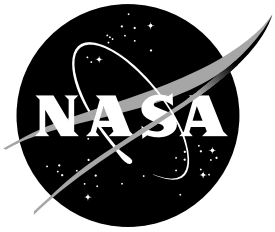
NASA Technical Memorandum 112867



Aerodynamic Design on Unstructured Grids for Turbulent Flows

W. Kyle Anderson and Daryl L. Bonhaus
Langley Research Center, Hampton, Virginia

June 1997

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Aerodynamic Design on Unstructured Grids for Turbulent Flows

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

Abstract

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. In addition, a user interface is described that significantly reduces the time required to set up the design problems. Recommendations on directions of further research into the Navier-Stokes design process are made.

Introduction

Because of rapid advances in computer speeds, and improvements in flow-solver and grid-generation algorithms, a renewed emphasis has been placed on extending computational fluid dynamics (CFD) 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, 4, 7, 8, 11-13, 17, 19-24, 29, 30 and 34.

Although most of the above mentioned references deal with inviscid flows, a few have addressed viscous computations of turbulent flows. In Ref. 18, Hou et al. used a direct differentiation approach in which the derivatives of the flow solver were obtained with ADIFOR.[9] In Ref. 18, the turbulence model used was the Baldwin-Lomax[6] algebraic model, which was differentiated along with the flow equations. Jameson recently developed a design methodology for turbulent flows based on an adjoint formulation.[21] 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.[34]

For unstructured grids, the work to date has been primarily focused on inviscid computations in both two and three dimensions.[1,8,13,14,27,28] In Ref. 1, the adjoint equations and boundary conditions were derived for the incompressible Navier-Stokes equations, and some design examples were demonstrated. However, turbulence effects were not included. In the work of Mohammadi,[26] two-dimensional Navier-Stokes results were presented in which turbulence effects were included. In this reference, automatic differentiation was used to differentiate the necessary components of the flow solver.

The purpose of the present study is to extend the work in Ref. 1 to the compressible Navier-Stokes equations, including a fully coupled field-equation turbulence model. However, because a continuous adjoint approach for unstructured grids requires accurately computed second derivatives,[1] a discrete adjoint approach is used in the present study. The methodology is discussed, and the accuracy of the derivatives is established. A few design examples are given to demonstrate the technology. Also, a user interface has been developed to facilitate setup of the designs, and a description of the interface is included.

Nomenclature

A	area of control volume
a	speed of sound
C^*	constant used in Sutherland's law for viscosity
c_l	lift coefficient
c_l	drag coefficient
c_{b_1}, c_{b_2}, c_{v_1}	constants used in Spalart-Allmaras turbulence model
c_{w_1}, c_{w_2}, c_{w_3}	constants used in Spalart-Allmaras turbulence mode
$\mathbf{D}$	vector of design variables
D_i	component i of design variable vector
d	distance to nearest surface
E	total energy per unit volume
$\hat{\mathbf{F}}$	fluxes of mass, momentum, and energy
$\hat{\mathbf{F}}_i$	inviscid contribution to fluxes
$\hat{\mathbf{F}}_v$	viscous contribution to fluxes
$\mathbf{f}$, $\mathbf{g}$	components of inviscid fluxes
$\mathbf{f}_v$, $\mathbf{g}_v$	components of viscous fluxes
f_{v_1}, f_{v_2}	functions used in the turbulence model
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
κ	Karman constant
M_∞	free-stream Mach number
N	B-spline basis functions
$\hat{\mathbf{n}}$	unit normal to boundary of control volume
Pr	Prandtl number
Pr_t	turbulent Prandtl number

p	pressure
$\mathbf{Q}$	vector of dependent variables
q_x, q_y	components of heat flux
$\mathbf{R}$	residual for a control volume
Re	Reynolds number
S	magnitude of vorticity
s	parameterization variable for B-splines
T	temperature
t	time
$\mathbf{U}$	magnitude of velocity
u, v	Cartesian components of velocity
$\mathbf{X}$	grid-point locations
x, y	Cartesian coordinates
y^+	wall coordinate
α	angle of attack
$\partial\Omega$	boundary of control volume
γ	ratio of specific heats
μ	laminar viscosity
μ_t	turbulent viscosity
υ	μ/ρ
υ_t	μ_t/ρ
$\tilde{\upsilon}$	dependent variable for turbulence model
ρ	density
σ	constant for turbulence model
τ_{xx}, τ_{xy}, τ_{yy}	shear stress terms
Ψ	costate variables

Superscripts:

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

Subscripts:

∞	free-stream quantities

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 free-stream density, speed of sound, temperature, viscosity, thermal conductivity, and a reference length) are given as

$$A\frac{\partial \mathbf{Q}}{\partial t} + \oint_{\partial\Omega} \vec{\mathbf{F}}_i \cdot \hat{\mathbf{n}}\,dl - \oint_{\partial\Omega} \vec{\mathbf{F}}_v \cdot \hat{\mathbf{n}}\,dl = 0 \tag{1}$$

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

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

$$\vec{\mathbf{F}}_i = \mathbf{f}\hat{\imath} + \mathbf{g}\hat{\jmath} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix}\hat{\imath} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}\hat{\jmath} \tag{3}$$

and

$$\vec{\mathbf{F}}_v = \mathbf{f}_v\hat{\imath} + \mathbf{g}_v\hat{\jmath} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}\hat{\imath} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{bmatrix}\hat{\jmath} \tag{4}$$

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

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

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

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

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

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

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:

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

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

Turbulence model

For the current study, the turbulence model of Spalart-Allmaras is used.[35] This is a one-equation turbulence model given as

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

where

$$f_{v_1} = \frac{\chi^3}{\chi^3+c_{v_1}} \qquad (13)$$

$$\chi = \frac{\tilde{\upsilon}}{\upsilon} \qquad (14)$$

$$\tilde{S} = S+\frac{M_\infty}{Re}\frac{\tilde{\upsilon}}{\kappa^2 d^2}f_{v_2} \qquad (15)$$

and

$$f_{v_2} = 1-\frac{\chi}{1+\chi f_{v_1}} \qquad (16)$$

In these equations, S is the magnitude of the vorticity, and d is the distance to the nearest wall. The function f_w is given as

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

where

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

and

$$r = \frac{M_\infty}{Re}\frac{\tilde{\upsilon}}{\tilde{S}\kappa^2 d^2} \qquad (19)$$

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, 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 $\tilde{\upsilon}$, the eddy viscosity is computed as

$$\mu_t = \rho\upsilon_t = \rho\tilde{\upsilon}f_{v_1} \qquad (20)$$

Adjoint Equations

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

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

where $\mathbf{R}$ represents the vector of discrete residuals, $\mathbf{X}$ is the location of the grid points, $\mathbf{D}$ is the vector of design variables, and Ψ are the Lagrange multipliers (also referred to as the costate or adjoint variables). In Eq. (21), $I_c(\mathbf{Q},\mathbf{D})$ 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

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

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

$$\left[\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\right]^T\{\Psi\}+\left\{\frac{\partial I_c}{\partial\mathbf{Q}}\right\} = 0 \qquad (23)$$

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

$$\delta I = \left(\frac{\partial I_c}{\partial\mathbf{D}}+\Psi^T\left(\frac{\partial\mathbf{R}}{\partial\mathbf{D}}+\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\frac{\partial\mathbf{X}}{\partial\mathbf{D}}\right)\right)\tilde{\mathbf{D}} \qquad (24)$$

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, $\mathbf{Q}$, include the conserved flow variables as well as $\tilde{\upsilon}$ so that solving for the costate variables with Eq. (23) requires the solution of a block 5x5 system of equations for two-dimensional calculations and a 6x6 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 in order 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. 21 and 34 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.

In Eq. (24), the terms that involve $\Psi^T[(\partial\mathbf{R}/\partial\mathbf{X})\cdot(\partial\mathbf{X}/\partial\mathbf{D})]\tilde{\mathrm{D}}$ represent the change in the cost function that results from a change in the grid. Reference 1 shows that the contributions of these terms diminish as the grid is refined except at geometric singularities such as trailing edges. Because the position of the trailing edge is fixed in the present work, these terms are currently not included in the computations so that Eq. (24) can be evaluated by looping over only a small subset of edges in the mesh rather than the entire mesh. However, in order to effectively deal with cases that require derivatives at the trailing edge and to better ensure convergence of the optimiza-

tion procedure, even on coarse grids, it is recommended that these terms be included.

Solution Procedures

For the flow equations, the inviscid flux contributions are evaluated by using an approximate Riemann solver,[31] 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. 2, 3, and 10. The adjoint equations are a linear system of equations that can be solved with a technique such as preconditioned GMRES.[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 (LU) 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. 25. 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 end points 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. 38. 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.

Accuracy of Derivatives

The accuracy of the derivatives is established by comparing results obtained by using the adjoint formulation with finite-difference derivatives. The case considered here is a symmetric airfoil at a free-stream Mach number of 0.55, an angle of attack of 1°, and a Reynolds number of 9 million, based on the chord of the airfoil. The grid used is fairly coarse with only 2700 nodes and a spacing at the wall of about 5×10^{-4}. (See 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 to be perturbed without moving the interior grid points. This is done in order to remain consistent with the assumption that the interior mesh sensitivities are neglected. Although this grid is obviously inadequate for resolving the boundary layer accurately, it is sufficient for verifying the linearization of the flow solver. 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.[18] 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 2500.

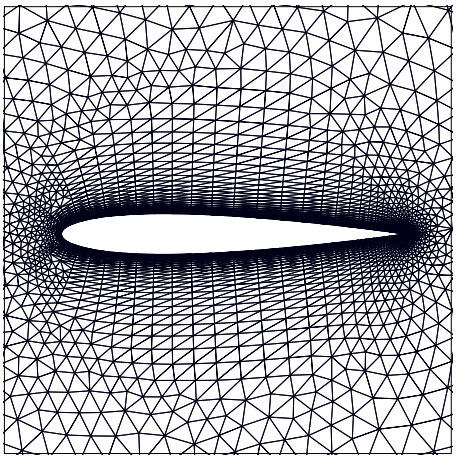


Figure 1. Grid used for studying accuracy of derivatives.

For this test, the geometry of the airfoil is described with a third order B-spline with 39 control points. The derivatives of lift and drag coefficients with respect to angle of attack and to three shape design variables are evaluated. The shape design variables correspond to the y-position of three control points located at $x/c = 0.103$, $x/c = 0.789$, and $x/c = 0.972$ and designated as D_1, D_2, and D_3, respectively. As seen in the tables below, the derivatives obtained with the adjoint approach are in good agreement with the finite-difference derivatives.

Table 1. Accuracy of Derivatives for Lift Coefficient

	Finite-difference	Adjoint
$\frac{\partial c_l}{\partial \alpha}$	5.8278	5.8278
$\frac{\partial c_l}{\partial D_1}$	−1.9065	−1.9067
$\frac{\partial c_l}{\partial D_2}$	1.3505	1.3505
$\frac{\partial c_l}{\partial D_3}$	0.44746	0.44746

Table 2. Accuracy of Derivatives for Drag Coefficient

	Finite-difference	Adjoint
$\frac{\partial c_d}{\partial \alpha}$	0.057503	0.057503
$\frac{\partial c_d}{\partial D_1}$	–0.50394	–0.50401
$\frac{\partial c_d}{\partial D_2}$	–0.063550	–0.063547
$\frac{\partial c_d}{\partial D_3}$	–0.0084115	–0.0084114

Surface Representation and Graphical Interface

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 by choosing a linear polynomial and an appropriate number of control points. Conversely, a single polynomial curve of degree M (known as a Bezier curve) can be used to describe the geometry by choosing the number of control points 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.

Spline fitting of input coordinates

Rather than using 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. 15. They are defined as the sum of products of control-point coordinates and corresponding basis functions. The basis functions depend on the parametrization of the spline and a knot sequence and are defined recursively as follows:

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

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 end points 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:

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

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 *j*-th 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. 16. 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.

Graphical interface

A user interface has been written to facilitate the setup of each design. The intent is to not only speed up the process of setting up each case but to help eliminate errors. Because a B-spline representation is used to describe the geometries, the points on the surface of the airfoil must lie on this surface. Therefore, a least-squares fit of a B-spline curve to the point description of the airfoil is used. This process is described in a previous section and is demonstrated in Fig. 2 for a two-element airfoil, where the positions of the control points are indicated by the symbols. The control panel on the right shows that for the second element (the flap) a third-order B-spline with 25 control points is used to define the surface. The positions of these control points are subsequently used as the design variables. Note that although a cubic representation is used in the present example the order of the spline can range from linear to $M - 1$, where M is the number of points that define the airfoil. In the case for which M points are used to define a curve of degree $M - 1$, the resulting curve corresponds to a Bezier representation. In the present example, the B-spline is evaluated at 129 points, which are then used as input to the grid generation to define the surface. Although not shown, a similar procedure has also been used for the main element. In Fig. 2, a “tear off” menu is also shown, which allows the user to choose whether various symbols are displayed.

In the next figure (Fig. 3), upper and lower limits have been placed on the y-coordinate of many of the control points, with the limits depicted by the extent of the lines above and below the current placement of the design variable. The limits are used as side constraints during the optimization process to prevent the occurrence of nonphysical geometries during the design process. The x-coordinates can also be chosen as design variables by placing distinct upper and lower bounds on their positions; if the upper and lower bounds are the same, then the design variable is not allowed to change and is not considered in the optimization process.

Other capabilities include the ability to zoom and to translate the geometry to manipulate it into position. This capability is particularly useful when placing side constraints near the trailing edges of the airfoils. Also, the shape of the airfoil surfaces can be changed simply by picking and moving the control point, and the initial position can be recovered.

After splining the surface and setting the limits of the design variables, output is written in a format that is suitable for the grid generation program. A file that contains the geometric information, such as the positions and limits of the control points, is also written. This file is continually updated during the design process to reflect the changing shape. This file can be read in at a later date to reset the limits on the design variables, to add more points to the surface definition, or to reshape the geometry.

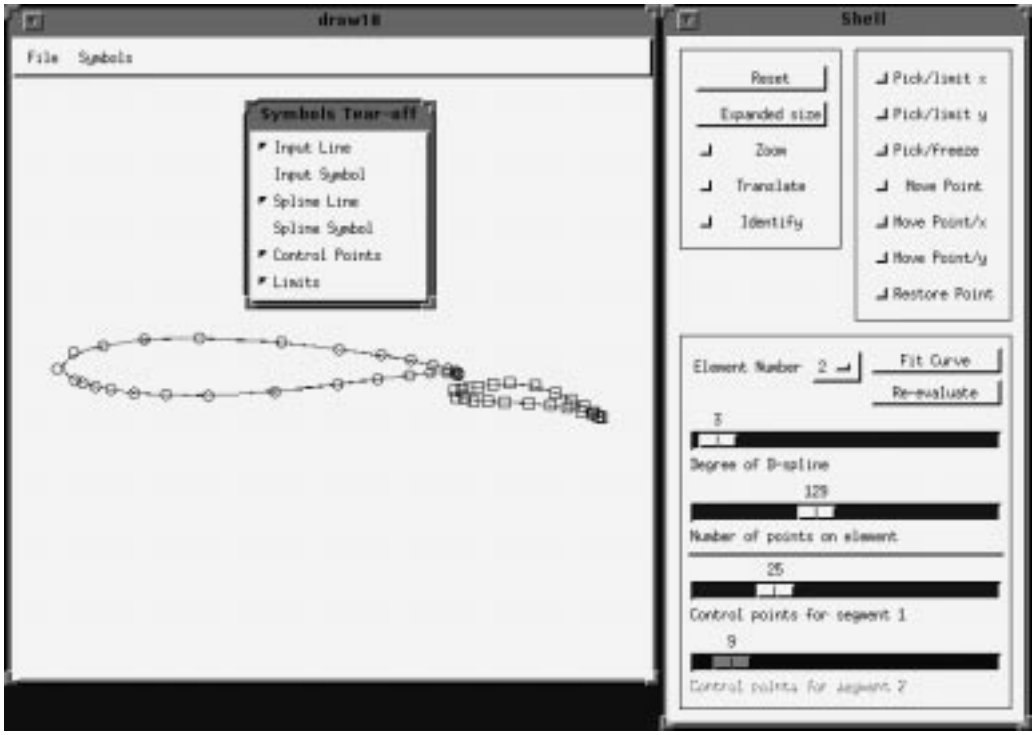


Figure 2. User interface for fitting B-splines to geometry.

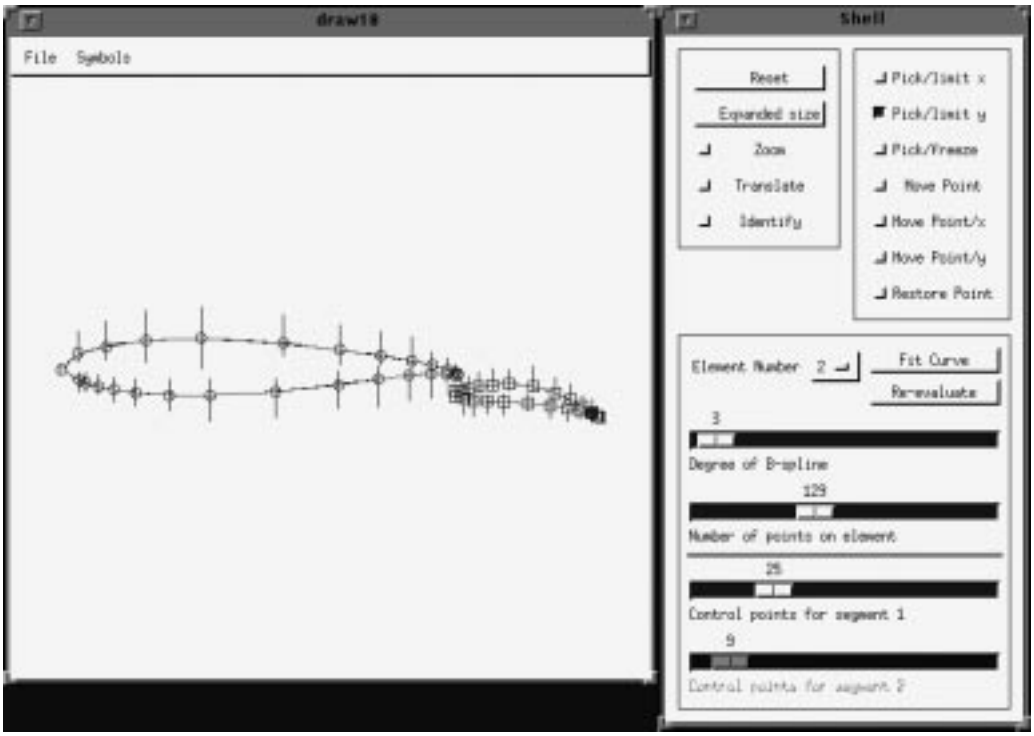


Figure 3. Setting the upper and lower limits on design variables.

Optimizer

The optimizer used in the current study is KSOPT,[39] 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 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

Two sample results are given below. The first case is a computation of the flow over an airfoil at a free-stream Mach number of 0.4, an angle of attack of 2°, and a Reynolds number of 5 million. The goal of the computation is simply to obtain a specified pressure distribution. The grid used for this computation is shown in Fig. 4 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. 5 and 6. As seen, the target pressure distribution is obtained, and the geometry returns to that of the airfoil in the perturbed position.

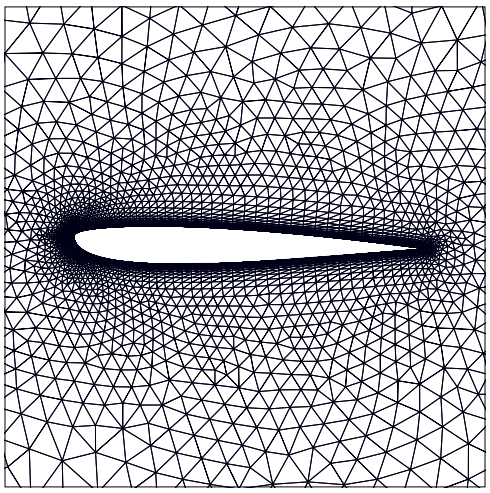


Figure 4. Grid used for first test case.

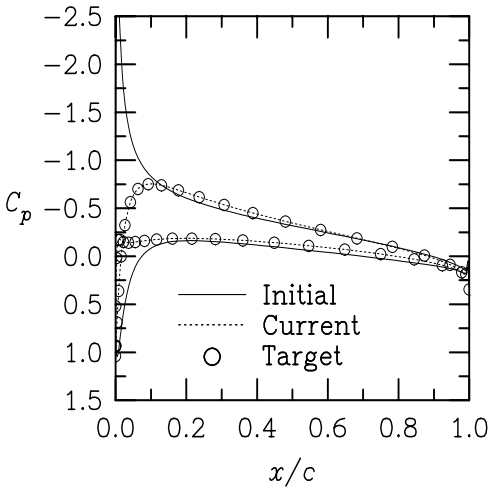


Figure 5. Initial and final pressure distributions for case 1.

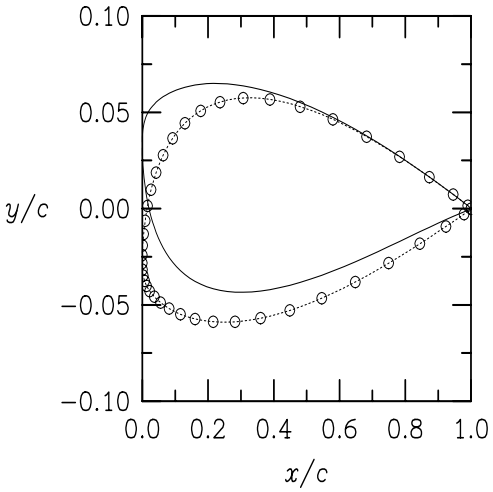


Figure 6. Initial and final geometries for case 1.

The next case is also an inverse design case in which the objective is to match a specified pressure distribution. However, this case is significantly more difficult because 75 design variables are used. The initial grid used for this case consists of about 5300 nodes, and has a spacing normal to the wall of 1.0×10^{-5}. (See Fig. 7.) The free-stream Mach number is 0.725, the angle of attack is 2.54°, and the Reynolds number is 6.5 million. For this case, 20 design cycles were run at one time and restarted 2 times for a total of 60 design 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.[5] As seen in Fig. 8, the desired pressure distribution is obtained reasonably well. However, slight waviness is noted in the final pressure distribution. The need for curvature con-

straints on the geometry is apparent. The final grid is shown in Fig. 9.

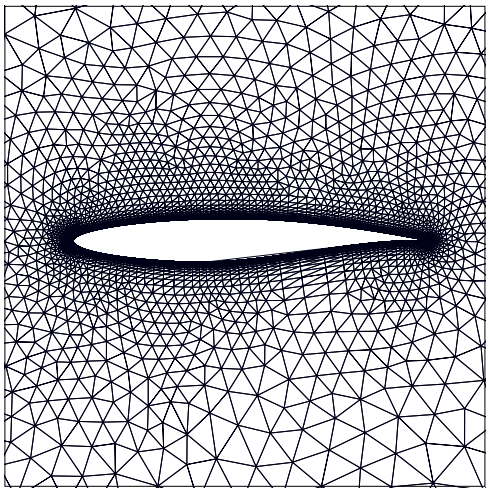


Figure 7. Initial grid for case 2.

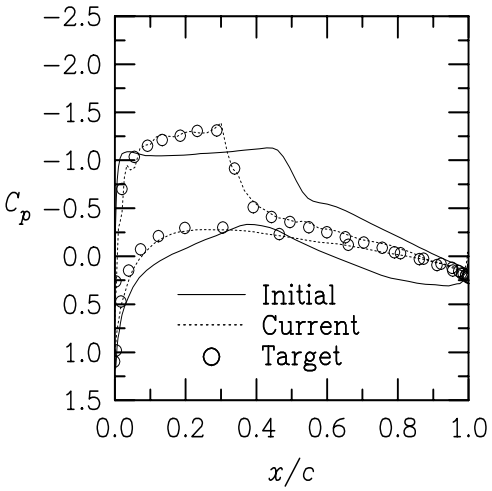


Figure 8. Initial and final pressure distributions for case 2.

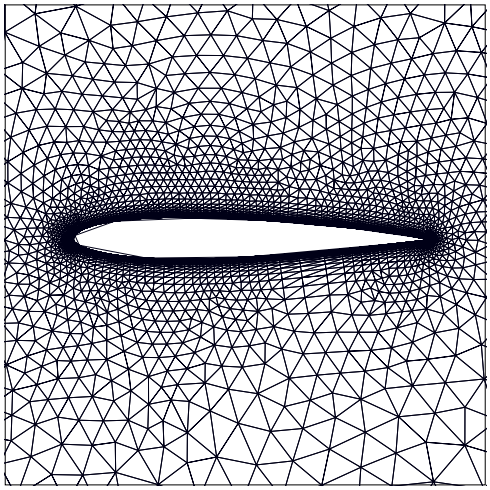


Figure 9. Final grid for case 2.

Summary and Concluding Remarks

A two-dimensional design optimization methodology is described. This research is an extension of the work in Ref. 1 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.

In this regard, several recommendations are offered. First, the slow convergence of the second test case, in which 75 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. 33, and 40.) Methods that approximate the Hessian, such as described in Ref. 5, should be thoroughly evaluated and extended to viscous flows. Other methods, such as pseudo-time techniques,[36] have been demonstrated for inviscid flow computations[19] and should be examined for applicability to viscous computations as well. In addition, the technique employed in Refs. 21, and 22 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. 5. However, for three dimensions, the technique in Ref. 5 requires the solution of an extra field equation. Finally, Ta'asan has shown in Ref. 37 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.

References

[1]Anderson, W. Kyle; and Venkatakrishnan, V.: Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation, AIAA Paper No. 97-0643, 1997.

[2]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.

[3]Anderson, W. K.; Rausch, R. D.; and Bonhaus, D. L.: Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids. *J. Comp. Phys*. vol. 128, 1996, pp. 391-408.

[4]Angrand, F.: Optimum Design for Potential Flows. *Int. J. Num. Methods in Fluids*, vol. 3, 1983, pp. 265-282.

[5]Arian, Eyal; and Ta'asan, Shlomo,: Analysis of the Hessian for Aerodynamic Optimization: Inviscid Flow. ICASE Report No. 96-28, 1996

[6]Baldwin, B.; and Lomax, H.: Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. AIAA Paper No. 78-257, 1978.

[7]Baysal, O.; and Eleshaky, M. E.: Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations. *J. of Fluids Engineering*, vol. 113, 1991, pp. 681-688.

[8]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.

[9]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.

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

[11]Burgreen, G. W.; and Baysal, O.: Three-Dimensional Aerodynamic Shape Optimization of Wings Using Discrete Sensitivity Analysis. *AIAA J*., vol. 34, no. 9, 1996, pp. 1761-1770.

[12]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.

[13]Elliott, J.; and Peraire, J.: Aerodynamic Design Using Unstructured Meshes. AIAA Paper No. 96-1941, 1996.

[14]Elliot, J.; and Peraire, J.: Practical 3D Aerodynamic Design and Optimization Using Unstructured Grids. AIAA Paper No. 96-4170, 1996.

[15]Farin, G.: *Curves and Surfaces for CAGD*. third edition, Academic Press, Inc., San Diego, CA, 1993.

[16]Golub, G; and van Loan, C.: *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1991.

[17]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.

[18]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 No. 95-1692, 1995.

[19]Iollo, Angelo; and Salas, Manuel, D.: Optimum Transonic Airfoils Based on the Euler Equations. ICASE report 96-76, 1996.

[20]Jameson, A.: Aerodynamic Design Via Control Theory. *J. Scientific Computing*, vol. 3, 1988, pp. 23-260.

[21]Jameson, A.; Optimum Aerodynamic Design Using the Control Theory. *Computational Fluid Dynamics Review*, 1995, pp. 495-528.

[22]Jameson, A.; Pierce, N. A.; and Martinelli, L.: Optimum Aerodynamic Design Using the Navier-Stokes Equations. AIAA Paper No. 97-0101, 1997.

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

[24]Marco, Nathalie: "Optimisation de Formes Aerodynamiques 2D et 3D Par Une Methode Multi-Niveau En Maillages Non Structures. Ph.D. Thesis, L'Universite De Nice-Sophia Antipolis, November 1995.

[25]Marcum, D. L.: Generation of Unstructured Grids for Viscous Flow Applications. AIAA Paper No. 95-0212, 1995.

[26]Mohammadi, B.: Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence. AIAA Paper No. 97-0099, 1997.

[27]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 No. 96-2464, 1996.

[28]Newman, James C.; Taylor, Arthur C.; and Barnwell, Richard W.: Aerodynamic Shape Sensitivity Analysis and Design Optimization of Complex Configurations Using Unstructured Grids. AIAA Paper No. 97-2275, 1997.

[29]Pironneau, O.: On Optimum Design in Fluid Mechanics. *J. Fluid Mech.*, vol. 64, 1974, pp. 97-110.

[30]Reuther, J.; and Jameson, A.; Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory. AIAA Paper No. 95-0123, 1995.

[31]Roe, P. L.: Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comp. Phys*., vol. 43, no. 2, 1981, pp. 357-372.

[32]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.

[33]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 No. 94-4262, 1994.

[34]Soemarwoto, Bambang; Multi-Point Aerodynamic Design by Optimization. Ph.D. Thesis, Delft University of Technology, December 1996.

[35]Spalart, P. R.; and Allmaras, S. R.: A One-Equation Turbulence Model for Aerodynamic Flows. AIAA Paper No.92-0439, 1991.

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

[37]Ta'asan, S.: Trends in Aerodynamic Design and Optimization: a Mathematical View Point. AIAA Paper No. 95-1731-CP, 1995.

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

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

[40]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 No. 94-4386.