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. In Ref. 18, the turbulence model used was the Baldwin-Lomax algebraic model, which was differentiated along with the flow equations. Jameson 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.

For unstructured grids, the work to date has been primarily focused on inviscid computations in both two and three dimensions. Refs. 1, 7, 9, 11-13, 17, 19-24, 29, 30. 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, 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, 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>area of control volume</td>
</tr>
<tr>
<td>a</td>
<td>speed of sound</td>
</tr>
<tr>
<td>C</td>
<td>constant used in Sutherland's law for viscosity</td>
</tr>
<tr>
<td>c_l</td>
<td>lift coefficient</td>
</tr>
<tr>
<td>c_d</td>
<td>drag coefficient</td>
</tr>
<tr>
<td>c_b, c_s, c_v</td>
<td>constants used in Spalart-Allmaras turbulence model</td>
</tr>
<tr>
<td>c_w_1, c_w_2, c_w_3</td>
<td>constants used in Spalart-Allmaras turbulence mode</td>
</tr>
<tr>
<td>D</td>
<td>vector of design variables</td>
</tr>
<tr>
<td>D_i</td>
<td>component i of design variable vector</td>
</tr>
<tr>
<td>d</td>
<td>distance to nearest surface</td>
</tr>
<tr>
<td>E</td>
<td>total energy per unit volume</td>
</tr>
<tr>
<td>F</td>
<td>fluxes of mass, momentum, and energy</td>
</tr>
<tr>
<td>F_i</td>
<td>inviscid contribution to fluxes</td>
</tr>
<tr>
<td>F_w</td>
<td>viscous contribution to fluxes</td>
</tr>
<tr>
<td>f, g</td>
<td>components of inviscid fluxes</td>
</tr>
<tr>
<td>f_v, g_v</td>
<td>components of viscous fluxes</td>
</tr>
<tr>
<td>f_1, f_2</td>
<td>functions used in the turbulence model</td>
</tr>
<tr>
<td>f_w, f_1, f_2</td>
<td>functions used in the turbulence model</td>
</tr>
<tr>
<td>I</td>
<td>augmented cost to be minimized</td>
</tr>
<tr>
<td>I_c</td>
<td>cost to be minimized</td>
</tr>
<tr>
<td>K</td>
<td>Karman constant</td>
</tr>
<tr>
<td>M_m</td>
<td>free-stream Mach number</td>
</tr>
<tr>
<td>N</td>
<td>B-spline basis functions</td>
</tr>
<tr>
<td>n</td>
<td>unit normal to boundary of control volume</td>
</tr>
<tr>
<td>P_T</td>
<td>Prandtl number</td>
</tr>
<tr>
<td>P_T_l</td>
<td>turbulent Prandtl number</td>
</tr>
</tbody>
</table>
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

$$\frac{\partial \mathbf{Q}}{\partial t} + \int_{\partial A} \mathbf{F}_i \cdot \hat{n} \, dl - \int_{\partial A} \mathbf{F}_v \cdot \hat{n} \, dl = 0 \quad (1)$$

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

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad (2)$$

$$\mathbf{F}_i = f_i^i + g_i^j = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho v^2 + p \\ (E + p)u \end{bmatrix} + \begin{bmatrix} \rho v \\ -\rho u v \\ p v u \\ (E + p)v \end{bmatrix} \quad (3)$$

and

$$\mathbf{F}_v = f_i^i + g_i^j = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{yy} \end{bmatrix} \hat{i} + \begin{bmatrix} 0 \\ -\tau_{xy} \\ \tau_{yy} \end{bmatrix} \hat{j} \quad (4)$$

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

$$\tau_{xx} = (\mu + \mu_t) \frac{M_2}{Re} (2u_x - v_x) \quad (5)$$

$$\tau_{yy} = (\mu + \mu_t) \frac{M_2}{Re} (2v_y - u_y) \quad (6)$$

$$\tau_{xy} = (\mu + \mu_t) \frac{M_2}{Re} (u_y + v_x) \quad (7)$$

$$q_x = -\frac{M_2}{Re(\gamma - 1)} \frac{\mu + \mu_t}{Pr_\tau} \frac{\partial \tilde{u}}{\partial x} \quad (8)$$

$$q_y = -\frac{M_2}{Re(\gamma - 1)} \frac{\mu + \mu_t}{Pr_\tau} \frac{\partial \tilde{v}}{\partial y} \quad (9)$$

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

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

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

$$\mu = \frac{\mu}{\tilde{\mu}} = \frac{(1 + C^*) (\tilde{T}/\tilde{T}_0)^{3/2}}{\tilde{T}/\tilde{T}_0 + C^*} \quad (11)$$

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

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

\[ \frac{D\tilde{\nu}}{Dt} = \frac{M_\infty}{\sigma Re} \{ \nabla \cdot \left[ (\nu + (1 + c_h)\nu)\nabla \tilde{\nu} \right] - c_h \tilde{\nu} \nabla^2 \tilde{\nu} \} \]

where

\[ f_{v_1} = \frac{x^3}{x^3 + c_{v_1}} \]

\[ \chi = \frac{\tilde{\nu}}{\nu} \]

\[ \tilde{S} = S + \frac{M_\infty}{\text{Re} \kappa^2 d^2} f_{v_2} \]

and

\[ f_{v_2} = 1 - \frac{\chi}{1 + \chi f_{v_1}} \]

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

\[ f_v = g \left( \frac{1 + c_k^2}{g^2 + c_k^2} \right)^{1/6} \]

where

\[ g = r + c_k^2 (r^6 - r) \]

and

\[ r = \frac{M_\infty}{\text{Re} S \kappa^2 d^2} \]

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 \( f_{v_1} \) and \( f_{v_2} \), which are associated with these terms, are not given. After Eq. (12) is solved for \( \tilde{\nu} \), the eddy viscosity is computed as

\[ \mu_\nu = \rho \nu v = \rho \tilde{\nu} f_{v_1} \]

Adjoint Equations

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

\[ I[Q, D, \Psi, X(D)] = I_c(Q, D) + \Psi^T R[Q, D, X(D)] \]

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

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

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

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

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

\[ \delta I = \left( \frac{\partial I_c}{\partial D} + \Psi^T \left( \frac{\partial R}{\partial D} \right) \right) \delta D \]

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, \( Q \), 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 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.
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 x 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 affected 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.](image)

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>
<thead>
<tr>
<th>Table 1. Accuracy of Derivatives for Lift Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite-difference</td>
</tr>
<tr>
<td>( \frac{\partial c_l}{\partial \alpha} )</td>
</tr>
<tr>
<td>( \frac{\partial c_l}{\partial D_1} )</td>
</tr>
<tr>
<td>( \frac{\partial c_l}{\partial D_2} )</td>
</tr>
<tr>
<td>( \frac{\partial c_l}{\partial D_3} )</td>
</tr>
</tbody>
</table>

---

Footnotes:

1. Hou et al.
2. Ref. 31
3. Ref. 32
4. Ref. 38
5. Ref. 1
6. Ref. 2
7. Ref. 3
8. Ref. 10
9. Ref. 25
range in the number of design variables and in surface fidelity can be
where \( M \) is the number of input coordinates. The knot sequence is
defined as the sequence of parameters, starting from zero:
that corresponds to each input coordinate and the number of the
control points correspond to the end points of the B-spline.
and the end of the knot sequence, respectively, so that the
maximum values of the parameter \( s \) appear \( n \) times at the beginning
and are more numerically meaningful than the original input coordinates.
The following is a description of the B-spline representation and the
nates, a B-spline of specified order and with a specified number of
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.
and are more geometrically meaningful than the original input coordinates.
The following is a description of the B-spline representation and the
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 parameterization of the spline and a knot sequence and are defined recursively as follows:
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 and are considered to be 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:
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 \( N_j^i \) 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:

\[
\bar{x}_j = \sum_{i=0}^{M} \bar{x}_i N_j^i (s_j)
\]

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 \( \bar{x}^* \) 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
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.

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
polyhedral 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 poly-
nomial 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 rep-resented
in a single curve.

Spline fitting of input coordinates
After splining the surface and setting the limits of the design vari-
ables, 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 de-
ting, 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, 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 $1 \times 10^{-5}$ 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 $1.5 \times 10^{-1}$ to $3.0 \times 10^{-7}$, and the root mean square of the gradients is reduced from 1.4 to $5.8 \times 10^4$. 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.

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 $1.0 \times 10^{-3}$. (See Fig. 7.) The free-stream Mach number is 0.725, the angle of attack is $2.54^\circ$, 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. 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 co-
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, have been demonstrated for inviscid flow computations 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.
An aerodynamic design algorithm for turbulent flows using unstructured grids is described. The current approach uses adjoint (costate) variables for obtaining derivatives of the cost function. The solution of the adjoint equations is obtained 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 example computations are shown. In addition, a user interface is described which significantly reduces the time required for setting up the design problems. Recommendations on directions of further research into the Navier Stokes design process are made.
References