AIAA JOURNAL
Vol. 35, No. 9, September 1997

Practical Three-Dimensional Aerodynamic Design and Optimization Using Unstructured Meshes

Jonathan Elliott* and Jaume Peraire†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

A methodology for performing optimization on three-dimensional unstructured grids based on the Euler equations is presented. The same, low-memory-cost explicit relaxation algorithm is used to resolve the discrete equations that govern the flow, linearized direct, and adjoint problems. The analysis scheme is a high-resolution local-extremum-diminishing-type scheme that uses Roe decomposition for the dissipative fluxes. Mesh movement is performed in such a way that optimization of arbitrary geometries is allowed. The parallelization of the algorithm, which permits its extension to optimization of realistic, complete aircraft geometries, is presented. Two sample optimizations are performed. The first is the inverse design of a transonic wing/body configuration using a surface target pressure distribution found by analyzing the geometry for known design variable deflections. The second exercise is the inverse design of a business jet configuration consisting of wing, body, strut, nacelle, horizontal fin, and vertical fin. The surface target pressure distribution in this case is provided by an analysis of the configuration with no strut-nacelle.

I. Introduction

ADVANCES in both computer technology and algorithms have allowed computational fluid dynamics (CFD) to occupy a vital part of the aerospace, marine, and automotive vehicle design processes, significantly reducing the costs associated with testing of physical models. However, it is possible that even greater dividends are being missed due to the absence, to a large extent, of formal aerodynamic/hydrodynamic optimization in these design processes. Indeed, the coupling of aerodynamic sensitivities with, for example, structural weight and propulsion sensitivities in a multidisciplinary design optimization (MDO) process promises even richer rewards and is thought to be vital to ensure the viability of future vehicles such as the High Speed Commercial Transport (HSCT).

The main obstacle to aerodynamic optimization has been the high cost associated with the calculation of sensitivities. However, in recent years impressive efforts have been made using structured grids that allow for efficient sensitivity calculation.[1,2]

For the solution of flowfields over complex domains, unstructured grid methods have several advantages over their alternative structured counterparts. For example, the time associated with grid generation for multiblock schemes and the errors associated with lack of conservation and grid generation problems (to a lesser extent) for Chimera-type schemes limit their usefulness. Unstructured mesh methods using tetrahedra offer the ability to model flows around complex geometries and can incorporate mesh adaptive procedures. The issue of weaknesses associated with resolving boundary layers and viscous wakes is currently being tackled, and it is believed that unstructured methods will have viscous resolution capabilities matching those of structured grids in the not too distant future.

Transonic/supersonic wing design in the presence of strut-nacelles and aircraft high-lift system design are examples of tasks associated with inherently complex geometric domains for which unstructured flow solvers, with automatic mesh generation, have a decided advantage. Furthermore, the capability to refine and derefine unstructured meshes introduces the possibility of distributing the mesh degrees of freedom more efficiently and therefore concentrating elements where they are required to reduce the local error to a desired level.

Received Oct. 15, 1996; accepted for publication April 8, 1997.

*Graduate Research Assistant, Department of Aeronautics and Astronautics, Room 37-471, 77 Massachusetts Avenue. E-mail: jkelliot@mit.edu. Member AIAA.

†Associate Professor, Department of Aeronautics and Astronautics, Room 37-451, 77 Massachusetts Avenue. Email: peraire@mit.edu. Senior Member AIAA.

Hence a capability to perform optimization using unstructured flow solvers is perceived to be an important one. We present a capability for performing practical three-dimensional optimization on unstructured grids in which the cost of the gradient calculation is practically independent of the number of design variables. This is done using the discrete adjoint method[9–11,13,21] and solving the resulting system with the same explicit multistage timestepping algorithm used to solve the flow equations.

For structured grids, generation of analytic sensitivities based on the Euler equations in two and three dimensions has recently been performed using the analytic adjoint approach,[13,18,19] in which flow and adjoint equations are discretized separately. The same relaxation algorithm was used to solve flow and adjoint equations. In Ref. 13 an alternating direction method with upwind biasing and multigrid was used to drive the residuals to zero. In Ref. 18 an explicit multistage relaxation scheme was used in a scheme incorporating scalar dissipation with pressure switching to allow shock capturing.

Newman et al.[15] and Newman and Taylor[14] presented a capability to generate two-dimensional and, recently, three-dimensional unstructured sensitivities, respectively, for a second-order scheme using Van Leer flux vector splitting. This was done using an incremental iterative implicit time-marching scheme. They used the discrete sensitivity analysis approach in which differentiation of the discrete flow analysis scheme is performed directly. They extended this capability to second order by using the first-order Jacobian on the left-hand side with its lower required core memory (due to the smaller stencil) and the second-order Jacobian on the right-hand side.

A three-dimensional unstructured sensitivity capability based on the Euler equations was also reported in Ref. 5 by the present authors. The extension of this capability to the optimization of realistic, complete aircraft geometries is discussed in this paper. This extension has been made possible (for practical design times) by the parallelization of the algorithm.

All sensitivity analyses reported here use the discrete sensitivity analysis approach. Introduction of an artificial time-dependent term into the adjoint and sensitivity equations permits their efficient solution by using a relaxation scheme identical to the flow analysis scheme. The flow variables are discretized on two- and three-dimensional unstructured grids. Local-extremum-diminishing (LED) properties are obtained by introducing just enough dissipation to upwind each characteristic component through the use of a Roe-averaged flux Jacobian.[12] Minmod-limited antidiffusion is added to improve the accuracy of the scheme to second order in smooth regions of the solution. Furthermore, the mesh movement algorithm can handle arbitrary geometries and consists of finding the equilibrium position of the unstructured mesh considered as a system of interconnected springs.

Section II presents the details of the flow and sensitivity analysis algorithms. Section III presents applications of these algorithms to sample optimization exercises. Validation of the sensitivities produced by the algorithm has been presented in Ref. 5, and here we focus on the details of the optimization exercises. We summarize the capabilities developed and indicate future research directions in Sec. IV. Note that the algorithm extends trivially to optimization of other more general cost functions such as drag.

II. Flow and Sensitivity Analysis Algorithms

Flow Analysis

For illustration purposes, the algorithm is described based on the two-dimensional algorithm. The three-dimensional algorithm is a direct extension thereof. The integral form of the Euler equations reads

$$\int_\Omega \frac{\partial \boldsymbol{U}}{\partial t}\, dx\, dy = -\int_\Omega \nabla \cdot \vec{\mathbb{F}}\, dx\, dy = -\oint_\Gamma \vec{\mathbb{F}} \cdot \hat{n}\, ds \tag{1}$$

where

$$\boldsymbol{U} = [\rho, \rho u, \rho v, \rho E]^T \tag{2}$$

and Ω is a closed control volume with boundary Γ and $\vec{\mathbb{F}}$ is the usual flux tensor. The discretization of $\boldsymbol{U}$ and $\vec{\mathbb{F}}$ on an unstructured triangulation of the domain is accomplished using piecewise linear polynomials. The spatial discretization is completed by using the finite volume formulation with control volumes consisting of all triangles having vertex i and performing exact integration around the outer boundary of this control volume. For illustrative purposes, the semidiscrete form is shown here to be advanced in time using a forward Euler scheme. A multistage Runge–Kutta was used in the reported computations. After discretization, the following set of equations results:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \sigma(\Delta t/A)_i \boldsymbol{R}_i(\boldsymbol{U}^n) \tag{3}$$

where a cell-dependent timestep Δt_i has been used for convergence acceleration, A_i is the area of the control volume, and σ represents the local Courant–Friedrichs–Lewy (CFL) number. The residual at a point can be calculated as a sum over all of the edges that contain that point[17]:

$$\boldsymbol{R}_i = \sum_i^K [\boldsymbol{R}]_{ik} \tag{4}$$

where ik represents the edge connecting nodes i and k and the residual increment for an interior edge is

$$[\boldsymbol{R}]_{ik} = -\frac{\vec{\mathbb{F}}_i + \vec{\mathbb{F}}_k}{2} \cdot \boldsymbol{S}_{ik} \tag{5}$$

where $\boldsymbol{S}_{ik}$ is the area vector associated with edge ik and calculated as follows:

$$\boldsymbol{S}_{ik} = (y_{k+1} - y_{k-1}, x_{k-1} - x_{k+1}) \tag{6}$$

(see Fig. 1).

The residual contribution for a wall boundary edge is

$$[\boldsymbol{R}_i^b]_{il} = -\left\{ \frac{p_i + p_l}{2} \left(\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right) \right\} \cdot \boldsymbol{S}_b - \frac{\vec{\mathbb{F}}_i + \vec{\mathbb{F}}_l}{2} \cdot \boldsymbol{S}_{l,l+1} \tag{7}$$

where the second term ensures closure of the control volumes (Fig. 1) and

$$\boldsymbol{S}_b = (y_l - y_i, x_i - x_l) \tag{8}$$

For a far-field boundary edge, the contribution to the residual becomes

$$[\boldsymbol{R}_i^b]_{il} = -\frac{\vec{\mathbb{F}}_i^b + \vec{\mathbb{F}}_l^b + 2\vec{\mathbb{F}}_i}{2} \cdot \boldsymbol{S}_b - \frac{\vec{\mathbb{F}}_i + \vec{\mathbb{F}}_l}{2} \cdot \boldsymbol{S}_{l,l+1} \tag{9}$$

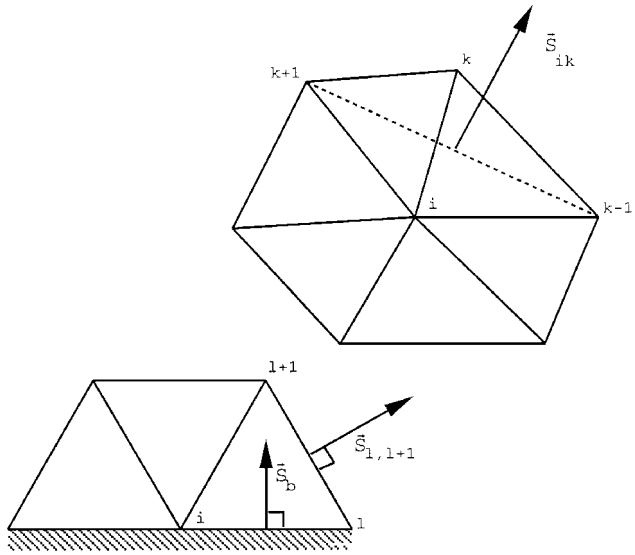

Fig. 1 Area vectors for boundary and interior edges.

where the superscript b indicates that the flux includes a first-order diffusive contribution that achieves upwinding using Roe averaging, as described next, between the given node and the freestream state vector $\boldsymbol{U}_\infty$:

$$\vec{\mathbb{F}}_i^b \cdot \boldsymbol{S}_b = \frac{\vec{\mathbb{F}}_i + \vec{\mathbb{F}}_\infty}{2} \cdot \boldsymbol{S}_b - \frac{S_b}{2} |\boldsymbol{A}|(\boldsymbol{U}_i - \boldsymbol{U}_\infty)$$

where $\boldsymbol{A} = \partial(\vec{\mathbb{F}} \cdot \hat{n}_b)/\partial \boldsymbol{U}$ is the standard Roe matrix[20] resolved in the direction of the boundary normal $\hat{n}_b = \boldsymbol{S}_b/S_b$ and can be formed from the Roe averaging of $\boldsymbol{U}_i$ and $\boldsymbol{U}_\infty$. These expressions for the fluxes at the boundaries ensure that correct boundary conditions are imposed and that we have $R(\boldsymbol{U}) = 0$ at steady state everywhere in the domain, including the nodes on the boundary. This is important for it guarantees that the implicit function theorem holds and that correct sensitivities will be calculated (as discussed in the Sensitivity Analysis section).

To stabilize the scheme, a dissipative term, premultiplied by a matrix coefficient, is added to the preceding equations for interior and farfield boundary fluxes:

$$\boldsymbol{D}_{ik} = (S/2)|\boldsymbol{A}(\boldsymbol{U}_i, \boldsymbol{U}_k)|(\boldsymbol{U}_i - \boldsymbol{U}_k) = (S/2)\boldsymbol{P}|\Lambda|\boldsymbol{P}^{-1}(\boldsymbol{U}_i - \boldsymbol{U}_k) \tag{10}$$

where $S = |\boldsymbol{S}_{ik}|$ and $\boldsymbol{A}$ is now resolved in the direction of the face normal $\hat{n}_{ik} = \boldsymbol{S}_{ik}/S$ and is based on a Roe averaging of $\boldsymbol{U}_i$ and $\boldsymbol{U}_k$. This ensures positivity of the coefficients multiplying increments of characteristic variables, $\Delta \boldsymbol{W}_{ik} = P^{-1}\Delta \boldsymbol{U}_{ik}$. However, the added dissipation makes the scheme first-order accurate and causes it to be too diffusive. To correct this, we add antidiffusion that is limited in a nonlinear manner to maintain the LED properties. Hence the diffusive flux takes the form

$$\boldsymbol{D}_{ik} = (S/2)\boldsymbol{P}|\Lambda|[\boldsymbol{P}^{-1}\Delta \boldsymbol{U}_{ik} - L(\boldsymbol{P}^{-1}\Delta \boldsymbol{U}^-, \boldsymbol{P}^{-1}\Delta \boldsymbol{U}^+)] \tag{11}$$

Several options are available for the limiter function $L(u, v)$. Here we simply take

$$L(u, v) = \text{minmod}(u, v) \tag{12}$$

and $\Delta \boldsymbol{U}^-$, $\Delta \boldsymbol{U}^+$ are formed exactly from the gradients in the cells that either end of the edge points into.[3] For example,

$$\Delta \boldsymbol{U}^+ = (\nabla \boldsymbol{U})^+ \cdot (\boldsymbol{x}_i - \boldsymbol{x}_k) = \epsilon_{ir}\Delta \boldsymbol{U}_{ir} + \epsilon_{is}\Delta \boldsymbol{U}_{is} \tag{13}$$

where the nodes in the element that the edge points into have been labeled i, r, s and ϵ_{ir}, ϵ_{is} are, by construction, always positive. The addition of antidiffusion causes the scheme to now resemble a high-order MUSCL scheme. For further details of similar algorithms, see Refs. 4, 16, and 17.

Timestep Calculation

It was found for three-dimensional cases that use of a linearized Fourier stability analysis to determine the allowable timestep[6] allows convergence to machine accuracy only for very low CFL numbers. For practical CFL levels, limit cycling was observed. It is found that, for analysis, the solutions produced are acceptable when this limit cycling occurs at residual levels that are low. However, the sensitivity analyses performed for baseline state vectors $\boldsymbol{U}$ whose solution convergence histories exhibited this limit-cycling behavior can be unstable at the CFL numbers at which the analyses were run. For example, at $\sigma = 0.5$, limit cycling was typically observed for flow analysis and the sensitivity analysis was unstable, whereas for $\sigma = 0.05$, convergence to machine accuracy was typically found for both flow and sensitivity analyses, although this was very slow.

The problem was solved by supplementing the local timestep calculation with a check on violations of the TVD (Total Variation Diminishing) CFL-like condition. For a scalar equation, TVD conditions for the semidiscrete form of a scheme that can be expressed as

$$\frac{du_0}{dt} = \sum_{i=1}^{K} c_{0i}(u_i - u_0) \tag{14}$$

are

$$c_{0i} > 0 \qquad i = 1, \ldots, K \tag{15}$$

where K represents the total number of nodes that contribute to the residual at node i. When the forward Euler time discretization is performed, the fully discrete scheme is

$$u_0^{n+1} = u_0^n + \Delta t \sum_{i=1}^{K} c_{0i}(u_i - u_0) = \sum_{i=1}^{K} d_{0i} u_i + \mu_0 u_0 \tag{16}$$

In this case, the TVD conditions become

$$d_{0j} = \Delta t c_{0j} > 0 \qquad j = 1, \ldots, K \tag{17}$$

$$\mu_0 = 1 - \Delta t \sum_{i=1}^{K} c_{0i} > 0 \tag{18}$$

The latter is a multidimensional form of the TVD CFL-like condition first discussed in Ref. 8. For our scheme, this takes the form

$$\mu_i = 1 - \Delta t_i \left[\sum_{i=1}^{\text{edges}} (C_{ki}^{\text{int}} + C_{ki}^{\text{upw}} + C_{ki}^{\text{ad}}) + \sum_{i=1}^{\text{bou}} C_{kip}^{\text{bou}} \right] \tag{19}$$

where the four coefficients in the inner parentheses represent maximum multipliers of characteristic differences, $\Delta \boldsymbol{W} = \boldsymbol{P}^{-1} \Delta \boldsymbol{U}$. They correspond to interior "physical" fluxes, first-order diffusive fluxes, antidiffusive fluxes, and boundary fluxes, respectively, and are given by

$$C_{ki}^{\text{int}} = C_{ki}^{\text{upw}} = (\sigma / A_i) S \lambda_{\max} \tag{20}$$

$$C_{ki}^{\text{ad}} = (\sigma / A_i)(\epsilon_{ir} + \epsilon_{is}) S \lambda_{\max} \phi(r)\big|_{\max} \tag{21}$$

where $\phi(r)$ is the limiter function first introduced in Ref. 22 and is related to $L(u, v)$ in Eq. (12) by $L(u, v) = u\phi(v/u)$. For the present case, with the minmod limiter, $\phi_{\max} = 1$.

The supplemental TVD check consisted of modifying Δt_i such that $\mu_i > \mu_t = 0$ was obtained. In fact, it was found that this led to rather small values for Δt, and consequently, slow convergence and subsequent investigation revealed that using a threshhold value of $\mu_t < 0$ allowed quick convergence without limit cycling. An example presented in Ref. 5 shows dramatically improved convergence properties with $\mu_t = -1$ for an analysis case. Furthermore, the sensitivity calculation, with an identical Δt, was found to be stable for the CFL number at which the analysis scheme had been run.

Sensitivity Analysis

Gradient-based optimization methods require the gradient, or sensitivity, of a cost function $I(\boldsymbol{U}, \beta)$ with respect to the vector of design variables β. Sensitivities can be calculated using either the direct or the adjoint method. In either case, the implicit function theorem, which results from direct differentiation of the steady-state discrete residual equation, is used:

$$\boldsymbol{R}(\boldsymbol{U}, \beta) = 0 \tag{22}$$

with respect to the design variables, producing

$$\frac{d\boldsymbol{R}}{d\beta} = 0 \Rightarrow \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \frac{\partial \boldsymbol{U}}{\partial \beta} = -\frac{\partial \boldsymbol{R}}{\partial \beta} \tag{23}$$

We can calculate the cost function sensitivity as follows:

$$\frac{dI}{d\beta} = \frac{\partial I}{\partial \beta} + \frac{\partial I}{\partial \boldsymbol{U}} \frac{\partial \boldsymbol{U}}{\partial \beta} \tag{24}$$

With the direct method, one solves relation (23) directly for $\boldsymbol{U}_\beta = \partial \boldsymbol{U} / \partial \beta$, whereas with the adjoint method, one solves for the adjoint variables ψ that satisfy

$$\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \right)^T \psi = \left(\frac{\partial I}{\partial \boldsymbol{U}} \right)^T \tag{25}$$

The cost function sensitivities are then calculated according to

$$\frac{dI}{d\beta} = \frac{\partial I}{\partial \beta} - \frac{\partial I}{\partial \boldsymbol{U}} \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \right)^{-1} \left(\frac{d\boldsymbol{R}}{d\beta} \right) = \frac{\partial I}{\partial \beta} - \psi^T \left(\frac{d\boldsymbol{R}}{d\beta} \right) \tag{26}$$

It is possible to adopt exactly the same relaxation algorithm to solve Eqs. (22), (23), and (25). Hence to find $\boldsymbol{U}_\beta$ or ψ, we can drive the following linear equations to steady state:

$$\boldsymbol{U}_{\beta_i}^{n+1} = \boldsymbol{U}_{\beta_i}^n + \sigma \left(\frac{\Delta t}{A} \right)_i \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \boldsymbol{U}_\beta + \left(\frac{d\boldsymbol{R}}{d\beta} \right)_U \right] \tag{27}$$

$$\psi_i^{n+1} = \psi_i^n + \sigma \left(\frac{\Delta t}{A} \right)_i \left(\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{U}} \psi - \frac{\partial I}{\partial \boldsymbol{U}}^T \right) \tag{28}$$

For the direct method, Eq. (23) is solved for each design variable, whereas for the adjoint method, Eq. (25), requiring a similar computational effort, needs to be solved only once, assuming that only one objective function is considered. This represents a large cost savings when the number of design variables is large.

These schemes are stable because the error modes are subject to the same amplification matrix as those for the linearized flow analysis scheme (3):

$$\Delta \boldsymbol{U}_i^{n+1} \approx \Delta \boldsymbol{U}_i^n + \sigma \left(\frac{\Delta t}{A} \right)_i \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \Delta \boldsymbol{U}^n \tag{29}$$

The stability characteristics of Eqs. (27) and (28) are the same because transposing a matrix does not change its eigenvalues. Indeed, for all calculations performed, the asymptotic convergence rate of analysis, direct, and adjoint schemes was found to be very close.

The matrices $\partial \boldsymbol{R} / \partial \boldsymbol{U}$ and $\partial \boldsymbol{R} / \partial \boldsymbol{X}$ [the residual sensitivities to grid nodal coordinates, which are needed in the calculation of $(\partial \boldsymbol{R} / \partial \beta)$] are constructed in exactly the same fashion as $\boldsymbol{R}(\boldsymbol{U})$, namely by looping over the edges of the triangulation. This allows for debugging and comparison with finite difference quantities at the edge level. The equation corresponding to Eq. (4) for sensitivities is

$$\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{U}_j} \boldsymbol{U}_{\beta_j} = \sum_{i=1}^{K} \left[\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{U}_j} \right]_{ik} \boldsymbol{U}_{\beta_j} \tag{30}$$

where, for example, for an interior edge,

$$\left[\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{U}_j} \right]_{ik} \boldsymbol{U}_{\beta_j} = \left\{ -\frac{\delta_{ij}}{2} \frac{\partial \mathbb{F}_k}{\partial \boldsymbol{U}} \cdot \boldsymbol{S}_{ik} + \frac{\partial}{\partial \boldsymbol{U}_j} (S \boldsymbol{P} |\Lambda|) \times [\Delta \boldsymbol{W}_{ik} - L(\Delta \boldsymbol{W}^-, \Delta \boldsymbol{W}^+)] + (S \boldsymbol{P} |\Lambda|) \frac{\partial}{\partial \boldsymbol{U}_j} \times [\Delta \boldsymbol{W}_{ik} - L(\Delta \boldsymbol{W}^-, \Delta \boldsymbol{W}^+)] \right\} \boldsymbol{U}_{\beta_j}$$

For the adjoint problem, the transposed Jacobian is required:

$$\left[\frac{\partial \boldsymbol{R}_j}{\partial \boldsymbol{U}_i}\right]^T \psi_i = \sum_{i\in K}\left[\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{U}_j}\right]_{ik} \psi_i \tag{31}$$

The chain rule is applied to the forcing term derivatives. Thus, we have

$$\frac{\partial \boldsymbol{R}}{\partial \beta} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} \tag{32}$$

and $d\boldsymbol{X}/d\beta$, the variation of the nodal coordinates $\boldsymbol{X}$ with respect to the design variables β, is also calculated by looping over the edges.

Grid motion is accomplished using Jacobi-like iteration based on the force equilibrium of springs, one of which lies along each edge of the triangulation. The relaxation scheme is

$$\delta_i^{n+1} = \frac{\sum k_{ij}\delta_j^n}{\sum k_{ij}} \tag{33}$$

where $k_{ij} = |\Delta x_{ij}|^{-1}$ and

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \delta^{n+1} \tag{34}$$

and δ is the vector of nodal displacements. Therefore the grid sensitivity is simply

$$\frac{d\boldsymbol{X}}{d\beta} = \frac{\partial \delta^N}{\partial \beta} = \frac{\partial \delta^N}{\partial \delta^{N-1}}\frac{\partial \delta^{N-1}}{\partial \delta^{N-2}}\cdots\frac{\partial \delta^1}{\partial \delta^0}\frac{\partial \delta^0}{\partial \beta} \tag{35}$$

where we have performed N Jacobi iterations of the grid relaxation scheme. The factors of the form $\partial\delta^i/\partial\delta^{i-1}$ in the preceding product are found from Eq. (33), whereas $\partial\delta^0/\partial\beta$ is determined from the expression for the wing surface design variable perturbation. It is simple and inexpensive to check the correctness of these sensitivity calculations vs finite difference, and this has become our standard practice.

Much discussion of boundary conditions for the adjoint equation has taken place in the literature.[9–11,13] We take the approach that, because we have implicitly included the boundary conditions in the expression for the flow equation residual, the grid and flow derivatives of these residuals will automatically include the boundary conditions for both adjoint and flow sensitivity variables.

Implementation Issues

The resulting solution algorithm is readily parallelized using domain decomposition and explicit message-passing. The implementation is such that the message-passing libraries MPI or PVM can be employed and the code can run on a variety of platforms, ranging from clusters of workstations to mainframe parallel machines, such as the IBM SP2 and the Cray T3D. At present, the mesh generation is carried out in a serial manner and the computational mesh is decomposed after the mesh generation is completed. It has been decided to follow a strategy in which the mesh is partitioned in such a way that edges belong to a single subdomain, whereas nodal points may be shared between several subdomains. The domain is decomposed using a coordinate bisection algorithm. All of the communication arrays, necessary for the transfer of information between the subdomains, are evaluated during the domain partioning stage. Within each subdomain, the edges are subdivided into two groups. One group contains the interdomain boundary nodes, i.e., the nodes requiring communication. The second group contains interior nodes only. Typically, a loop over all of the edges containing boundary nodes is performed first. The values accumulated at the boundary nodes are then broadcast to the neighboring domains. The interior edges are considered next, and the information from the neighboring domains is received. In this way, communication and computation take place concurrently. The parallel efficiency achieved using this strategy with 12 processors on the IBM SP2 at the Massachusetts Institute of Technology is higher than 95% for the simulation of practical problems. With this implementation, the flow analysis code requires 0.32 μs per time step per mesh point, whereas the adjoint requires 2.42 μs.

In Ref. 5, it was found that the edge contributions to the $\partial\boldsymbol{R}/\partial\boldsymbol{U}$ matrix could be stored easily for the two-dimensional case. Eliminating the cost of forming this matrix every iteration lowers the per-iteration cost significantly. It was found that, in two dimensions, the sensitivity calculation CPU cost was approximately half that of the flow analysis, whereas it was a factor of eight larger (than flow analysis) in three dimensions, where saving these terms on an edge basis was not done originally due to the overwhelmingly large memory requirements for a serial implementation. However, for small three-dimensional cases it was found that saving the $\partial\boldsymbol{R}/\partial\boldsymbol{U}$ matrix reduced the per-iteration cost by a factor of about four. Furthermore, the parallel implementation allows storing of $\partial\boldsymbol{R}/\partial\boldsymbol{U}$ for much larger systems because only those values corresponding to the local domain and interface regions need to be stored.

III. Results

Wing/Body Optimization

This case was based on a seed geometry that included a fuselage with circular cross sections and a wing lofted by placing NACA0012 sections into a planform with the following parameters: leading edge sweep $\Lambda_{LE} = 25$ deg, aspect ratio $A_r = 9$, and taper ratio $\lambda = 0.2$. Freestream conditions were chosen to be $M_\infty = 0.8$ and $\alpha = 1.25$ deg.

The baseline grid and pressure contours are shown in Figs. 2 and 3, respectively. The mesh contains 65,000 nodes and 350,000 elements. This optimization case was performed using the serial algorithm. Four design variables were used. The associated mode shapes were chosen to be products of linear functions $g_j(y)$ in the spanwise direction and NACA four-series meanline [$f_1(x/c)$ with maximum at 40% chord] and thickness [$f_2(x/c)$] functions in the chordwise direction. Specifically, the wing upper and lower surface parameterizations are

$$\begin{aligned} z_{\text{upper}}(x,y) &= z^0_{\text{upper}}(x,y) + \sum^{N_{\text{des}}} \beta_i h_i(x,y) \\ &= z^0_{\text{upper}}(x,y) + \sum_{i}^{N_i}\sum_{j}^{N_j} \beta_{k_{ij}} f_i(x/c) g_j(y) \\ z_{\text{lower}}(x,y) &= z^0_{\text{lower}}(x,y) + \sum^{N_{\text{des}}} \varphi_i \beta_i h_i(x,y) \\ &= z^0_{\text{lower}}(x,y) + \sum_{i}^{N_i}\sum_{j}^{N_j} \varphi_{k_{ij}} \beta_{k_{ij}} f_i(x/c) g_j(y) \end{aligned} \tag{36}$$

where $k_{ij} = i + 2(j-1)$; φ_k is -1 and 1 for thickness and camber design variables, respectively; and z^0_{upper}, for example, represents the baseline upper surface. The functions f_i and g_j for this case are summarized in Table 1.

A target pressure distribution was generated by perturbing the camber design variables by an amount corresponding to vertical movement of the mean line of $\Delta(y_c/c)(x/c = 0.4) = 0.02$ (at the spanwise design variable maxima) and to a change in maximum thickness of $-0.04c$. Hence the geometry of the target corresponds closely to a linear loft into the baseline planform of NACA2408

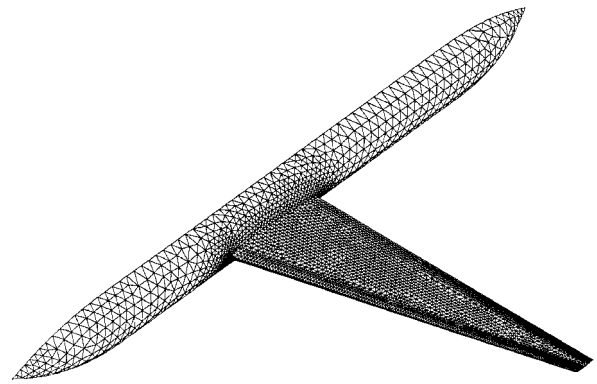

Fig. 2 Surface grid for baseline three-dimensional wing/body calculation.

Table 1 Surface parameterization for wing/body problem

Function	Definition	Range
f_1	$[0.8(x/c) - (x/c)^2]/0.16$	$0 < x/c < 0.4$
	$[0.2 + 0.8(x/c) - (x/c)^2]/0.7056$	$0.4 < x/c < 1$
f_2	$0.2969\sqrt{(x/c)} - 0.12810(x/c) - 0.35160(x/c)^2 + 0.28430(x/c)^3 - 0.1015(x/c)^4$	$0 < x/c < 1$
g_1	$(y_{\text{midspan}} - y)/(y_{\text{midspan}} - y_{\text{root}})$	$y_{\text{root}} < y < y_{\text{midspan}}$
	$(y - y_{\text{midspan}})/(y_{\text{tip}} - y_{\text{midspan}})$	$y > y_{\text{midspan}}$
g_2	0	$y_{\text{root}} < y < y_{\text{midspan}}$
	$(y - y_{\text{midspan}})/(y_{\text{tip}} - y_{\text{midspan}})$	$y > y_{\text{midspan}}$

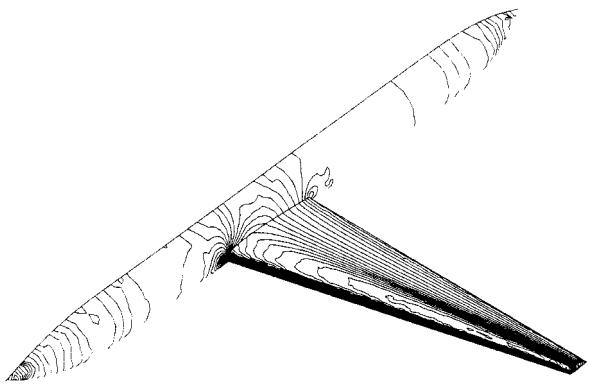

Fig. 3 Surface pressure contours for three-dimensional wing/body at $M = 0.8$ and $\alpha = 1.25$ deg; pressure from 0.6464 to 1.7296.

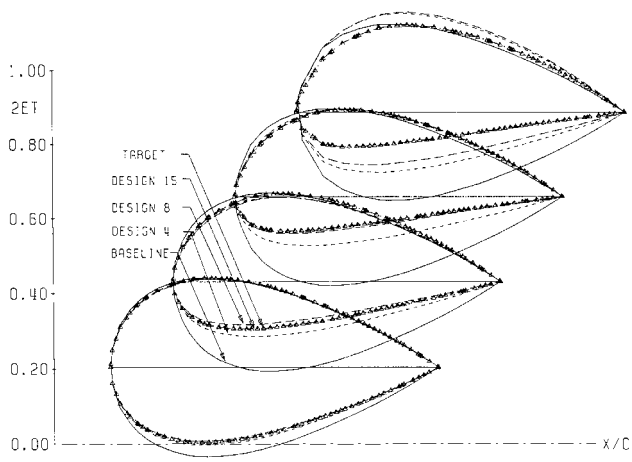

Fig. 4 Geometry evolution for wing/body optimization exercise.

sections from $\eta = 55$ to 100% and a NACA0012 section at the wing-body intersection. This represents a one-third reduction in thickness and a very large camber change, both of which are probably much larger than the magnitudes of thickness and camber changes that might typically be found in a transport wing design process. They therefore represent a more challenging target than would typically be found.

Using the resulting pressure distribution as the target pressure p_{targ} in the expression for the cost function, cost function sensitivities were calculated by the adjoint method. Modal design optimization was performed using the BFGS method. The BFGS method is a quasi-Newton method in which an increasingly accurate estimate of the Hessian matrix H is made based on changes in the gradient vector along the optimization path.[7] Resulting geometry and surface pressure evolutions are shown in Figs. 4 and 5, whereas the evolution of design variables and cost function is shown in Fig. 6. Note that, after the eighth iteration, the BFGS optimization process had to be restarted because it was found that the optimization had reached a local minimum. This is not to be unexpected considering the large design variable changes that were being demanded. Also, at this point the thickness design variables were rescaled such that unit thickness design variable change caused approximately the same

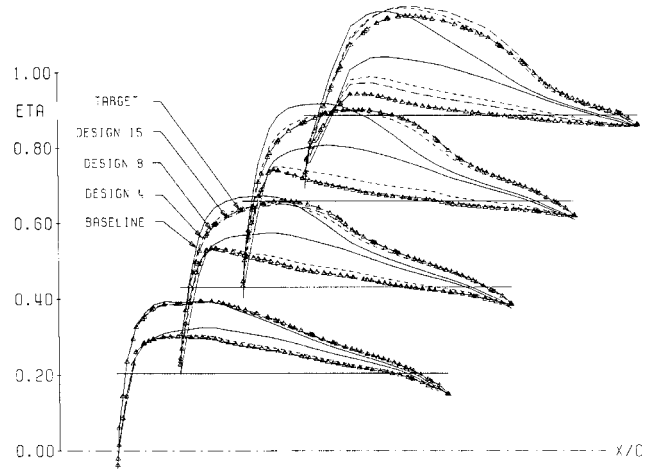

Fig. 5 Pressure evolution for wing/body optimization exercise.

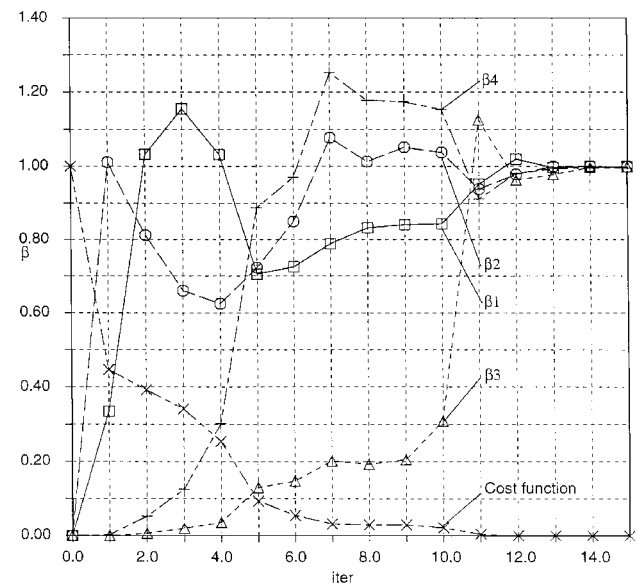

Fig. 6 Cost function and design variable evolution for wing/body optimization exercise.

magnitude of maximum airfoil surface movement as unit camber design variable change. It can be seen that after the restart/rescaling action, the design process rapidly drives geometry and pressure to the target distributions. The robust convergence to the target, in addition to validation exercises (such as those performed in Ref. 5), provides confidence that the sensitivities are being accurately calculated by the current scheme.

Business Jet Optimization

The second optimization example presented here is that of a business jet configuration consisting of wing, body, horizontal and vertical fins, and fuselage-mounted engines. Freestream conditions are $M_\infty = 0.85$ and $\alpha = 2$ deg. Engine exhaust total pressure and temperature increases are not modeled, i.e., flow-through conditions are assumed. The baseline grid and pressure contours are shown in Figs. 7 and 8, respectively. The grid contains 160,000 nodes and 860,000 elements. The target pressure distribution was chosen to be that found on the wing when the strut-nacelle assembly is removed at the same conditions. This pressure distribution is shown in Fig. 9. It can be seen that the "clean wing" pressure distribution is more desirable as it carries more lift inboard due to the shock being much farther aft. Six design variables whose associated mode shapes were combinations of three chordwise functions and two spanwise functions were used. The selected design variables result in a wing parameterization given by Eq. (36), and the functions f_i and g_j for this case are listed in Table 2. Note that the chordwise functions are given by a shear function (which is similar to a twist variable for small geometry perturbations) and two Hicks–Henne functions. The

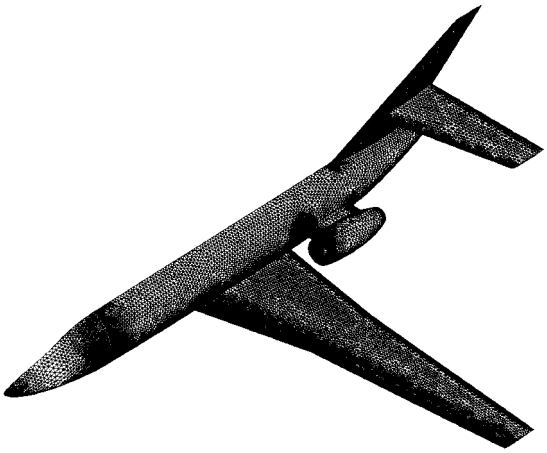

Fig. 7 Surface grid for baseline business jet calculation.

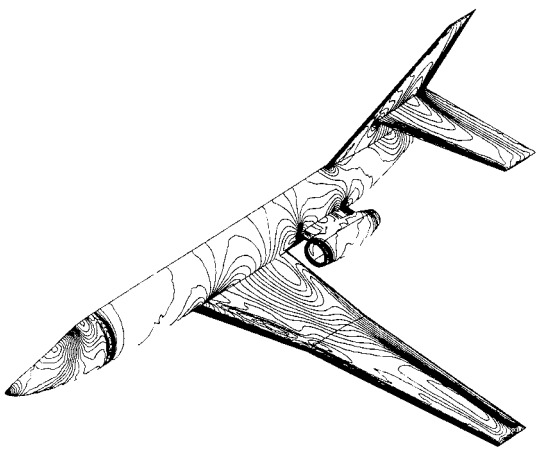

Fig. 8 Baseline surface pressure contours for business jet at $M = 0.85$ and $\alpha = 2$ deg; pressure from 0.3398 to 1.6767.

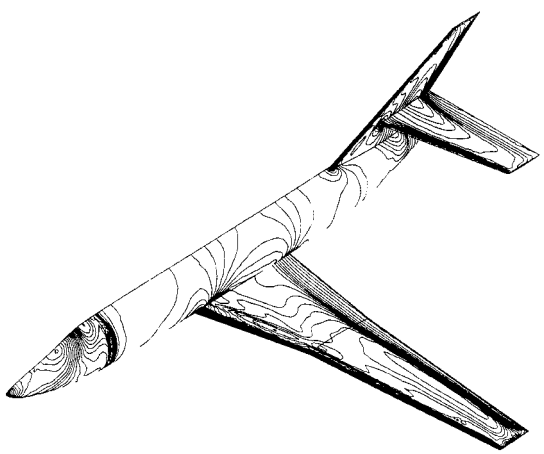

Fig. 9 Target surface pressure contours for business jet at $M = 0.85$ and $\alpha = 2$ deg; pressure from 0.3398 to 1.6770.

spanwise functions have maxima at the wing body intersection (y_{root}) and at the nacelle centerline (y_{engine}). They are both zero outboard of a location y_2, which lies halfway between the engine centerline and the planform break. Slope and function continuity in the final geometry is ensured with this spanwise variation.

The grid perturbation algorithm is more challenging for this case because movement of the wing body intersection requires nodes lying on the fuselage surface to move also. This is to avoid generation of highly skewed elements or even elements with negative volume. This was done by taking advantage of the circular geometry of the fuselage sections. Points close to the wing/body intersection were moved circumferentially by an amount determined by the product of a function g, taken to be linear in θ (the angular coordinate for the circular section associated with the local value of x), and the

Table 2 Surface parameterization for business jet problem

Function	Definition	Range
f_1	x/c	$0 < x/c < 1$
f_2	$\sin[\pi(x/c)^{2.40942}]$	$0 < x/c < 1$
f_3	$\sin^2[\pi(1 - x/c)]$	$0 < x/c < 1$
g_1	$\cos^2[(\pi/2)(y - y_{root})/(y_{engine} - y_{root})]$	$0 < y < y_{engine}$
	0	$y > y_{engine}$
g_2	$\sin^2[(\pi/2)(y - y_{root})/(y_{engine} - y_{root})]$	$0 < y < y_{engine}$
	$\cos^2[(\pi/2)(y - y_{engine})/(y_2 - y_{engine})]$	$y_{engine} < y < y_2$
	0	$y > y_2$

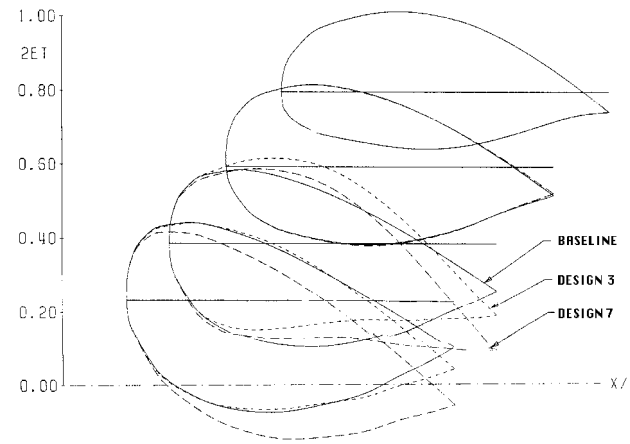

Fig. 10 Geometry evolution for business jet optimization exercise.

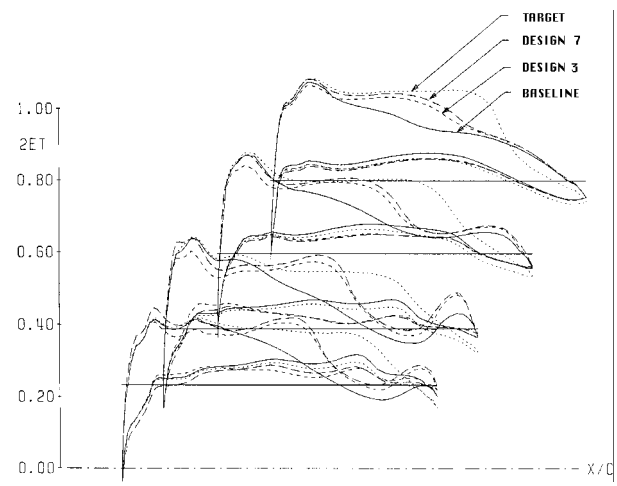

Fig. 11 Pressure evolution for business jet optimization exercise.

circumferential movement of the wing/body intersection at that $\boldsymbol{x}$ location:

$$\Delta\theta[x, \theta(z)] = \Delta\theta_{w/b}(x)g[\theta(z)] \tag{37}$$

The linear function in θ is constructed such that it tends to zero at the symmetry plane below and the circumferential coordinate corresponding to the lower limit of the strut that is a constrained obstacle. The chain rule is applied to find grid sensitivities associated with this grid movement. Ideally, a CAD tool should be used to find the new surfaces and intersection curves, along with a scheme to automatically relocate the grid points on these surfaces. This process should be readily differentiable, which would retain the capability of calculating exact grid sensitivities. In the interest of expediency, this task has been left as future work.

Sensitivities were calculated using the adjoint method, and optimization proceeded using the BFGS method. Note that golden section line searches were used for the second and third design iterations to ensure adequate reduction in the cost function and hence global convergence. This is a common tactic for quasi-Newton optimization strategies. Convergence to the minimum was found after seven iterations. Also, regridding was performed for the modified geometry after four design iterations because of excessive stretch-

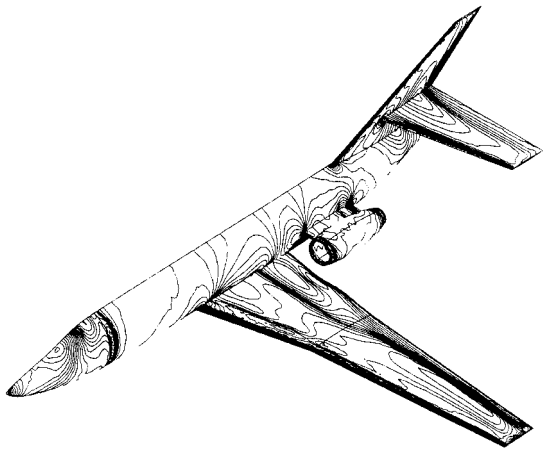

Fig. 12 Surface pressure contours for business jet at $M = 0.85$ and $\alpha = 2$ deg after seven design iterations; pressure from 0.3398 to 1.6770.

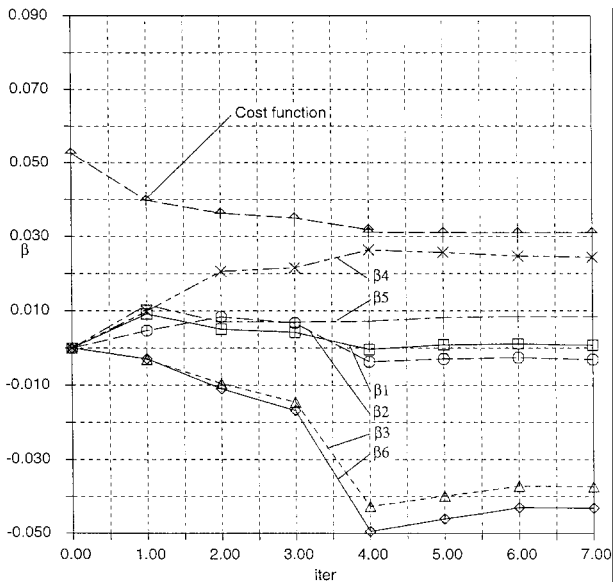

Fig. 13 Cost function and design variable evolution for business jet optimization exercise.

ing of elements at the wing/body intersection. Surface geometry and pressure evolutions are shown in Figs. 10 and 11. The final pressure distribution is shown in Fig. 12. Cost function and design variable evolutions are plotted in Fig. 13. It can be seen that much of the lift found for the "clean wing" case has been recovered. As expected, some of this has been achieved by moving the wing away from the strut-nacelle assembly whose blockage is the source of the high-pressure region. Unexpectedly, more leverage on reducing the cost function appears to have been found by cambering the wing and hence increasing the lift. At the minimum, the cost function has decreased by a factor of 43%.

Various partitioning strategies of the domain were used in this exercise, with the number of domains used varying from 6 to 20.

IV. Conclusion

In Ref. 5, the authors demonstrated, for the discrete two- and three-dimensional Euler equations, the capability to accurately calculate "exact" sensitivities. Per-iteration times for the sensitivity analysis were found to be a factor of 2 lower than those for the analysis in two dimensions. In three dimensions, the corresponding per-iteration times were found to be a factor of slightly less than 8 higher. However, when the $\partial \boldsymbol{R}/\partial \boldsymbol{U}$ matrix is stored, the increase in cost is reduced to a factor of about 2. The extension of the three-dimensional capability to practical optimization of complete aircraft configurations by parallelization of the scheme has been presented and found to be accurate and efficient. Furthermore, use of the adjoint method practically eliminates the dependence of the cost function gradient calculation on the number of design variables. However, the number of iterations required to reach a minimum for optimization exercises based on the BFGS strategy will tend to depend on N_{des} because a good estimate of the Hessian H is not typically found until after N_{des} iterations.

Acknowledgments

The authors thank the Flight Test Information Systems Branch of the NASA Dryden Research Center (under Grant NAG4-100) and the Aerothermodynamics Branch of the NASA Langley Research Center (under Grant NAG-1-1587) for partially supporting this research.

References

[1]"Computational Methods for Aerodynamic Design (Inverse) and Optimization," AGARD Rept. 463, May 1989.

[2]"Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications," AGARD Rept. 780, Nov. 1990.

[3]Arminjon, P., and Dervieux, A., "Construction of TVD-Like Artificial Viscosities on 2-Dimensional Arbitrary FEM Grids," *Journal of Computational Physics*, Vol. 106, May 1993, pp. 176–198.

[4]Elliott, J., "Aerodynamic Optimization Based on the Euler and Navier–Stokes Equations Using Unstructured Grids," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA (to be published).

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

[6]Giles, M., "An Energy Stability Analysis of Multi-Step Methods on Unstructured Meshes," Computational Fluid Dynamics Lab., TR 87-1, Massachusetts Inst. of Technology, Cambridge, MA, March 1987.

[7]Gill, P. E., Murray, W., and Wright, M. H. (eds.), *Practical Optimization*, Academic, London, 1981, pp. 116–125.

[8]Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, March 1983, pp. 357–393.

[9]Iollo, A., Kuruvila, G., and Taasan, S., "Pseudo-Time Method for Optimal Shape Design Using the Euler Equations," ICASE, TR 95-59, Aug. 1995.

[10]Iollo, A., and Salas, M. D., "Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks," ICASE, TR 95-20, March 1995.

[11]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, May 1988, pp. 233–260.

[12]Jameson, A., "Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows," AIAA Paper 93-3359, July 1993.

[13]Jameson, A., "Optimum Aerodynamic Design via Boundary Control," Dept. of Mechanical and Aerospace Engineering, MAE Rept. 1996, Princeton Univ., Princeton, NJ, April 1994.

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

[15]Newman, J., Taylor, A., and Burgreen, G., "An Unstructured Grid Approach to Sensitivity Analysis and Shape Optimization Using the Euler Equations," AIAA Paper 95-1646, June 1995.

[16]Peraire, J., Morgan, K., Vahdati, M., and Peiro, J., "The Construction and Behaviour of Some Unstructured Grid Algorithms for Compressible Flows," *Proceedings of the ICFD Conference on Numerical Methods for Fluid Dynamics*, Clarendon, Oxford, England, UK, 1992.

[17]Peraire, J., Peiro, J., and Morgan, K., "Finite Element Multigrid Solution of Euler Flows Past Installed Aeroengines," *Computational Mechanics*, Vol. 11, 1993, pp. 433–451.

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

[19]Reuther, J., and Jameson, A., "Supersonic Wing and Wing-Body Shape Optimization Using an Adjoint Formulation," *CFD for Design and Optimization: ASME International Mechanical Engineering Congress and Exposition*, American Society of Mechanical Engineers, New York, 1995.

[20]Roe, P., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, July 1981, pp. 357–372.

[21]Shubin, G., and Frank, P., "A Comparison of Two Closely-Related Approaches to Aerodynamic Design Optimization," 3rd International Conf. on Inverse Design Concepts and Optimization in Engineering Sciences, Washington, DC, Oct. 1991, pp. 67–78.

[22]Sueby, P. K., "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," *SIAM Journal on Numerical Analysis*, Vol. 21, Oct. 1984, pp. 995–1011.

J. Kallinderis
Associate Editor