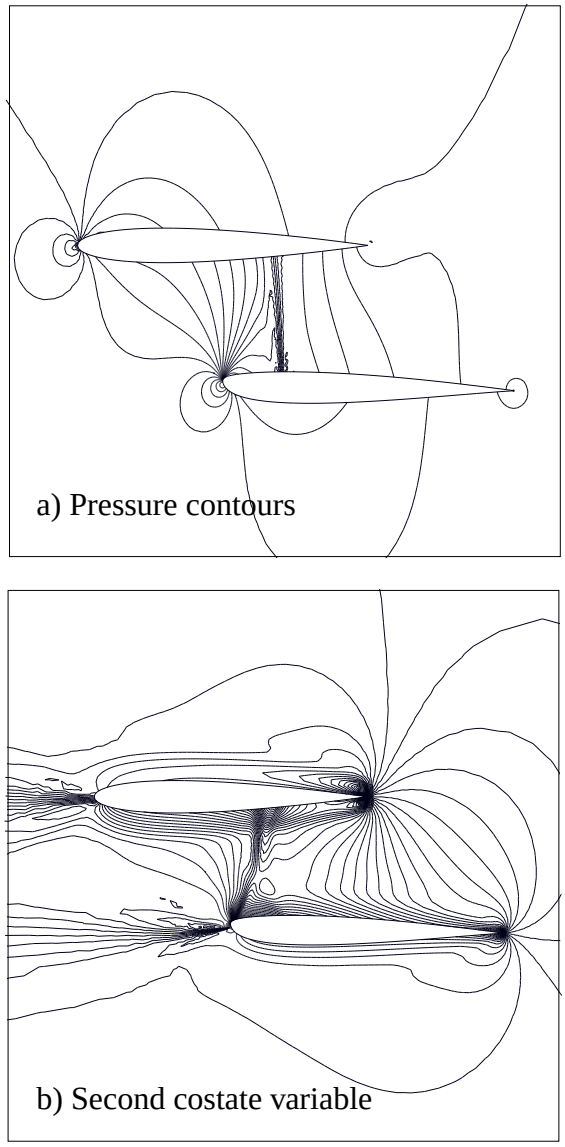


Figure 18.Contours of pressure coefficient and Ψ_2 for two-element configuration.

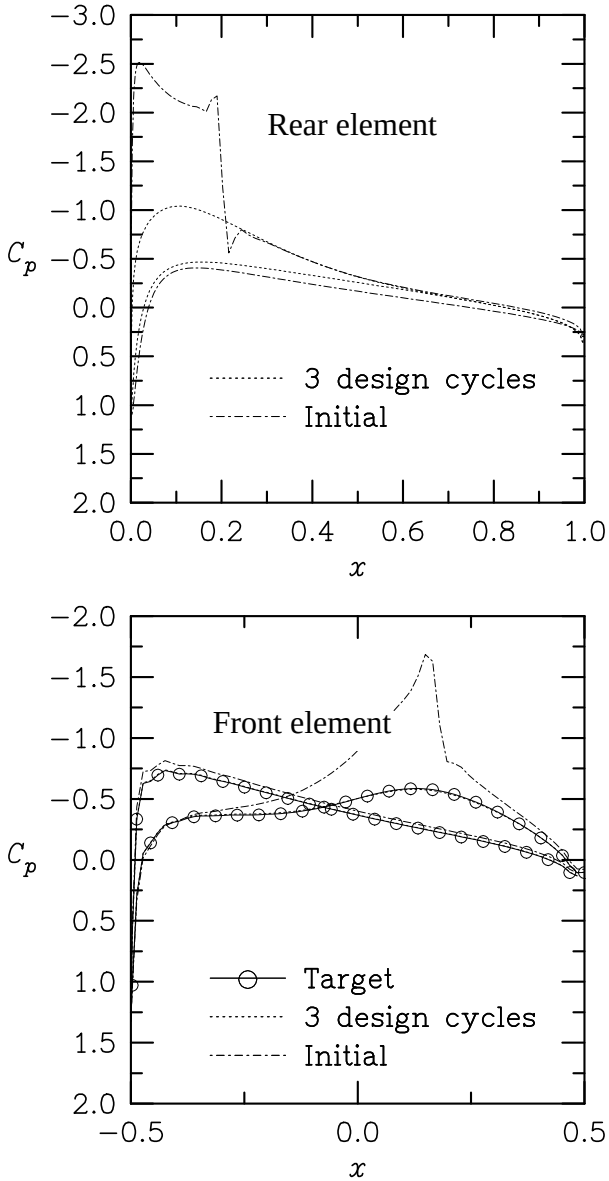


Figure 19. Initial and final pressure distributions for two-element test case.

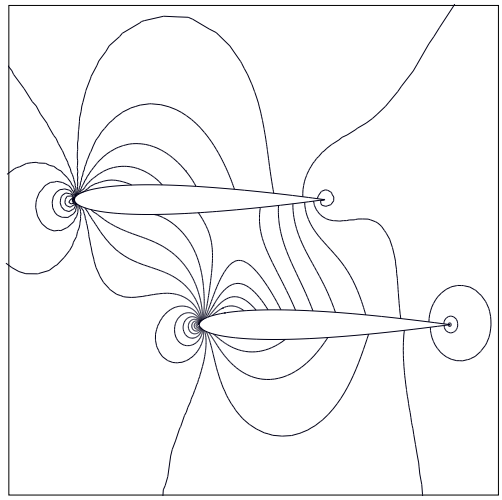


Figure 20. Final airfoils and pressure coefficient for two-element airfoil.

Viscous Design Examples

For the first viscous case, the objective is to maximize the lift of an isolated airfoil by modifying the shape, with the angle of attack held constant. An initial computation has been performed for an airfoil at a Reynolds number of 5000 and an angle of attack of 2°. The mesh used for this computation has 6951 nodes of which 128 lie on the surface of the airfoil. The airfoil geometry is described by using a 12th-order Bezier representation similar to that described earlier, except that several of the control points have been modified so that the airfoil is no longer symmetric (see Fig. 21). For this case, nine design variables have been used. These correspond to the y-coordinates of the control points away from the immediate vicinity of the trailing edge. The initial lift coefficient is 0.0950, and the initial drag coefficient is 0.0545. After three design cycles, the lift has been increased to 0.2571 and 5 of the nine design variables have hit their imposed side constraints. Although no constraint or objective was placed on the drag, the drag coefficient has dropped to 0.0509. Note that for this case both the initial and final configurations have a small separated region that extends over the last 25 percent of the airfoil. Despite the presence of separation, a steady flow field is obtained. In the event of unsteady separation, the adjoint approach as described would not be applicable because the steady-state residual is assumed to be zero and is used as a constraint for the optimization.

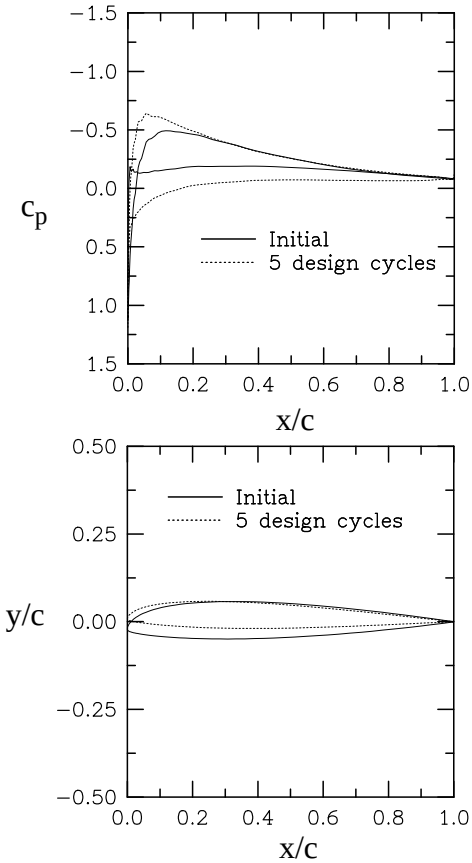


Figure 21. Viscous design for maximizing lift.

For the final case shown in Fig. 22, the objective is to match a desired pressure distribution that has been obtained from a previous analysis of an NACA 0012 airfoil. The initial airfoil geometry has been obtained by simply displacing several of the B-spline control points that define the original airfoil. The Reynolds number is 5000, based on the chord of the airfoil, and the angle of attack is held fixed at 2°. The mesh used for this is similar to that used in the previous test case and has approximately 7000 nodes. For the current test, the cost function has been reduced by 4.5 orders of magnitude after three design cycles, and the gradients have been reduced by 1.5 orders of magnitude after the second design cycle. As seen in Fig. 22, the target pressure distribution is obtained, and the final airfoil shape is that of an NACA 0012 airfoil.

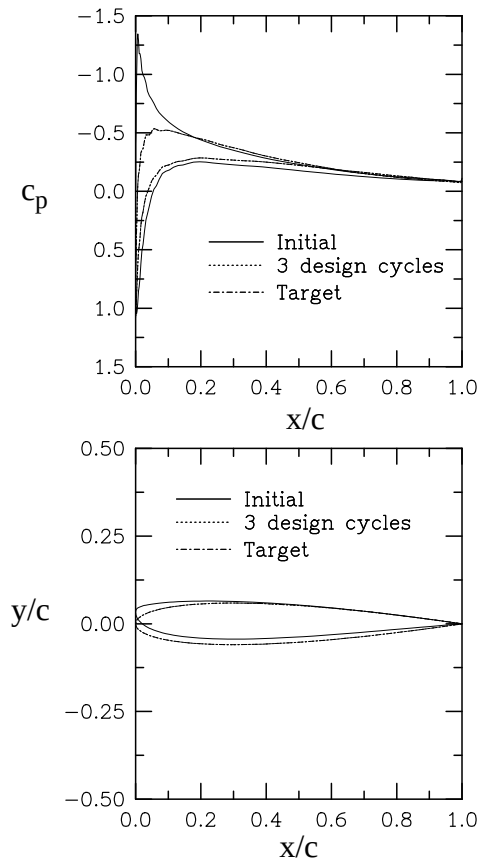


Figure 22. Initial and final pressure distribution for viscous flow.

Discussion and Conclusions

The purpose of the present investigation has been to develop and analyze the continuous adjoint approach for obtaining sensitivity derivatives on unstructured grids for the Euler and Navier-Stokes equations. During the course of the study, several drawbacks have been uncovered. The most significant is the need for accurate second derivatives of the velocities required for computing the shape sensitivity derivatives for viscous flows. In general, consistent second derivatives cannot be obtained with spatially second-order accurate schemes. This problem can be circumvented by mapping the domain to a fixed computational coordinate system as is usually employed for structured grids. This approach, however, is restrictive in its generality and is at odds with the flexibility offered by unstructured grids. The absence of a mapping is a fundamental difference between structured and unstructured grids. The requirement for second derivatives can also be overcome by considering a higher order discretization of the flow field, so that consistent second derivatives can be obtained. However, this represents a significant level of effort because the entire flow field must be computed to higher order accuracy. It appears that the most

expedient and cost-effective means for alleviating this problem is to essentially abandon the purely continuous adjoint approach in favor of a more discrete approach, as described in the present paper. This approach has the added benefit that the contributions to the sensitivity derivatives due to the grid may be included. These terms do not naturally appear in the continuous framework unless a mapping to a fixed computational domain is first employed. However, it is shown in this paper that these terms are critical in obtaining accurate derivatives for geometries with singularities.

Another possible drawback of the continuous approach is that a restriction is placed on the allowable cost functions that can be used. This restriction stems from the need for a suitable balance between terms in the cost function and corresponding terms from the residual that are used to eliminate variations in pressure or velocity gradients. For inviscid flows, the allowable cost functions are those that involve only the pressure. For viscous flows, an additional requirement is that terms from the entire stress tensor, including both the pressure and viscous terms, must be included. Although this limitation does not appear to occur in a discrete adjoint approach, the full implications remain unclear.

The continuous adjoint approach requires more "up front" derivations than the discrete approach before a computer implementation can be pursued. Also, each new cost function requires a certain level of effort to not only arrive at the appropriate boundary conditions but to determine whether boundary conditions can even be obtained. On the other hand, the continuous approach may provide insight into which cost functions are controllable. For example, in the case of inviscid compressible flow, pressure is the only surviving term in the boundary flux upon application of the flow tangency condition. Therefore, it stands to reason that only cost functions that involve pressure can be controlled. In the discrete adjoint approach, new cost functions are more easily added because they enter the problem only through the right-hand side of a linear system of equations. After a subroutine has been written to evaluate a cost function, it is usually a simple matter to obtain all the necessary derivatives by differentiating the code directly using the chain rule. Furthermore, this procedure does not require detailed knowledge of the equations and can be accomplished by using a computational tool such as ADIFOR.[8]

A technique is presented in this paper that is derived from a continuous adjoint approach but appeals to the discrete approach where expedient. A discretization of the adjoint equations for viscous and inviscid flow is presented that corresponds exactly to a discrete adjoint formulation for first-order spatial accuracy. The discretization differs from the discrete adjoint approach for higher order schemes only in the artificial dissipation terms. This approach is simple to implement and yields derivatives that are reasonably accurate in comparison with finite-difference calculations, even on coarse grids. Alternatively, the same scheme could be obtained from a discrete adjoint point of view by appealing to the continuous approach for making suitable approximations. The adjoint approach is coupled with an optimization algorithm and is augmented with a mesh movement strategy for restructuring the mesh in response to surface displacements. The mesh movement technique is applicable for meshes used in inviscid computations as well as for meshes with high aspect ratio triangles typically used in viscous computations. The resulting approach has been used in several design examples.

Acknowledgments

The authors would like to thank Antony Jameson, Geojoe Kuruvila, Manny Salas, Shlomo Ta'asan, and Jim Thomas, for many useful discussions.

References

[1]Anderson, W. K. and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[2]Anderson, W. K, Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," AIAA 95-1740, June 1995.

[3]Angrand, F., "Optimum Design for Potential Flows," *Int. J. Num. Methods in Fluids*, Vol. 3, 1983, pp. 265–282.

[4]Barth, T. J. "Analysis of Implicit Local Linearization Techniques for TVD and Upwind Algorithms," AIAA 87-0595, 1987.

[5]Baysal, O. and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *J. of Fluids Engineering*, Vol. 113, 1991, 681-688.

[6]Beux, F. and Dervieux, A., "Exact-Gradient Shape Optimization of a 2-D Euler Flow," *Finite Elements in Analysis and Design*, Vol. 12, 1992, 281-302.

[7]Beux, F. and Dervieux, A., "A Hierarchical Approach for Shape Optimization," INRIA Report 1868, February 1993.

[8]Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P., "ADIFOR: Generating Derivative Codes from Fortran Programs," ADIFOR Working Note No. 1, Argonne Preprint MCS-P263-0991, Argonne National Laboratory, September 1991.

[9]Bonhaus, D. L., "An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes," M.S. Thesis, George Washington University, 1993.

[10]Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Wings Using Discrete Sensitivity Analysis," AIAA J., Vol. 34, No. 9, September 1996, pp. 1761-1770.

[11]Cabuk, H. and Modi, V., "Shape Optimization Analysis: First- and Second-Order Necessary Conditions," *Optimal Control Applications and Methods*, Vol. 11, 1990, pp. 173–190.

[12]Cabuk, H. Sung, C.-H., and Modi, V., "Adjoint Operator Approach to Shape Design for Internal Incompressible Flows," *Proceedings of the Third International Conference on Inverse Design and Optimization in Engineering Sciences*, G. S. Dulikravich, ed., 1991, pp.

391-404.

[13]Elliott, J. and Peraire, J., “Aerodynamic Design using Unstructured Meshes,” AIAA 96-1941, 1996.

[14]Farin, Gerald, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, Boston, 1993.

[15]Hou, G. J.-W., Taylor, A. C., and Korivi, V. M., “Discrete Shape Sensitivity Equations for Aerodynamic Problems,” *Int. J. for Numerical Methods in Engineering*, Vol. 37, 1994, pp. 2251–2266.

[16]Huffman, W. P., Melvin, R. G., Young, D. P., Johnson, F. T., Bussoletti, J. E., Bieterman, M. B., and Hilmes, C. L., “Practical Design and Optimization in Computational Fluid Dynamics,” AIAA 93-3111, Presented at the AIAA 24th Fluid Dynamics Conference, July 6-9, 1993.

[17]Iollo, A., Salas, M., and Ta’asan, S., “Shape Optimization Governed by the Euler Equations Using an Adjoint Method,” ICASE report 93-78, November 1993.

[18] Iollo, A. and Salas, M., “Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks,” J. Comp. Phys., Vol. 125, 1996, pp. 124-134.

[19]Jameson, A., “Aerodynamic Design Via Control Theory,” *J. Scientific Computing*, Vol. 3, 1988, pp. 23–-260.

[20]Jameson, A., “Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag,” MAE Report 1881, Presented at the 31st Israel Annual Conference on Aviation and Aeronautics, February, 1990.

[21]Jameson, A., “Optimum Aerodynamic Design Using CFD and Control Theory,” *AIAA 12th Computational Fluid Dynamics Conference*, AIAA 95-1729-CP, June 1995.

[22]Jameson, A.,”The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics,” Proceeding of the AGARD 77th Fluid Dynamics Panel Symposium, Seville, Spain, 1995.

[23]Korivi, V. M., Taylor, A. C., Newman, P. A., Hou, G. J.-W., and Jones, H. E., “An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives,” AIAA 92-4746, AIAA/USAF/NASA/OAI 4th Symposium on Multidisciplinary Analysis and Optimization, September 21-23, 1992.

[24]Kuruvila, G., Ta’asan, S., and Salas, M., “Airfoil Optimization by the One-Shot Method,” AGARD-FDP-VKI, Special Course on Optimum Design Methods in Aerody-

namics, April 1994.

[25]Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, New York, 1971.

[26]Marco, N. and Beux, F., "Multilevel Optimization: Application to One-Shot Shape Optimum Design," INRIA Report 2068, Oct. 1993.

[27]Marco, N. and Dervieux, A. "Multilevel Parametrization for Aerodynamical Optimization of 3D Shapes," INRIA Report 2949, July 1996.

[28]Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA 95-0212, Jan. 1995.

[29]Pironneau, O., "On Optimum Design in Fluid Mechanics," *J. Fluid Mech.*, Vol. 64, 1974, pp. 97–110.

[30]Pironneau, O., "On Optimum Profiles in Stokes Flow," *J. Fluid Mech.*, Vol. 59, 1973, pp. 117–128.

[31]Pironneau, O., *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, New York, 1982.

[32]Reuther, J. and Jameson, A., "Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory," AIAA 95-0123, January 1995.

[33]Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA 96-0094, January 1996.

[34]Roe, P. L., "Approximate Riemann Solver, Parameter Vectors and Difference Schemes," *J. Comput. Phys.*, Vol. 43, No. 2, 1981, pp. 357–372.

[35]Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, 1986, pp. 856-869.

[36]Sagan, H., *Introduction to the Calculus of Variations*, Dover Publications, Inc., New York, 1992.

[37]Shubin, G. R. and Frank, P. D., "A Comparison of the Implicit Gradient Approach and the Variational Approach to Aerodynamic Design Optimization," Boeing Computer Services Report AMS-TR-163, April 1991.

[38]Ta'asan, S. and Kuruvila, G., "Aerodynamic Design and Optimization in One Shot," AIAA 92-0025, January 1992.

[39]Thomas, J., Krist, S., and Anderson, W. K., "Navier-Stokes Computations of Vortical

Flows Over Low-Aspect-Ratio Wings," *AIAA J.*, Vol. 28, No. 2, 1990, pp. 205–212.

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

[41]Vatsa, V. N., Sanetrik, M. D., and Parlette, E. B., "Development of a Flexible and Efficient Multigrid Based Multiblock Flow Solver," AIAA 93-0677, January 1993.

[42]Venkatakrishnan, V. and Mavriplis, D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," J. Comp. Phys., Vol. 127, 1996, pp. 380-397.