AIAA JOURNAL
Vol. 37, No. 2, February 1999

Airfoil Design on Unstructured Grids for Turbulent Flows

W. Kyle Anderson* and Daryl L. Bonhaus†
NASA Langley Research Center, Hampton, Virginia 23681-0001

An aerodynamic design algorithm for turbulent flows using unstructured grids is described. The current approach uses adjoint (costate) variables to obtain derivatives of the cost function. The solution of the adjoint equations is obtained by using an implicit formulation in which the turbulence model is fully coupled with the flow equations when solving for the costate variables. The accuracy of the derivatives is demonstrated by comparison with finite difference gradients, and a few sample computations are shown. Recommendations on directions of further research into the Navier–Stokes design process are made.

Nomenclature

A	= area of control volume
a	= speed of sound
C^*	= constant used in Sutherland's law for viscosity
$c_{b_1}, c_{b_2}, c_{v_1}, c_{w_1}, c_{w_2}, c_{w_3}$	= constants used in Spalart–Allmaras turbulence model
c_d	= drag
c_l	= lift
c_1	= lift coefficient, drag coefficient
$\boldsymbol{D}$	= vector of design variables
D_i	= component i of design variable vector
d	= distance to nearest surface
E	= total energy per unit volume
$\boldsymbol{F}$	= fluxes of mass, momentum, and energy
$\boldsymbol{F}_i$	= inviscid contribution to fluxes
$\boldsymbol{F}_v$	= viscous contribution to fluxes
$\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{f}_v, \boldsymbol{g}_v$	= components of viscous fluxes
$f_{v_1}, f_{v_2}, f_w, f_{t_1}, f_{t_2}$	= functions used in the turbulence model
I	= augmented cost to be minimized
I_c	= cost to be minimized
M_∞	= freestream Mach number
N	= B-spline basis functions
$\hat{\boldsymbol{n}}$	= unit normal to boundary of control volume
Pr	= Prandtl number
Pr_t	= turbulent Prandtl number
p	= pressure
$\boldsymbol{Q}$	= vector of dependent variables
q_x, q_y	= components of heat flux
$\boldsymbol{R}$	= residual for a control volume
Re	= Reynolds number
S	= magnitude of vorticity
s	= parameterization variable for B-splines
T	= temperature
t	= time
$\boldsymbol{U}$	= magnitude of velocity
u, v	= Cartesian components of velocity
$\boldsymbol{X}$	= grid-point locations
x, y	= Cartesian coordinates
y^+	= wall coordinate
α	= angle of attack
γ	= ratio of specific heats
$\partial\Omega$	= boundary of control volume
κ	= Karman constant
μ	= laminar viscosity
μ_t	= turbulent viscosity
ν	= μ/ρ
ν_t	= μ_t/ρ
$\tilde{\nu}$	= dependent variable for turbulence model
ρ	= density
σ	= constant for turbulence model
$\tau_{xx}, \tau_{xy}, \tau_{yy}$	= shear stress terms
Ψ	= costate variables

Subscript

∞	= freestream quantities

Superscripts

$\hat{}$	= dimensional quantity
$\tilde{}$	= variation

Introduction

BECAUSE of rapid advances in computer speeds and improvements in flow-solver and grid-generational gorithms, a renewed emphasis has been placed on extending computational fluid dynamics beyond its traditional role as an analysis tool to design optimization. Among the methodologies often employed are gradient-based techniques, in which a specified objective is minimized. In this framework, the gradients of the objective function with respect to the design variables are used to update the design variables in a systematic manner to reduce the cost function and to arrive at a local minimum. Many techniques have been used to obtain the necessary derivatives, including finite differences, direct differentiation, and adjoint methods. Many of the methodologies and implementations are discussed in Refs. 1–18.

Although most of the preceding references deal with inviscid flows, a few have addressed viscous computations of turbulent flows. In Ref. 9, Hou et al. used a direct differentiation approach in which the derivatives of the flow solver were obtained with ADIFOR.[19] In Ref. 9, the turbulence model used was the Baldwin–Lomax[20] algebraic model, which was differentiated along with the flow equations. Jameson[12] recently developed a design methodology for turbulent flows based on an adjoint formulation. Here, the Baldwin–Lomax turbulence model was also employed but was assumed constant and was, therefore, not differentiated. This same assumption was also recently used in the work of Soemarwoto.[18]

For unstructured grids, the work to date has been primarily focused on inviscid computations in both two and three dimensions.[1,4,7,21–23] In Ref. 1, the adjoint equations and boundary conditions were derived for the incompressible Navier–Stokes equations, and some design examples were demonstrated. A discrete adjoint implementation for the compressible Navier–Stokes equations has been presented subsequently in Ref. 24. However, in Refs. 1 and 24, turbulence effects were not included. In the work of Mohammadi,[25]

Received Jan. 21, 1998; accepted for publication Aug. 18, 1998.

*Senior Research Scientist, Aerodynamics and Acoustic Methods Branch. Associate Fellow AIAA.

†Research Scientist, Aerodynamics and Acoustic Methods Branch. Member AIAA.

two-dimensional Navier–Stokes results were presented in which turbulence effects were included using a κ–ε turbulence model in conjunction with wall functions. In this reference,[25] automatic differentiation was used to differentiate the necessary components of the flow solver. In Ref. 26, a methodology for design using turbulent Navier–Stokes is described using a one-equation turbulence model that is integrated to the wall. However, in this reference,[26] the effects of interior mesh sensitivities were neglected, which could cause failure of the optimization, particularly on coarser meshes and for cases involving translation/rotation of geometries.

The purpose of the present study is to further extend the work in Refs. 1 and 26 to the compressible Navier–Stokes equations, including a fully coupled field-equation turbulence model. A fully discrete adjoint approach is used that includes the effects of the interior mesh sensitivities in the formulation. The methodology is discussed, and the accuracy of the derivatives is established. A few design examples are given to demonstrate the technology.

Governing Equations

Flow Equations

The governing equations are the time-dependent Reynolds-averaged Navier–Stokes equations. The equations are expressed as a system of conservation laws that relate the time rate of change of mass, momentum, and energy in a control volume of area A to the spatial fluxes of these quantities through the volume. The equations (nondimensionalized by freestream density, speed of sound, temperature, viscosity, thermal conductivity, and a reference length) are given as

$$A\frac{\partial \boldsymbol{Q}}{\partial t} + \oint_{\partial\Omega} \boldsymbol{F}_i \cdot \hat{\boldsymbol{n}}\, \mathrm{d}l - \oint_{\partial\Omega} \boldsymbol{F}_v \cdot \hat{\boldsymbol{n}}\, \mathrm{d}l = 0 \tag{1}$$

where $\hat{\boldsymbol{n}}$ is the outward-pointing normal to the control volume. The vector of dependent variables $\boldsymbol{Q}$ and the flux vectors $\boldsymbol{F}_i$ and $\boldsymbol{F}_v$ are given as

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \tag{2}$$

$$\boldsymbol{F}_i = \boldsymbol{f}\hat{\boldsymbol{i}} + \boldsymbol{g}\hat{\boldsymbol{j}} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix}\hat{i} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}\hat{j} \tag{3}$$

and

$$\boldsymbol{F}_v = \boldsymbol{f}_v\hat{\boldsymbol{i}} + \boldsymbol{g}_v\hat{\boldsymbol{j}} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}\hat{i} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{bmatrix}\hat{j} \tag{4}$$

Here, $\boldsymbol{F}_i$ and $\boldsymbol{F}_v$ are the inviscid and viscous flux vectors, respectively; the shear stress and heat conduction terms are given as

$$\tau_{xx} = (\mu + \mu_t)(M_\infty/Re)\tfrac{2}{3}(2u_x - v_y) \tag{5}$$

$$\tau_{yy} = (\mu + \mu_t)(M_\infty/Re)\tfrac{2}{3}(2v_y - u_x) \tag{6}$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t)(M_\infty/Re)(u_y + v_x) \tag{7}$$

$$q_x = -\frac{M_\infty}{Re(\gamma-1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial a^2}{\partial x} \tag{8}$$

$$q_y = -\frac{M_\infty}{Re(\gamma-1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial a^2}{\partial y} \tag{9}$$

The equations are closed with the equation of state for a perfect gas:

$$p = (\gamma-1)\left[E - \rho\frac{(u^2+v^2)}{2}\right] \tag{10}$$

and the laminar viscosity is determined through Sutherland's law:

$$\mu = \frac{\hat{\mu}}{\hat{\mu}_\infty} = \frac{(1+C^*)(\hat{T}/\hat{T}_\infty)^{\frac{3}{2}}}{\hat{T}/\hat{T}_\infty + C^*} \tag{11}$$

where $C^* = 198.6/460.0$ is Sutherland's constant divided by a freestream reference temperature, which is assumed to be 460°R.

Turbulence Model

For the current study, the turbulence model of Spalart–Allmaras[27] is used. This is a one-equation turbulence model given as

$$\frac{\mathrm{D}\tilde{\nu}}{\mathrm{D}t} = \frac{M_\infty}{\sigma Re}\left(\nabla\cdot\left\{\left[\nu + (1+c_{b_2})\tilde{\nu}\right]\nabla\tilde{\nu}\right\} - c_{b_2}\tilde{\nu}\nabla^2\tilde{\nu}\right)$$
$$- \frac{M_\infty}{Re}\left(c_{w_1}f_w - \frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2$$
$$+ c_{b_1}(1-f_{t_2})\tilde{S}\tilde{\nu} + \frac{Re}{M_\infty}f_{t_1}\Delta U^2 \tag{12}$$

where

$$f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3} \tag{13}$$

$$\chi = \tilde{\nu}/\nu \tag{14}$$

$$\tilde{S} = S + (M_\infty/Re)(\tilde{\nu}/\kappa^2 d^2)f_{v_2} \tag{15}$$

and

$$f_{v_2} = 1 - \frac{\chi}{1+\chi f_{v_1}} \tag{16}$$

In these equations, d is the distance to the nearest wall. The function f_w is given as

$$f_w = g\left(\frac{1+c_{w_3}^6}{g^6 + c_{w_3}}\right)^{\frac{1}{6}} \tag{17}$$

where

$$g = r + c_{w_2}(r^6 - r) \tag{18}$$

and

$$r = \frac{M_\infty}{Re}\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \tag{19}$$

The last term in Eq. (12) is used when specifying the transition location. Although the flow solver includes this term, the computations in the present paper are all assumed to be fully turbulent, and so this term is not used. Therefore, the definitions of f_{t_1} and f_2, which are associated with these terms, are not given. After Eq. (12) is solved for $\tilde{\nu}$, the eddy viscosity is computed as

$$\mu_t = \rho\nu_t = \rho\tilde{\nu}f_{v_1} \tag{20}$$

Adjoint Equations

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

$$I[\boldsymbol{Q}, \boldsymbol{D}, \Psi, \boldsymbol{X}(\boldsymbol{D})] = I_c(\boldsymbol{Q}, \boldsymbol{D}) + \Psi^T\boldsymbol{R}[\boldsymbol{Q}, \boldsymbol{D}, \boldsymbol{X}(\boldsymbol{D})] \tag{21}$$

where $\boldsymbol{R}$ is the vector of discrete residuals and Ψ are the Lagrange multipliers (also referred to as the costate or adjoint variables). In Eq. (21), $I_c(\boldsymbol{Q}, \boldsymbol{D})$ represents the cost that is to be minimized. Examples of suitable cost functions include the difference between the lift coefficient for the airfoil and a desired lift, the drag coefficient, and the difference between the pressure distribution and a desired pressure distribution.

The variation of Eq. (21) is given by

$$\delta I = \frac{\partial I_c}{\partial \boldsymbol{Q}}\tilde{\boldsymbol{Q}} + \frac{\partial I_c}{\partial \boldsymbol{D}}\tilde{\boldsymbol{D}} + \Psi^T\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\tilde{\boldsymbol{Q}} + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}} + \frac{\partial \boldsymbol{R}\partial \boldsymbol{X}}{\partial \boldsymbol{X}\partial \boldsymbol{D}}\right)\tilde{\boldsymbol{D}}\right] \tag{22}$$

The terms involving $\tilde{\boldsymbol{Q}}$ can be eliminated by regrouping terms and requiring the coefficients of $\tilde{\boldsymbol{Q}}$ to vanish; the costate variables are the solution of the following equations:

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right]^T \{\Psi\} + \left\{\frac{\partial I_c}{\partial \boldsymbol{Q}}\right\} = 0 \tag{23}$$

The remaining terms for the variation in the cost function are then given by

$$\delta I = \left[\frac{\partial I_c}{\partial \boldsymbol{D}} + \Psi^T\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}} + \frac{\partial \boldsymbol{R}\partial \boldsymbol{X}}{\partial \boldsymbol{X}\partial \boldsymbol{D}}\right)\right]\tilde{\boldsymbol{D}} \tag{24}$$

After the costate variables are determined from Eq. (23), they are used in Eq. (24) to obtain the sensitivity derivatives. Note that this process requires the solution of both the flow equations and the costate equations. However, the derivatives of the cost function with respect to all design variables are obtained independently of the number of design variables.

By examining Eqs. (5–9) along with Eqs. (12–20), it is apparent that the solution of the flow equations and the turbulent viscosity are highly dependent on one another. Therefore, the vector of residuals that require linearization in Eqs. (23) and (24) includes the contributions from both the flow equations and the turbulence model. Likewise, the dependent variables $\boldsymbol{Q}$ include the conserved flow variables as well as $\tilde{\nu}$ so that solving for the costate variables with Eq. (23) requires the solution of a block 5×5 system of equations for two-dimensional calculations and a 6×6 system in three dimensions.

Many of the terms in Eqs. (12–20) have a complex dependency on the dependent variables, the design variables, and the distance to the wall; these terms must be accurately differentiated to obtain accurate derivatives. In the present work, the differentiation of both the flow equations and the turbulence model is accomplished by hand differentiating the code. Although this procedure is somewhat tedious, experiments in which the eddy viscosity was assumed to be constant (and, therefore, not differentiated) yielded very poor accuracy with many derivatives of incorrect sign when compared with gradients obtained with finite differences. The strong coupling of the flow equations and the turbulence model is in contrast to Refs. 12 and 18, where the constant viscosity assumption was used. However, in those references, an algebraic turbulence model is used, whereas here, a field equation is solved to obtain the eddy viscosity.

Solution Procedures

For the flow equations, the inviscid flux contributions are evaluated by using an approximate Riemann solver,[28] and the viscous contributions are discretized with a central-difference approach. The solution is obtained by using an implicit solution methodology with multigrid acceleration. Details may be found in Refs. 29–31. The adjoint equations are a linear system of equations that can be solved with a technique such as the preconditioned Generalized Minimum Residual (GMRES) method.[32] However, in this work, a time derivative is added to the equations so that they can be solved with a time-marching procedure. The motivation for adding the time term is that this approach often converges in situations for which the preconditioned GMRES might otherwise fail. This feature is particularly useful when the turbulence model is fully coupled because the turbulence production term tends to reduce diagonal dominance. Because the adjoint equations represent a linear system of equations, the matrix–vector products are currently 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 {incomplete lower/upper (ILU) decomposition with no fill [ILU(0)]}, so that the resulting scheme requires roughly the same amount of memory as the flow solver. Note that this procedure essentially requires recomputation of the linearization of the residual for each matrix–vector product.

Grid Generation and Mesh Movement

The unstructured meshes used in this work are generated with the software package described in Ref. 33. This employs an advancing front method that generates good quality grids for both inviscid and viscous calculations.

During the design process, the mesh is continuously updated as the shape of the geometry changes. This is accomplished using the technique described in Ref. 1, which shifts nodes near viscous surfaces by interpolating the changes in the coordinates at the endpoints of the nearest surface edge. This technique is blended with a smoothing procedure so that away from the highly stretched cells near the surface the mesh movement reverts to that of the smoothing/edge-swapping procedure described in Ref. 34. The combined procedure has been found to work well for viscous grids with highly stretched triangles and very close spacing normal to the wall. Further details can be found in Ref. 1. Also, the effects on the sensitivity derivatives due to the movement of the interior mesh points are taken into account by differentiating the mesh movement process described earlier.

Surface Representation

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 in surface fidelity can be obtained. On one hand, the design variables can be made to correspond to the individual grid points on the surface when a linear polynomial and an appropriate number of control points are chosen. Conversely, a single polynomial curve of degree M (known as a Bezier curve) can be used to describe the geometry when the number of control points is chosen to be $M + 1$. In addition, through the knot sequence associated with the spline, curves with 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.

Rather than a conventional cubic spline of the input coordinates, a B-spline of specified order and with a specified number of control points is matched to the input coordinates with a least-squares procedure. The design variables are then the coordinates of the B-spline control points, which can be considerably fewer in number and are more geometrically meaningful than the original input coordinates. The following is a description of the B-spline representation and the least-squares procedure.

B-spline curves are described in detail in Ref. 35. They are defined as the sum of products of control-point coordinates and corresponding basis functions. The basis functions depend on a knot sequence s_i and are defined recursively as follows:

$$N_i^n(s) = \frac{s - s_{i-1}}{s_{i+n-1} - s_{i-1}} N_i^{n-1}(s) + \frac{s_{i+n} - s}{s_{i+n} - s_i} N_{i+1}^{n-1}(s)$$

$$N_i^0(s) = \begin{cases} 1, & s_{i-1} \le s < s_i \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

where n is the degree of the basis function. The minimum and the maximum values of the parameter s appear n times at the beginning and the end of the knot sequence, respectively, so that the first and last control points correspond to the endpoints of the B-spline.

A uniform parameterization is formed by setting the parameter s that corresponds to each input coordinate equal to the number of the coordinate in the sequence, starting from zero,

$$s_j = j, \qquad j \in [0, M'] \tag{26}$$

where M' is the number of input coordinates. The knot sequence is formed by uniform division of the parameter space. At each of the s_j, each of the M basis functions N_i^n is computed, which forms an $M' \times M$ matrix.

Given the values of the basis functions at each input coordinate, an overdetermined linear system is obtained:

$$\bar{x}'_j = \sum_{i=0}^{M} \bar{x}_i N_i^n(s_j) \tag{27}$$

where $\bar{x}'_j$ is the jth input coordinate and M is the number of control points. The $\bar{x}_i$ are the unknown control point coordinates, and the $\bar{x}'_j$ are the input coordinates. The first and last control points

are set equal to the first and last input coordinates, and the corresponding equations are removed from the system. The system is then solved in the least-squares sense by using Householder transformations, as described in Ref. 36. After fitting each segment of a curve, the B-spline segments are concatenated into a single B-spline by concatenating the knot sequences and merging the control point coordinates.

Optimizer

The optimizer used in the current study is KSOPT,[37] 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 was chosen because it is capable of multipoint design and it can handle both equality and inequality constraints. In addition, upper and lower bounds can be placed on design variables; this approach is currently used to enforce the geometric constraints necessary to maintain a viable geometry throughout the design cycle.

Results

Accuracy of Derivatives

The accuracy of the derivatives is established by comparing results obtained using the adjoint formulation with finite difference derivatives. The case considered here is a two-element airfoil at a freestream Mach number of 0.25, an angle of attack of 1 deg, and a Reynolds number of 9×10^6, based on the chord of the airfoil. The grid used is fairly coarse with only 4901 nodes and a spacing at the wall of about 1×10^{-4} (Fig. 1). The spacing at the wall has been chosen to be large enough so that a stretched mesh can be obtained while allowing the surface of the airfoil to be perturbed without necessarily moving the interior grid points. This is consistent with the assumption that the interior mesh sensitivities are neglected and will allow a direct comparison of various derivatives computed with and without these terms. Although this grid is obviously inadequate for resolving the boundary layer accurately, it is sufficient for verifying the consistency of the derivatives obtained with the adjoint formulation to those computed using finite differences. When the gradients are computed with finite differences, a central-difference formula is used with a fixed step size for each design variable, and all computations are converged to machine accuracy. For grids in which closer spacing at the wall is used, Hou et al.[9] have shown that obtaining derivatives from finite differences can be highly sensitive to the step size and to the level of convergence of the flow solver. With the spacing at the wall used here, the flow solver is easily converged to machine zero, and numerical experiments indicate that the derivative is not drastically effected by the choice of step size. For this grid under these flow conditions, the maximum turbulent viscosity in the flow field is approximately 2600.

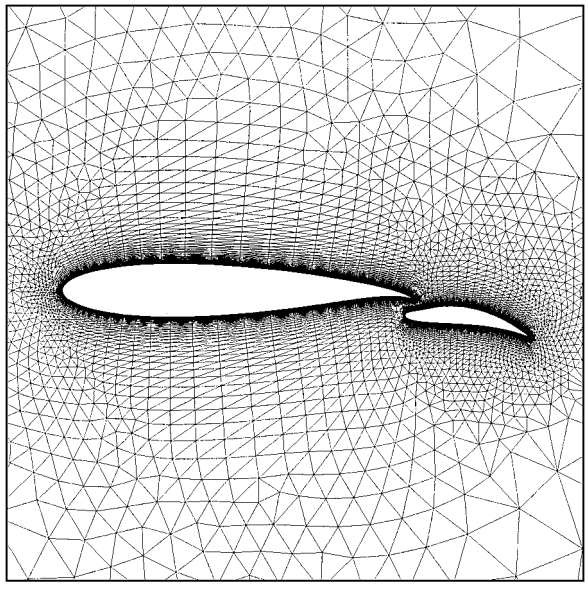

Fig. 1 Grid used for studying accuracy of derivatives.

Table 1 Accuracy of derivatives for lift coefficient with mesh sensitivities included

Location	Finite difference	Adjoint	% Diff.
Nose (main)	−1.1046	−1.1046	0.000
Rear (main)	0.48400	0.48428	0.058
Nose (flap)	−2.2819	−2.2819	0.000
Rear (flap)	0.74900	0.74860	−0.053

Table 2 Accuracy of derivatives for lift coefficient with mesh sensitivities neglected

Location	Finite difference	Adjoint	% Diff.
Nose (main)	1.5960	1.5962	0.013
Rear (main)	9.6079	9.5976	−0.107
Nose (flap)	−0.00105	−0.00109	3.809
Rear (flap)	3.3796	3.3794	−0.006

Table 3 Accuracy of derivatives for drag coefficient with mesh sensitivities included

Location	Finite difference	Adjoint	% Diff.
Nose (main)	0.22753	0.22752	−0.004
Rear (main)	−0.02530	−0.02531	0.040
Nose (flap)	0.17795	0.17795	0.000
Rear (flap)	0.06521	0.06529	0.123

Table 4 Accuracy of derivatives for drag coefficient with mesh sensitivities neglected

Location	Finite difference	Adjoint	% Diff.
Nose (main)	−0.69105	−0.69105	0.000
Rear (main)	−0.95156	−0.95060	−0.101
Nose (flap)	−0.072715	−0.072679	−0.050
Rear (flap)	−0.016764	−0.016788	0.143

Table 5 Accuracy of derivatives of lift coefficient for flap translation with mesh sensitivities included

Flap	Finite difference	Adjoint	% Diff.
x translation	0.35470	0.35481	0.031
y translation	−7.1370	−7.1370	0.000

For this test, the geometry of each airfoil is described with a third-order B-spline. The derivatives of lift and drag coefficients with respect to the vertical position of four shape design variables are evaluated. Two of these design variables are located on the main airfoil, and two are located on the flap. For each element, one design variable is located on the upper surface near the nose of the airfoil, and one is located near the rear. A comparison of derivatives of lift and drag coefficients with respect to these design variables is shown in Tables 1–4. In Table 1, the sensitivity of the lift coefficient is shown for the case in which the interior mesh sensitivities are included, whereas Table 2 shows the corresponding derivatives in which the mesh sensitivities are neglected. A similar comparison for the drag coefficient is shown in Tables 3 and 4. As seen, the derivatives obtained with the adjoint approach are in very good agreement with the finite difference derivatives for all cases. Furthermore, it is seen that the derivatives computed by neglecting the interior mesh sensitivities differ significantly from those computed by including these terms. In particular, the sign of the derivatives near the nose of the main element are not of the same sign as when the mesh sensitivities are included. Because the interior of the mesh is allowed to change during an actual design, failure to include these terms could lead to a failure of the design.

In Tables 5 and 6, sensitivity derivatives of the lift coefficient with respect to x and y translations of the flap are given. In Table 5, the derivative of the lift computed with the adjoint approach is compared with finite differences for the case when the mesh sensitivities

Table 6 Accuracy of derivatives of lift coefficient for flap translation with mesh sensitivities neglected

Flap	Finite difference	Adjoint	% Diff.
x translation	8.4127	8.4547	0.500
y translation	44.116	44.136	0.045

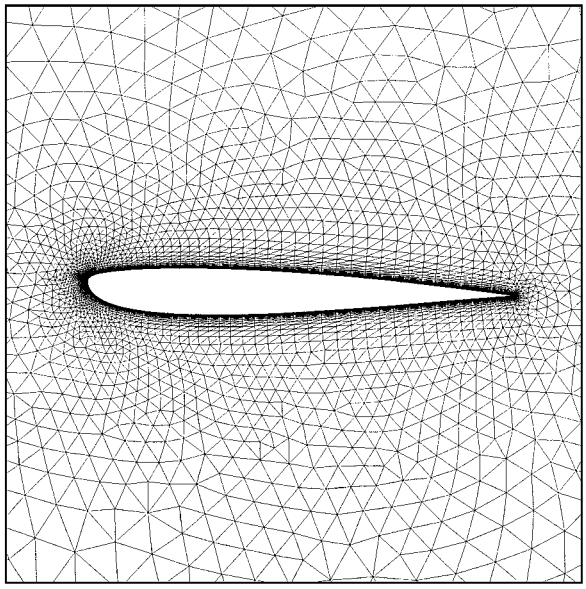

Fig. 2 Grid used for first test case.

are included. The corresponding derivatives obtained by neglecting the interior mesh sensitivities are shown in Table 6. It is apparent in both cases that the derivatives computed with the adjoint approach are in excellent agreement with those computed with finite differences. Furthermore, if the influence of the mesh sensitivities are neglected, the sign of the derivative when the flap is translated in the y direction differs from the case where these terms are included. It has been shown in Ref. 1 that for geometries with singularities, such as sharp trailing edges, the effects of the mesh sensitivities on the derivatives associated with translation are dependent on the manner in which the mesh is moved in response to the changing position of the airfoil. It is also demonstrated that this effect does not vanish as the mesh is refined so that for computing derivatives due to translation of elements in a multielement configuration, these terms must be included.

Design Examples

Two sample results are given. The first case is a computation of the flow over an airfoil at a freestream Mach number of 0.4, an angle of attack of 2 deg, and a Reynolds number of 5×10^6. The goal of the computation is simply to obtain a specified pressure distribution. The grid used for this computation is shown in Fig. 2 and consists of approximately 5500 nodes with 128 nodes on the surface of the airfoil. The spacing at the wall is 1×10^{-5} of the chord length yielding a y^+ of about 2. For this case, a single eighth-order Bezier curve is used to parameterize the surface, and only three design variables are allowed to change during the design process. The geometry is perturbed by displacing three of the control points in the initial B-spline definition, and the solution over this geometry is used for the target pressures. After 10 design cycles, the cost function is reduced from approximately 1.5×10^{-1} to 3.0×10^{-7}, and the root mean square of the gradients is reduced from 1.4 to 5.8×10^{-4}. The initial and final pressure distributions and geometries are shown in Figs. 3 and 4. As seen, the target pressure distribution is obtained, and the geometry returns to that of the airfoil in the perturbed position.

The objective of the next case is to reduce the drag coefficient while maintaining a specified lift. The geometry is a two-element airfoil designed to operate at transonic speeds and is the precursor to modern supercritical airfoils.[38] The geometry and mesh for the

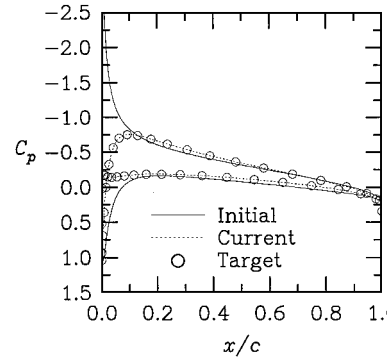

Fig. 3 Initial and final pressure distributions for case 1.

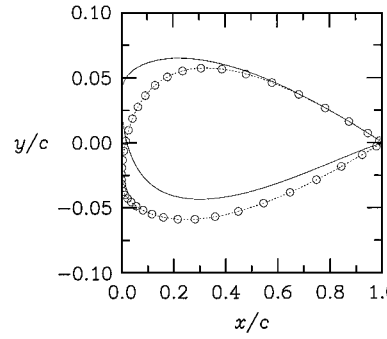

Fig. 4 Initial and final geometries for case 1.

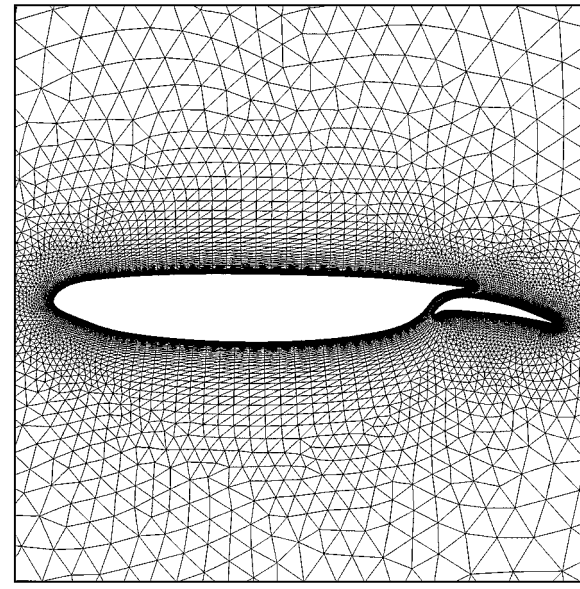

Fig. 5 Initial grid for slotted airfoil.

initial configuration are shown in Fig. 5. For this mesh, there are 18,905 nodes with 193 lying on the surface of the main element and 161 on the flap. The spacing normal to the wall is 1.0×10^{-5}. The freestream Mach number is 0.75, the initial angle of attack is 0 deg, and the Reynolds number is 5×10^6. Under these conditions, the initial lift and drag coefficients are 1.039 and 0.0376, respectively. For this case, the objective function is defined as a linear combination of the lift and drag

$$I_c = 25c_d^2 + (c_l - 1.039)^2 \tag{28}$$

Here, the drag is weighted more heavily than the lift to provide more balance between the terms so that both contributions are approximately equal. A better way to minimize the drag while maintaining a specified lift may be to treat the lift as a constraint during the design process. This requires an additional solution of the adjoint equations but would eliminate the need for adjusting the weights to obtain the desired lift. For this case, there are 42 geometric design variables in addition to the angle of attack.

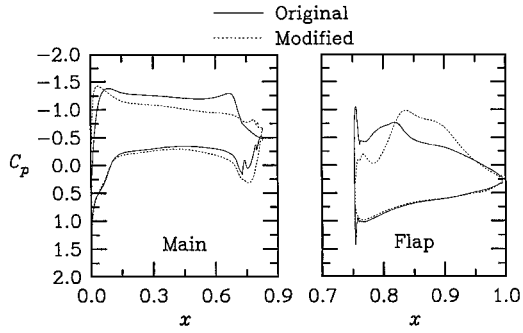

Fig. 6 Pressure distributions for slotted airfoil.

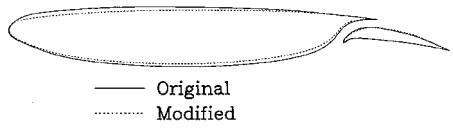

Fig. 7 Initial and modified geometries for slotted airfoil.

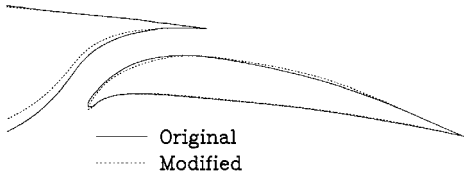

Fig. 8 Near view of flap for slotted airfoil.

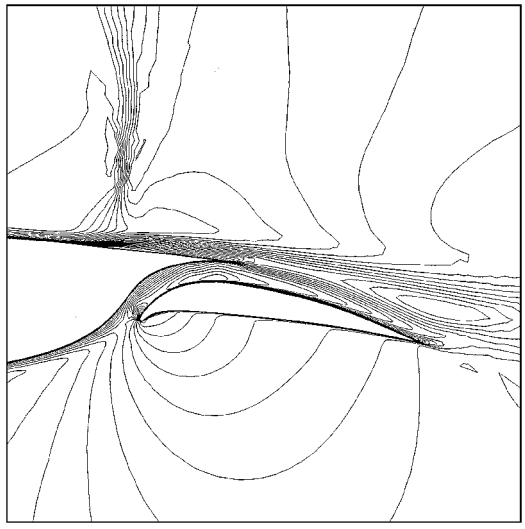

Fig. 9 Mach number contours for initial slotted airfoil.

After 24 design cycles, the drag has been reduced from 0.0376 to 0.0209. Continuing to run the design further reduces the drag slightly to 0.0201 after 96 cycles. The slower convergence of the design process with the increase in the number of design variables is attributable to the poor performance of quasi-Newton methods for aerodynamic design problems with many design variables directly describing the geometry.[39] The initial and final pressure distributions are shown in Fig. 6. As seen, the initial pressure distribution exhibits a shock toward the rear of the main element, whereas in the final pressure distribution, this shock has been eliminated and the contribution to the lift due to the flap has noticeably increased. A comparison of the initial and final geometries is shown in Figs. 7 and 8. Figure 7 shows a noticeable change in the geometry of the main element on both the upper and lower surfaces. A near-field view of the flap, shown in Fig. 8, shows that the flap has also been modified, although to a lesser extent. Mach number contours for the initial geometry and the modified geometry are shown in Figs. 9 and 10, respectively. As seen in Fig. 9, the solution for the initial geometry

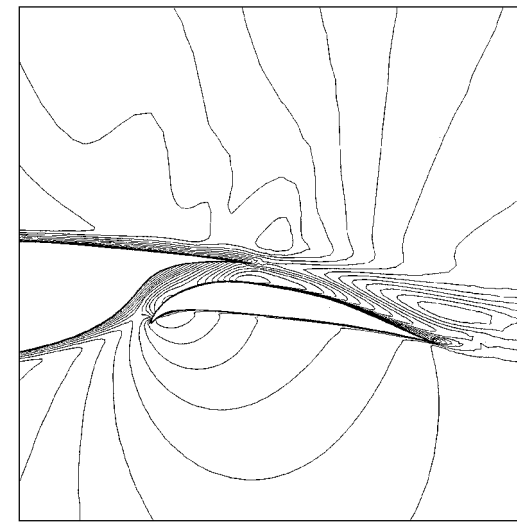

Fig. 10 Mach number contours for modified slotted airfoil.

exhibits noticeable thickening of the boundary layer aft of the shock on the upper surface and is accompanied by a small region of separation. Mach contours for the modified geometry further demonstrate that the shock has been eliminated.

Summary

Several recommendations are offered. First, the slow convergence of the second test case, in which 43 design variables were used, shows that the quasi-Newton method is insufficient for problems with many design variables because a large number of design iterations is required before a good approximation of the Hessian can be obtained. Even then, with many design variables, this Hessian may remain inaccurate because much of the information is obtained much earlier in the design process and may not represent the Hessian in the vicinity of the minimum. However, direct computation of the Hessian for turbulent Navier–Stokes design cases is not currently very efficient or practical because it requires the solution of a linear system of equations for each design variable as well as one for the adjoint (see, e.g., Refs. 40 and 41). Methods that approximate the Hessian, such as described in Ref. 39, should be thoroughly evaluated and extended to viscous flows. Other methods, such as pseudotime techniques,[42] have been demonstrated for inviscid flow computations[10] and should be examined for applicability to viscous computations as well. In addition, the technique employed in Refs. 12 and 13 should also be further evaluated. This technique is essentially a time-marching technique in which the gradients are smoothed at each step. For two-dimensional flows, this technique is similar in application to that of the preconditioning method described in Ref. 39. However, for three dimensions, the technique in Ref. 39 requires the solution of an extra field equation. Finally, Ta'asan has shown in Ref. 43 that designing for the slopes of the geometry instead of the location of the surface presents a design that is easier and faster to converge. The use of slopes and curvatures instead of points as design variables should, therefore, be considered.

Concluding Remarks

A two-dimensional design optimization methodology is described. This research is an extension of earlier work to include a turbulence model for viscous flows. However, a discrete adjoint approach is used instead of the continuous adjoint approach so that the sensitivity derivatives are more consistent with the flow solver. The turbulence model is strongly coupled with the flow equations, and the accuracy of the derivatives is demonstrated through a comparison with derivatives obtained by finite differences. A few examples are presented to demonstrate the methodology.

References

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

[2]Angrand, F., "Optimum Design for Potential Flows," *International Journal for Numerical Methods in Fluids*, Vol. 3, 1983, pp. 265–282.

[3]Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering*, Vol. 113, Dec. 1991, pp. 681–688.

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

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

[6]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.

[7]Elliott, J., and Peraire, J., "Aerodynamic Design Using Unstructured Meshes," AIAA Paper 96-1941, June 1996.

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

[9]Hou, G. J.-W., Maroju, V., Taylor, A. C., and Korivi, V. M., "Transonic Turbulent Airfoil Design Optimization with Automatic Differentiation in Incremental Iterative Forms," AIAA Paper 95-1692, June 1995.

[10]Iollo, A., and Salas, M. D., "Optimum Transonic Airfoils Based on the Euler Equations," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 96-76, Hampton, VA, Dec. 1996.

[11]Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, Vol. 3, Nov. 1988, pp. 23–260.

[12]Jameson, A., "Optimum Aerodynamic Design Using the Control Theory," *Computational Fluid Dynamics Review*, 1995, pp. 495–528.

[13]Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamic Design Using the Navier–Stokes Equations," AIAA Paper 97-0101, Jan. 1997.

[14]Marco, N., and Beux, F., "Multilevel Optimization: Application to One-Shot Shape Optimum Design," Institut de Recherche en Informatique et en Automatique, INRIA Rept. 2068, Sophia-Antipolis, France, Oct. 1993.

[15]Marco, N., "Optimisation de Formes Aerodynamiques 2D et 3D Par Une Methode Multi-Niveau En Maillages Non Structures," Ph.D. Dissertation, Faculté des Sciences, l'Universite De Nice-Sophia Antipolis, France, Nov. 1995.

[16]Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, Pt. 1, 1974, pp. 97–110.

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

[18]Soemarwoto, B., "Multi-Point Aerodynamic Design by Optimization," Ph.D. Dissertation, Dept. of Theoretical Aerodynamics, Delft Univ. of Technology, Delft, The Netherlands, Dec. 1996.

[19]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 Lab., Argonne, IL, Sept. 1991.

[20]Baldwin, B., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.

[21]Elliot, J., and Peraire, J., "Practical 3D Aerodynamic Design and Optimization Using Unstructured Grids," AIAA Paper 96-4170, Sept. 1996.

[22]Newman, J. C., and Taylor, A. C., "Three-Dimensional Aerodynamic Shape Sensitivity Analysis and Design Optimization Using the Euler Equations on Unstructured Grids," AIAA Paper 96-2464, June 1996.

[23]Newman, J. C., Taylor, A. C., and Barnwell, R. W., "Aerodynamic Shape Sensitivity Analysis and Design Optimization of Complex Configurations Using Unstructured Grids," AIAA Paper 97-2275, June 1997.

[24]Elliot, J., and Peraire, J., "Aerodynamic Optimization on Unstructured Meshes with Viscous Effects," AIAA Paper 97-1849, June 1997.

[25]Mohammadi, B., "Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence," AIAA Paper 97-0099, Jan. 1997.

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

[27]Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1991.

[28]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[29]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.

[30]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/ Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, 1996, pp. 391–408.

[31]Bonhaus, D. L., "An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes," M.S. Thesis, Dept. of Aerospace Engineering, George Washington Univ., Washington, DC, Aug. 1993.

[32]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, July 1986, pp. 856–869.

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

[34]Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 127, 1996, pp. 380–397.

[35]Farin, G., "Curves and Surfaces for CAGD," 3rd ed., Academic, San Diego, CA, 1993, pp. 157–186.

[36]Golub, G., and van Loan, C., *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1991, pp. 194–197.

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

[38]Whitcomb, R. T., and Clark, L. R., "An Airfoil Shape for Efficient Flight at Supercritical Mach Numbers," NASA TM-X-1109, April 1965.

[39]Arian, E., and Ta'asan, S., "Analysis of the Hessian for Aerodynamic Optimization: Inviscid Flow," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 96-28, Hampton, VA, 1996.

[40]Sherman, L. L., Taylor, A. C., Green, L. L., Newman, P. A., Hou, G. J.-W., and Korivi, V. M., "First- and Second-Order Aerodynamic Sensitivity Derivatives via Automatic Differentiation with Incremental Iterative Methods," AIAA Paper 94-4262, Sept. 1994.

[41]Young, D. P., Huffman, W. P., Melvin, R. G., Bieterman, M. B., Hilmes, C. L., and Johnson, F. T., "Inexactness and Global Convergence in Design Optimization," AIAA Paper 94-4386, Sept. 1994.

[42]Ta'asan, S., "Pseudo-Time Methods for Constrained Optimization Problems Governed by P.D.E," NASA CR-195081, May 1995.

[43]Ta'asan, S., "Trends in Aerodynamic Design and Optimization: A Mathematical View Point," AIAA Paper 95-1731, June 1995.

J. Kallinderis
Associate Editor