PRACTICAL 3D AERODYNAMIC DESIGN AND OPTIMIZATION USING UNSTRUCTURED MESHES *

Jonathan Elliott† and Jaime Peraire‡
Computational Aerospace Sciences Laboratory
MIT Department of Aeronautics and Astronautics

Abstract

A methodology for performing optimization on 3D unstructured grids based on the Euler equations is presented. The same, low-memory-cost explicit relaxation algorithm is used to resolve the discrete equations which govern the flow, linearized direct and adjoint problems. The analysis scheme is a high resolution Local-Extremum-Diminishing (LED) scheme which 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 pressure design of a transonic wing/body configuration using a target pressure found by analyzing the geometry for known design variable deflections. The second exercise is the inverse pressure design of a business jet configuration consisting of wing, body, strut, nacelle, horizontal fin and vertical fin. The target in this case is provided by an analysis of the configuration with no strut-nacelle.

1 Introduction

Advances in both computer technology and algorithms have allowed 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 commercial 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].

*AIAA-96-4170: presented at 6th AIAA/NASA/USAF Multidisciplinary Analysis & Optimization Symposium, Seattle, Sept 1996
†Graduate Research Assistant, Member AIAA
‡Associate Professor

Copyright ©1996 American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

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. Full potential solvers discretized on Cartesian grids with adaptive embedded meshing capabilities (such as TRANAIR) resolve flows around complex geometries efficiently. Their capabilities have been extended to modelling viscous flows through the use of viscous/inviscid coupling. However for flows with significant separation, finding a converged solution becomes difficult, while for flows with strong shocks, the potential flow assumption introduces significant error. Unstructured mesh methods using tetrahedra offer the ability to model flows around complex geometries and can incorporate mesh adaptive procedures. Weakness associated with resolving boundary layers and viscous wakes are 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.

For the HSCT, supersonic aircraft in general and for subsonic transports (especially in light of the recent trend towards close-mounted nacelles), engine integration is believed to be no longer a task that can be economically performed after the clean wing performance has been optimized. This may be the step that determines the economic viability of the vehicle. Also, optimization (possibly the multidisciplinary optimization) of an aircraft high lift system, is a task that ought to be performed early in the design process, not too far behind and preferably in concert with, detailed cruise design. Both of these tasks have inherently complex geometric domains associated with them 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.

Hence a capability to perform optimization using unstructured flow solvers is perceived to be an important one. We present herein a capability for performing practical 3D optimization on unstructured grids with *no limits on the number of design variables*. This is done using the discrete adjoint method [11, 10, 9, 8, 20] 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 2D and 3D has recently been performed using the analytic adjoint approach [11, 17, 18], in which flow and adjoint equations are discretized separately. The same relaxation algorithm was used to solve flow and adjoint equations. In [11] an alternating direction method with upwind biasing and multigrid was used to drive the residuals to zero. In [17] an explicit multistage relaxation scheme was used in a scheme incorporating scalar dissipation with pressure switching to allow shock capturing.

Beux/Dervieux [4] generated sensitivities and demonstrated optimization for 2D unstructured grids using a 1st order upwind scheme with Van Leer flux vector splitting. They generated analytic gradients of the cost function using the adjoint method. The equation resulting from the separate discretization of the adjoint variables was solved using an implicit linearized time-marching scheme. Problems were reported with extending this to second order accuracy.

Newman, Taylor and Burgreen on the other hand were able to generate 2D [12], and recently 3D [13], unstructured sensitivities for a 2nd order scheme using Van Leer flux vector splitting. This was done using a similar incremental iterative implicit algorithm to the one used in [4]. They used the discrete sensitivity analysis approach in which differentiation of the discrete flow analysis scheme is performed directly. They were able to extend to transonic flow and to 2nd order by using the 1st 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 3D unstructured sensitivity capability based on the Euler equations was also reported in [5] by the 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 herein 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 an identical relaxation scheme to the flow analysis scheme. The flow variables are discretized on 2D and 3D unstructured grids. LED properties are obtained by introducing just enough dissipation to upwind each characteristic component through the use of a Roe-averaged flux jacobian. Minmod-limited antidiffusion is added to improve the accuracy of the scheme to 2nd 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 2 presents the details of the flow and sensitivity analysis algorithms. Section 3 presents applications of these algorithms to sample optimization exercises. Validation of the sensitivities produced by the algorithm has been presented in [5] and we will focus herein on the the details of the optimization exercises. We summarize the capabilities developed and indicate future research directions in Section 4. It should also be noted that the algorithm extends trivially to optimization of other more general cost functions such as drag.

2 Flow and Sensitivity Analysis Algorithms

Flow Analysis

For illustration purposes, the algorithm is described based on the 2D algorithm. The 3D algorithm is a direct extension thereof. The integral form of the Euler equations reads:-

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t} dxdy = -\int_\Omega \nabla \cdot \vec{\mathbf{F}} dxdy = -\oint_\Gamma \vec{\mathbf{F}} \cdot \hat{n} ds \qquad (1)$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \vec{\mathbf{F}} = \left(\begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(p+\rho E) \end{bmatrix}, \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(p+\rho E) \end{bmatrix} \right) \qquad (2)$$

and Ω is a closed control volume with boundary, Γ. The discretization of $\mathbf{U}$ and $\vec{\mathbf{F}}$ 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 Forward Euler. A multistage Runge-Kutta was used in the reported computations. After discretization, the following set of equations results:-

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \sigma (\frac{\Delta t}{A})_i \mathbf{R}_i(\mathbf{U}^n) \qquad (3)$$

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 CFL number. The residual at a point can be calculated as a sum over all the edges that contain that point [16]:-

$$\mathbf{R}_i = \sum_{ik=1}^{K} [\mathbf{R}]_{ik} \qquad (4)$$

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

$$[\mathbf{R}]_{ik} = -\frac{\vec{\mathbf{F}}_i + \vec{\mathbf{F}}_k}{2} \cdot \vec{S}_{ik} \qquad (5)$$

where $\vec{S}_{ik}$ is the area vector associated with edge ik and calculated as follows (see Figure 1):-

$$\vec{S}_{ik} = (\frac{y_{k+1} - y_{k-1}}{3}, \frac{x_{k-1} - x_{k+1}}{3}) \qquad (6)$$

The residual contribution for a wall boundary edge is:-

$$[\mathbf{R}_i^b]_{il} = -(\begin{bmatrix} 0 \\ p \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ p \\ 0 \end{bmatrix}) \cdot \vec{S}_b - \frac{\vec{\mathbf{F}}_i + \vec{\mathbf{F}}_l}{2} \cdot \vec{S}_{l,l+1} \quad (7)$$

where the second term ensures closure of the control volumes (see Figure 1) and,

$$\vec{S}_{l,l+1} = (\frac{y_{l+1} - y_l}{3}, \frac{x_l - x_{l+1}}{3})$$

$$\vec{S}_b = (\frac{y_l - y_i}{3}, \frac{x_i - x_l}{3}) \quad (8)$$

For a farfield boundary edge the contribution to the residual becomes:-

$$[\mathbf{R}_i^b]_{il} = -\frac{\vec{\mathbf{F}}_i + \vec{\mathbf{F}}_i^b}{2} \cdot \vec{S}_b - \frac{\vec{\mathbf{F}}_i + \vec{\mathbf{F}}_l}{2} \cdot \vec{S}_{l,l+1} \quad (9)$$

where the superscript b indicates that the flux includes a 1st order diffusive contribution that effects upwinding using Roe-averaging, as described below, between the given node and the freestream state vector U_∞:-

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

These expressions for the fluxes at the boundaries ensure that correct boundary conditions are imposed and that we have $\mathbf{R}(\mathbf{U}) = 0$ 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. (Note that projection of the wall velocities ensures that in general the wall node residuals will not be zero for the normal momentum equations. In this approach, the normal momentum equation has to be discarded since knowledge of the wall velocity direction eliminates one equation.)

To stabilize the scheme, a dissipative term, premultiplied by a matrix coeffiecient, is added to the above equations for interior and farfield boundary fluxes.

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

where $|\mathbf{A}(\mathbf{U}_i, \mathbf{U}_k)|$ is the standard Roe matrix [19] and $S = |\vec{S}_{ik}|$. This ensures positivity of the coefficients multiplying increments of characteristic variables, $\Delta\mathbf{W}_{ik} = \mathbf{P}^{-1}\Delta\mathbf{U}_{ik}$, and causes the scheme to bear a strong resemblance to a first order upwind algorithm following MUSCL concepts [21, 15]. 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:-

$$\mathbf{D}_{ik} = \frac{S}{2}\mathbf{P}|\mathbf{\Lambda}|(\mathbf{P}^{-1}\Delta\mathbf{U}_{ik} - L(\mathbf{P}^{-1}\Delta\mathbf{U}^-, \mathbf{P}^{-1}\Delta\mathbf{U}^+)) \quad (11)$$

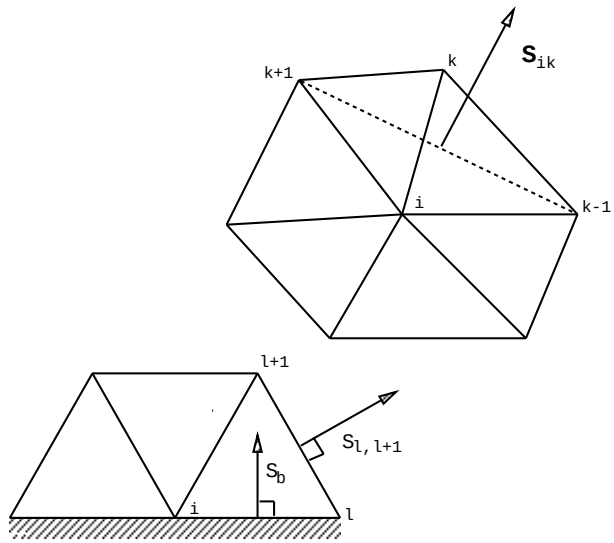

Figure 1: Area vectors for boundary and interior edges

where, for example,

$$L(u, v) = minmod(u, v) \quad (12)$$

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

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

where the nodes in the adjacent element have been labelled i,r,s and $\epsilon_{ir}, \epsilon_{is}$ 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, the reader is referred to [16, 15, 6].

Sensitivity Analysis

Gradient-based optimization methods require the gradient, or sensitivity, of cost function, I, with respect to design variables, β. Sensitivities can be calculated using either the direct or the adjoint method. In either case, the implicit function theorem is used which results from direct differentiation of the steady state discrete residual equation,

$$\mathbf{R}(\mathbf{U}, \beta) = 0 \quad (14)$$

with respect to the design variables, producing,

$$\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial\beta} = -\frac{\partial\mathbf{R}}{\partial\beta} \quad (15)$$

Hence we can calculate the cost function sensitivity as:-

$$\frac{dI}{d\beta} = \frac{\partial I}{\partial\beta} + \frac{\partial I}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial\beta} \quad (16)$$

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

$$(\frac{\partial\mathbf{R}}{\partial\mathbf{U}})^T\psi = (\frac{\partial I}{\partial\mathbf{U}})^T \quad (17)$$

The cost function sensitivities are then calculated according to

$$\frac{dI}{d\beta} = \frac{\partial I}{\partial \beta} - \frac{\partial I}{\partial \mathbf{U}}(\frac{\partial \mathbf{R}}{\partial \mathbf{U}})^{-1}(\frac{d\mathbf{R}}{d\beta}) = \frac{\partial I}{\partial \beta} - \psi^T(\frac{d\mathbf{R}}{d\beta}) \qquad (18)$$

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

$$U_{\beta_i}^{n+1} = U_{\beta_i}^n + \sigma(\frac{\Delta t}{A})_i(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\mathbf{U}_\beta + (\frac{d\mathbf{R}}{d\beta})_U) \qquad (19)$$

$$\psi_i^{n+1} = \psi_i^n + \sigma(\frac{\Delta t}{A})_i(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \psi - \frac{\partial I}{\partial \mathbf{U}}^T) \qquad (20)$$

For the direct method, equation (23) is solved for each design variable while for the adjoint method equation (25), requiring a similar computational effort, needs to be solved only once. This can represent a significant 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):

$$\Delta U_i^{n+1} \approx \Delta U_i^n + \sigma(\frac{\Delta t}{A})_i \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\Delta \mathbf{U^n} \qquad (21)$$

The stability characteristics of equations (27) and (28) are the same since transposing a matrix does not change its eigenvalues.

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

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_j}\mathbf{U}_{\beta_j} = \sum_{ik=1}^{K}[\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_j}]_{ik}\mathbf{U}_{\beta_j} \qquad (22)$$

where, for example, for an interor edge,

$$\begin{aligned}[\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_j}]_{ik}\mathbf{U}_{\beta_j} = [&-\frac{\vec{\mathbf{A}}_k\delta_{kj}}{2}\cdot\vec{S}_{ik} + \\ &\frac{\partial}{\partial \mathbf{U}_j}(S\mathbf{P}|\mathbf{\Lambda}|)(\Delta\mathbf{W}_{ik} - L(\Delta\mathbf{W}^-, \Delta\mathbf{W}^+)) + \\ &(S\mathbf{P}|\mathbf{\Lambda}|)\frac{\partial}{\partial \mathbf{U}_j}(\Delta\mathbf{W}_{ik} - L(\Delta\mathbf{W}^-, \Delta\mathbf{W})^+))]\mathbf{U}_{\beta_j}\end{aligned}$$

where $\vec{\mathbf{A}}_k = (\mathbf{A}_k, \mathbf{B}_k)$ is the vector of flux Jacobians.

For the adjoint problem the transposed Jacobian is required.

$$(\frac{\partial \mathbf{R}_j}{\partial \mathbf{U}_i})^T\psi_i = \sum_{ik=1}^{K}[\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_j}]_{ik}\psi_i \qquad (23)$$

The chain rule is applied to the forcing term derivatives. For example, we have:-

$$\frac{\partial \mathbf{R}}{\partial \beta} = \frac{\partial \mathbf{R}}{\partial \mathbf{X}}\frac{d\mathbf{X}}{d\beta} \qquad (24)$$

and this 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

$$\vec{\delta}_i^{n+1} = \frac{\sum k_{ij}\vec{\delta}_j^n}{\sum k_{ij}} \qquad (25)$$

where $k_{ij} = |\Delta\vec{x}_{ij}|^{-1}$ and

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \delta^{n+1} \qquad (26)$$

Therefore the grid sensivitity is simply

$$\frac{d\mathbf{X}}{d\beta} = \frac{\partial \delta^N}{\partial \beta} = \frac{\partial \delta^N}{\partial \delta^{N-1}}\frac{\partial \delta^{N-1}}{\partial \delta^{N-2}}\cdots\frac{\partial \delta^1}{\partial \delta^0}\frac{\partial \delta^0}{\partial \beta} \qquad (27)$$

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

Much discussion of boundary conditions for the adjoint equation has taken place in the literature [11, 10, 9, 8]. We take the approach that since we have implicitly included the boundary conditions in the expression for the flow equation residual, that 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 parallelised, 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 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 inter–domain boundary nodes i.e. the nodes requiring communication. The second group contains interior nodes only. Typically, a loop over all the edges containing boundary nodes is performed first. The values accumulated at the boundary nodes are then broadcast to the neighbouring domains. The interior edges are considered next and the information from the neighbouring domains is received. In this way, communication and computation take place concurrently. The parallel efficiency achieved from following

this strategy with 12 processors on the IBM SP2 at MIT is higher that 95% for the simulation of practical problems. With this implementation, the flow analysis code requires 0.32 microseconds per time step per mesh point while the adjoint (without the savings found by storing $\partial\mathbf{R}/\partial\mathbf{U}$, as discussed next) requires 2.42 microseconds.

In [5], it was found that the edge contributions to $\partial\mathbf{R}/\partial\mathbf{U}$ matrix could be stored easily for the 2D case. Eliminating the cost of forming this matrix every iteration lowers the per-iteration cost significantly. It was found that in 2D the sensitivity calculation took approximately half that of the flow analysis while it was a factor of eight larger in 3D 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 3D cases it was found that saving the $\partial\mathbf{R}/\partial\mathbf{U}$ matrix reduced the per-iteration cost by a factor of about four. Furthermore, the parallel implementation may allow storing of $\partial\mathbf{R}/\partial\mathbf{U}$ since only those values corresponding to the local domain and interface regions need to be stored.

3 Results

Wing/body optimization This case was based on a seed geometry which 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, $\Lambda_{LE} = 25^\circ$, aspect ratio, $A_r = 9$, taper ratio, $\lambda = 5$. Freestream conditions were chosen to be $M_\infty = 0.8$ and $\alpha = 1.25^\circ$. Intermediate results for this exercise were reported in [5]. Here the exercise is carried to completion.

The baseline grid and pressure contours are shown in Figures 2 and 3. The mesh contains 65,000 nodes and 350,000 elements. In spite of the large size of this grid, this optimization case was performed using the serial algorithm. Design variables were chosen to be products of linear functions in y in the spanwise direction and NACA 4-series mean-line, $f_1(x/c)$ (with maximum at 40% chord) and thickness $f_2(x/c)$ functions in the chordwise direction. Two linear spanwise hat functions $g_1(y), g_2(y)$ were chosen with peaks at $\eta = 2y/b = .55$ and $\eta = 1.0$, respectively. Hence we have a total of four design variables, with spatial variations given by $h_{ij}(x, y) = f_i(x/c)g_j(y)$. The spanwise and chordwise variations are shown in Figure 4.

A target pressure distribution was generated by perturbing the camber design variables by an amount corresponding to vertical movement of the mean line of $\Delta\frac{y_c}{c}(\frac{x}{c} = .4) = .02$ (at the spanwise design variable maxima) and to a change in maximum thickness of $-.04c$. Hence the geometry of the target corresponds closely to a linear loft into the baseline planform of NACA2408 sections from $\eta = 55\%$ to $\eta = 100\%$ and a NACA0012 section at side-of-body. It should be realized that 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 p_{targ} in the expression for the cost function, cost function sensitivities were calculated by the adjoint method. Modal design optimization was performed based on the sensitivities found using the adjoint method and the BFGS method [7]. Resulting geometry and surface pressure evolutions are shown in Figures 5 and 6 while the evolution of design variables and cost function is shown in Figure 7. It should be noted that after the eighth iteration, the BFGS optimization process was 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 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 [5]) provides confidence that the sensitivities are being accurately calculated by the current scheme.

Business jet optimization The second optimization example presented herein 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 $\alpha = 2^\circ$. Engine exhaust total pressure increases are not modelled - flow through conditions are assumed. The baseline grid and pressure contours are shown in Figures 12 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 are removed at the same conditions. This pressure distribution is shown in Figure 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. Chordwise variations for the design variables were chosen to be given by a shear function $f_1(x/c) = x/c$, and two Hicks Henne functions given by:-

$$f_2(x) = \sin(\pi(\frac{x}{c})^{2.40942})$$

$$f_3(x) = \sin^2(\pi(1 - \frac{x}{c}))$$

The spanwise functions were chosen to have the following piecewise variations

$$g_j(x) = \cos^2(\pi(\frac{y - y_{peak_j}}{y_{peak_j} - y_{peak_{j-1}}}))$$

$$g_j(x) = \cos^2(\pi(\frac{y - y_{peak_j}}{y_{peak_{j+1}} - y_{peak_j}}))$$

where y_{peak_j} represents the spanwise location of the maximum for that function. The two spanwise functions reported herein have maxima at the engine centerline and at the maximum y value of the wing/body intersection curve. It should be noted that this definition ensures slope continuity in the final geometry. 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 to avoid highly skewed elements or even elements with negative volume. The way this was done was to take 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, that is linear in θ (the angular coordinate for the circular section associated with the local value of x) and the circumferential movement of the wing/body intersection at that x-location:-

$$\Delta\theta(x, \theta(z)) = \Delta\theta_{w/b}(x) g(\theta(z)) \tag{28}$$

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 which is a constrained obstacle. The chain rule is applied quite easily 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. It should be noted that golden section line searches were used for the 2nd and 3rd design iterations to ensure adequate reduction in the cost function and hence global convergence. This is a typical tactic for quasi-Newton optimization strategies. Convergence to the minimum is found after 7 iterations. Also, it should be noted that regridding was performed for the geometry found after 4 design iterations due to excessive stretching of elements at the wing-body intersection. Surface geometry and pressure evolutions are shown in Figures 10 and 11. The final pressure distribution is shown in Figure 12. Cost function and design variable evolutions are plotted in Figure 14. 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.

4 Conclusion

In [5], the authors demonstrated for the 2D and 3D Euler equations, the capability to accurately calculate "exact" sensitivities for the discrete flow equations. Per-iteration times for the sensitivity analysis were found to be a factor of 2 lower than those for the analysis in 2D. In 3D, the corresponding per-iteration times were found to be a factor of slightly less than 8 *higher*. However, when the $\partial\mathbf{R}/\partial\mathbf{U}$ matrix is stored, the *increase* in cost is reduced to a factor of about 2. The extension of the 3D 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 reached a minimum for optimization exercises based on the BFGS strategy will tend to depend on this quantity.

Future work will be focused on development of a capability to perform optimization based on the thin layer Navier-Stokes equations and more general geometry parameterization procedures.

Acknowledgements

Thanks are in order to Professor Mark Drela for invaluable assistance at various stages of this project. The advice of Brian Nishida and Tom Sorensen has been of immense help. The authors would also like to 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 supporting this research.

References

[1] AGARD. Computational Methods for Aerodynamic Design (Inverse) and Optimization. Report No. 463, 1989.

[2] AGARD. Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications. Report No. 780, 1990.

[3] P. Arminjon and A. Dervieux. Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. INRIA Report 1111, 1989.

[4] F. Beux and A. Dervieux. Exact-gradient shape optimization of a 2-D Euler flow. *Finite Elements in Analysis and Design*, 12:281–302, 1992.

[5] J. Elliott and J. Peraire. Aerodynamic Design using Unstructured Meshes. *AIAA*, 96-1941, 1996.

[6] J. K. Elliott. *Aerodynamic Optimization based on the Euler and Navier-Stokes equations using Unstructured Grids.* PhD thesis, Dept. of Aeronautics and Astronautics, MIT, 1997. To be published.

[7] P. E. Gill, W. Murray, and M. H. Wright, editors. *Practical Optimization.* Academic Press, 1981.

[8] A. Iollo, G. Kuruvila, and S. Taasan. Pseudo-Time Method for Optimal Shape Design using the Euler Equations. Technical Report 95-59, ICASE, 1995.

[9] A. Iollo and M. D. Salas. Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks. Technical Report 95-28, ICASE, 1995.

[10] A. Jameson. Aerodynamic Design via Control Theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[11] A. Jameson. Optimum Aerodynamic Design via Boundary Control. Princeton University, MAE Report 1996, 1994.

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

[13] J. Newman and A. Taylor. Three-dimensional Aerodynamic Shape Sensitivity Analysis and Design Optimization Using the Euler Equations on Unstructured Grids. *AIAA*, 96-2464, 1996.

[14] J. Peiro, J. Peraire, O. Morgan, K.and Hassan, and N. Birch. The Numerical Simulation of flow about Installed Aero Engine Nacelles using a Finite Element Euler Solver on Unstructured Meshes. *The Aeronautical Journal of the Royal Aeronautical Society*, pages 224–232, 1992.

[15] J. Peraire, K. Morgan, M. Vahdati, and J. Peiro. The Construction and Behaviour of Some Unstructured Grid Algorithms for Compressible Flows. In *Proc. of the ICFD Conference on Numerical Methods for Fluid Dynamics*, 1992.

[16] J. Peraire, J. Peiro, and K. Morgan. Finite Element Multigrid Solution of Euler Flows past Installed Aeroengines. *Computational Mechanics*, 11:433–451, 1993.

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

[18] J. Reuther and A. Jameson. Supersonic Wing and Wing-Body Shape Optimization using an Adjoint Formulation. In *CFD for Design and Optimization: ASME International Mech Eng Congress and Exposition*, November 1995.

[19] P. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[20] G. Shubin and P. Frank. A Comparison of Two Closely-Related Approaches to Aerodynamic Design Optimization. In *3rd International Conference on Inverse Design Concepts and Optimization in Engineering Sciences*, October 1991.

[21] B. van Leer. Towards the ultimate conservative scheme: a second order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

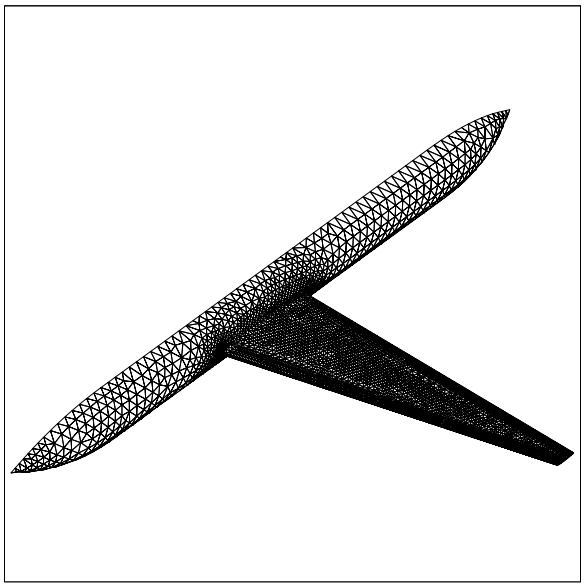

Figure 2: Surface grid for baseline 3D wing/body calculation

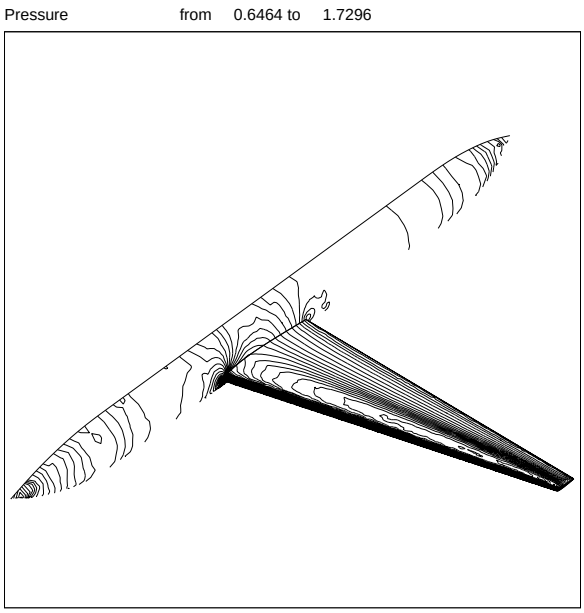

Figure 3: Surface pressure contours for 3D wing/body at M=0.8, aoa=1.25deg

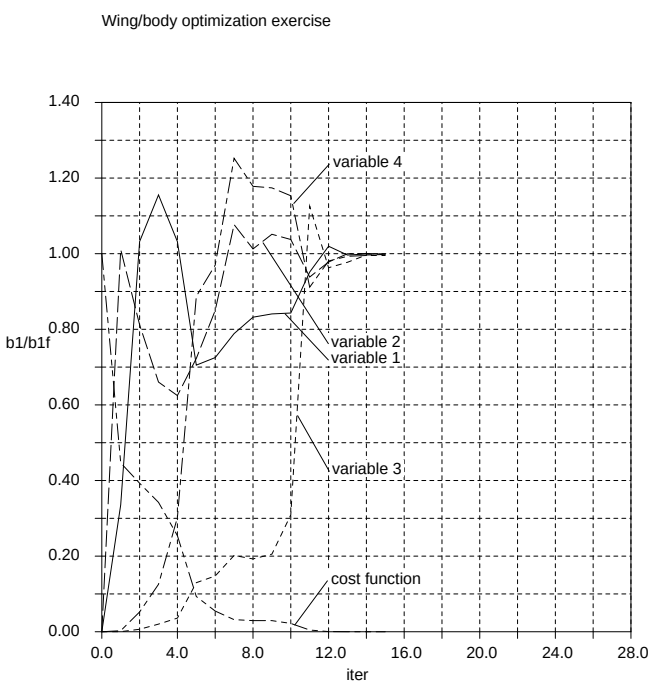

Figure 4: Cost function and design variable evolution for wing/body optimization exercise

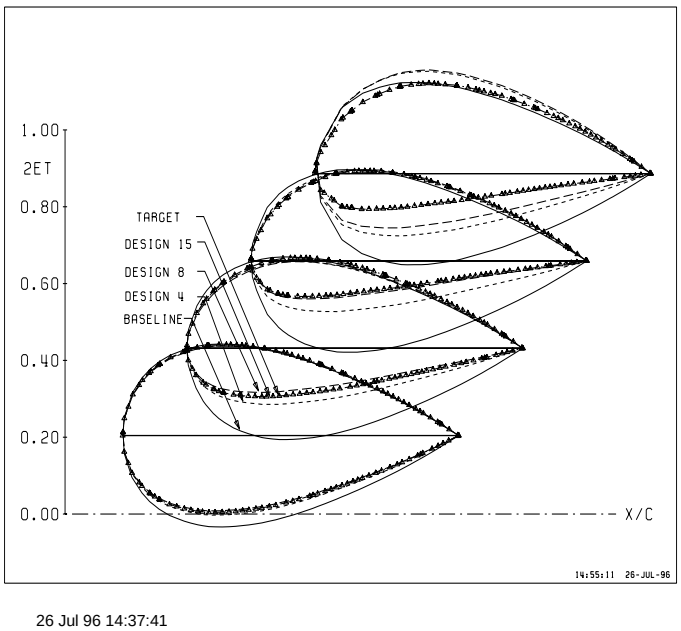

Figure 6: Geometry evolution for wing/body optimization exercise

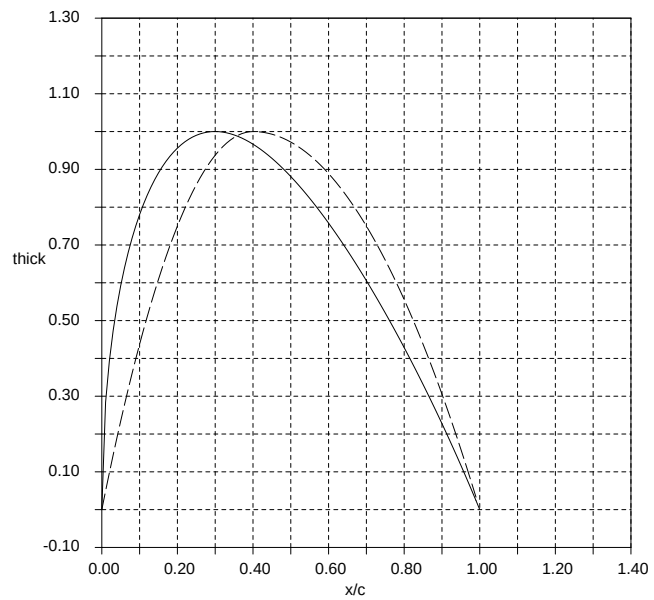

Figure 5: Chordwise components of design variables for Wing/body optimization exercise

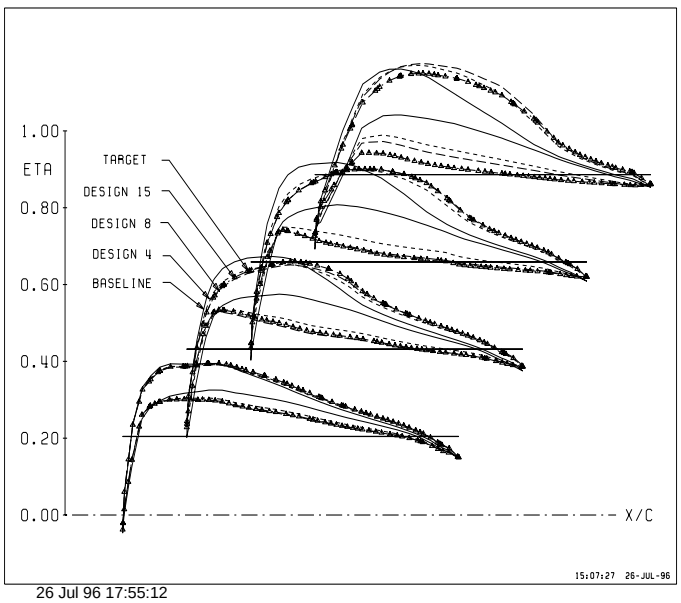

Figure 7: Pressure evolution for wing/body optimization exercise

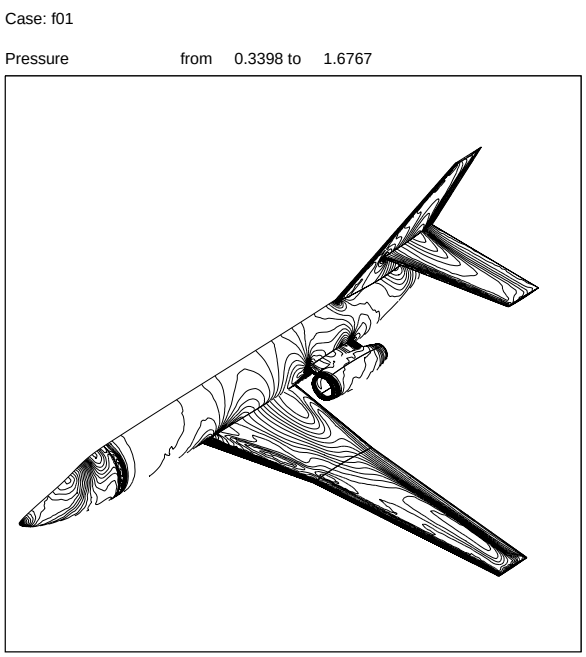

Figure 8: Baseline surface pressure contours for business jet at M=0.85, aoa=2deg

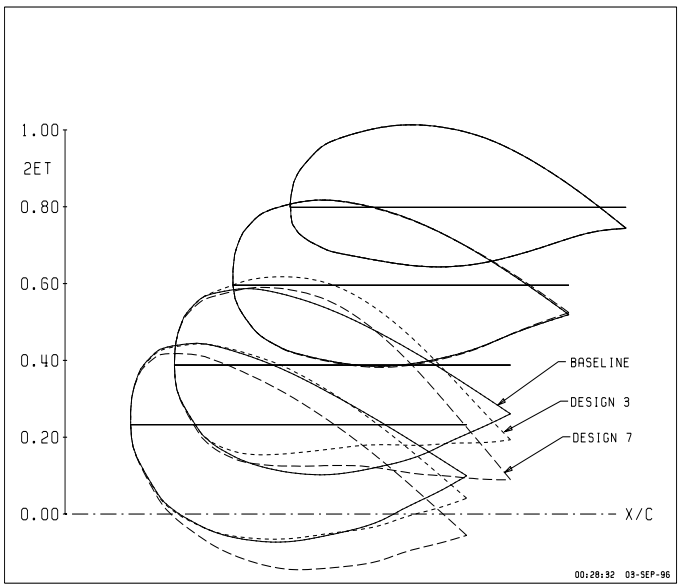

Figure 10: Geometry evolution for business jet optimization exercise

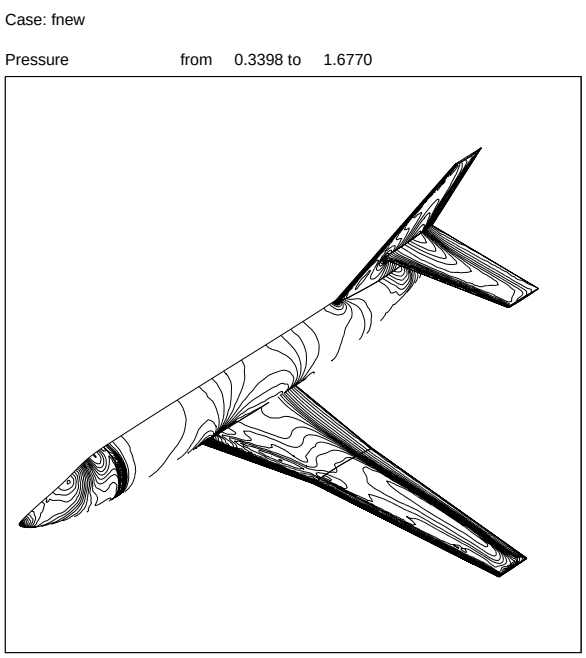

Figure 9: Target surface pressure contours for business jet at M=0.85, aoa=2deg

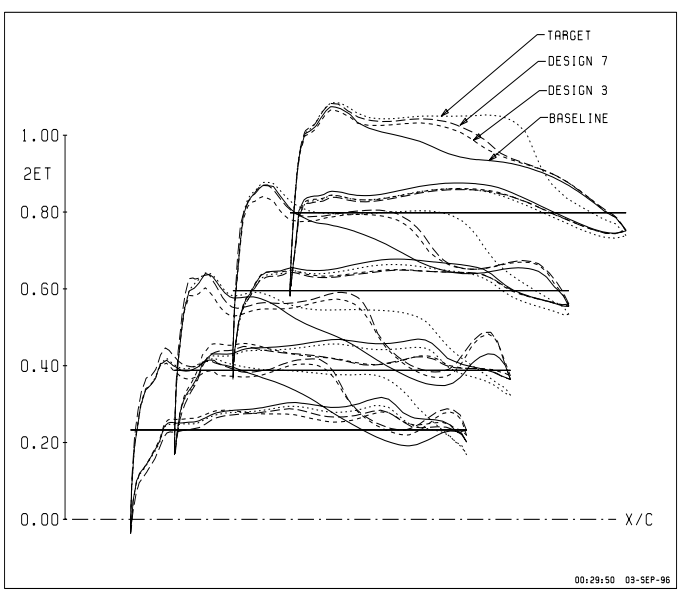

Figure 11: Pressure evolution for business jet optimization exercise

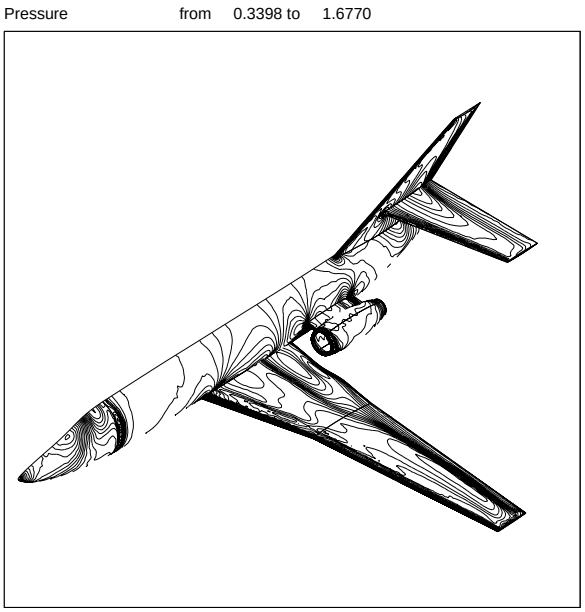

Figure 12: Surface pressure contours for business jet at M=0.85, aoa=2deg after 7 design iterations

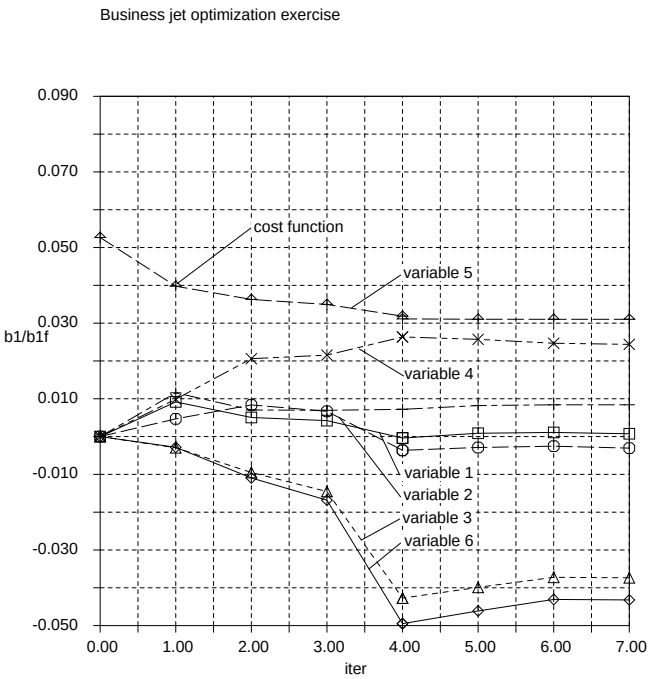

Figure 14: Cost function and design variable evolution for business jet optimization exercise

Figure 13: Surface grid for baseline business jet calculation