AERODYNAMIC DESIGN 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 2D and 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 schemes, for both 2D and 3D, are high resolution Local-Extremum-Diminishing (LED) schemes and use Roe decomposition for the dissipative fluxes. The local time-stepping relaxation scheme is based on a multidimensional equivalent of a TVD CFL-like condition guaranteeing convergence of flow and sensitivity computations to machine accuracy. Mesh movement is performed in such a way that optimization of arbitrary geometries is allowed. Sensitivities based on direct and adjoint methods are validated and sample optimizations are performed: the inverse pressure design of a multielement airfoil in high-lift mode, an infinite-span straight transonic wing and a transonic wing/body configuration. It is shown that, due to its near-*elimination* of CPU cost dependence on the number of design variables, the adjoint method is preferred over the direct and finite difference methods for practical single-discipline aerodynamic optimization.

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].

*Graduate Research Assistant, Member AIAA
†Associate Professor

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 [12, 11, 10, 9, 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 [12, 17, 18], in which flow and adjoint equations are discretized separately. The same relaxation algorithm was used to solve flow and adjoint equations. In [12] 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 [13] on the other hand were able to generate 2D 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.

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.

In this paper, we demonstrate the validity of our sensitivity generation algorithm by comparing flow variable sensitivities as calculated by both direct and adjoint methods with corresponding finite difference values. We demonstrate the power of these methods by performing 2D modal inverse design for single- and multi-element airfoils and 3D modal inverse design for an infinite straight wing and and a wing/body configuration. 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

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

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:-

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

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

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

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

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

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

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.)

Figure 1: Area vectors for boundary and interior edges

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

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

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:-

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

where, for example,

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

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,

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

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].

Timestep calculation

It was found that use of a linearized Fourier stability analysis to determine the allowable timestep [7], 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 $\mathbf{U}$ whose solution convergence histories exhibited this limit-cycling behaviour can be unstable at the CFL numbers that the analyses were run at.

The problem was solved by supplementing the local timestep calculation with a check on violations of the TVD CFL-like condition. TVD conditions for the semidiscrete form of a scheme that can be expressed as

$$\frac{du_o}{dt} = \sum_{i=1}^{K} c_{oi}(u_i - u_o) \tag{14}$$

are

$$c_{oi} > 0 \quad i = 1, \cdots, K \tag{15}$$

When the forward Euler time discretization is performed, the scheme can be rewritten as

$$u_o^{n+1} = u_o^n + \Delta t \sum_{i=1}^{K} c_{oi}(u_i - u_o) = \sum_{i=1}^{K} d_{oi} u_i + \mu_o u_o \tag{16}$$

In this case, the TVD conditions become:-

$$d_{oj} = \Delta t c_{oj} > 0 \quad j = 1, \cdots, K \tag{17}$$

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

The latter is a multidimensional TVD CFL-like condition. For our scheme this takes the form

$$\mu_i = 1 - \Delta t (\sum_{ki}^{edges} (C_{ki}^{int} + C_{ki}^{upw} + C_{ki}^{ad}) + \sum_{kip}^{bou} C_{kip}^{bou}) \tag{19}$$

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

$$C_{ki}^{int} = C_{ki}^{upw} = \frac{\sigma}{A_i} S \lambda_{max} \tag{20}$$

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

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

The supplemental TVD violations 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 slow convergence and subsequent investigation revealed that using a threshhold value of $\mu_t < 0$ allowed quick convergence without limit-cycling. Figure (2) shows the convergence histories of a typical analysis scheme (a) without a check on (b) with $\mu_t = -1$. It can be seen that the check leads to dramatically improved convergence properties. Better yet, the sensitivity calculation, with an identical Δt, was found to be stable for the CFL number the analysis scheme had been run at. Indeed for all calculations performed, the asymptotic convergence rate of analysis, direct and adjoint schemes was found to be very close.

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,

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

with respect to the design variables, producing,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \beta} = -\frac{\partial \mathbf{R}}{\partial \beta} \tag{23}$$

Hence we can calculate the cost function sensitivity as:-

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

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

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

The cost function sensitivities are then calculated according to

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

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.

$$U_{\beta_i}^{n+1} = U_{\beta_i}^n + \sigma(\frac{\Delta t}{A})_i (\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \mathbf{U}_\beta + (\frac{d\mathbf{R}}{d\beta})_U) \tag{27}$$

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

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):

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

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

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

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.

$$(\frac{\partial \mathbf{R}_j}{\partial \mathbf{U}_i})^T \psi_i = \sum_{ik=1}^{K} [\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_j}]_{ik} \psi_i \tag{31}$$

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

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

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

$$\vec{\delta}_i^{\,n+1} = \frac{\sum k_{ij} \vec{\delta}_j^{\,n}}{\sum k_{ij}} \qquad (33)$$

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

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \delta^{n+1} \qquad (34)$$

Therefore the grid sensivitity is simply

$$\frac{d\mathbf{X}}{d\beta} = \frac{\partial \delta^N}{\partial \beta} = \frac{\partial \delta^N}{\partial \delta^{N-1}} \frac{\partial \delta^{N-1}}{\partial \delta^{N-2}} \cdots \frac{\partial \delta^1}{\partial \delta^0} \frac{\partial \delta^0}{\partial \beta} \qquad (35)$$

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 [12, 11, 10, 9]. 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 It was found that the edge contributions to $\partial \mathbf{R}/\partial \mathbf{U}$ matrix could be stored easily for the 2D case. Doing away with the requirement of forming this matrix every iteration lowers the per-iteration cost significantly. It was found that 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 where was not done due to the large memory requirements. However, it should be pointed out that some of the most expensive components of the $\partial \mathbf{R}/\partial \mathbf{U}$ calculation could be saved and it is felt that this will allow a good proportion of the 2D savings to be made.

3 Results

2D Multi-element optimization To demonstrate the accuracy of the sensitivities, a modal inverse design optimization exercise on a multielement airfoil was performed. The baseline geometry was a NACA4412 main element with a scaled-down NACA4415 flap placed just below and aft of the main's trailing edge with 3% overlap. The initial grid is shown in Figure 3. The baseline solution at freestream conditions of $M_\infty = 0.425$, $\alpha = 9.$ deg is shown in Figure 4. Three design variables were chosen to be flap rotation angle about a point near the flap leading edge, flap vertical deflection and a camber design variable on the main element. This camber variable is given by a vertical surface perturbation with the same x variation as the NACA 4-series meanline [22] with maximum camber at 40% of chord. To allow modal inverse design optimization, the cost function was defined as

$$I = \oint_{\Gamma_w} (p - p_{targ})^2 ds$$

The target pressure distribution, p_{targ} for the inverse design exercise was generated by perturbing all three design variables. For example, the flap was rotated 5°.

The direct-method point flow sensitivities were validated by comparing them with corresponding finite difference quantities. An example of this comparison is shown in Figure 7 where density sensitivity is plotted versus axial location for finite difference and direct sensitivity analysis. Other variables exhibited similar agreement. Correlation away from the airfoil surface was also good. Cost function sensitivities as calculated by both direct and adjoint methods - using equations (27) and (28), respectively - were validated also by comparing with finite difference. These comparisons are presented in row 1 of Table 1 for the flap rotation design variable. Differences on the order of less than 1% were usually found between adjoint and direct methods (depending on the level of convergence), while comparison of the analytic sensitivities versus finite difference produced differences on the order of 2% or so but, in this case, also depending on the size of the design variable perturbation.

Next an inverse design optimization exercise was performed. Based on the sensitivities found using the direct method, the geometry was perturbed to a new state. In this case, a Newton algorithm [5, 8] was used in which an approximation of the Hessian matrix ($\mathbf{H}$) is made to estimate the location of the minimum.

$$\mathbf{H}_{ij} \approx \oint_{ws} \frac{\partial p}{\partial \beta_i} \frac{\partial p}{\partial \beta_j} d\Gamma \qquad (36)$$

This estimate becomes increasingly accurate as the target is approached. For this case, application of the Newton algorithm results in

$$\mathbf{H} \Delta \beta = -\nabla I \qquad (37)$$

Also, it should be pointed out that an "informal" form of the Golden section line search was performed for all optimization cases discussed herein. For those design iterations where the design variable update led to an increase in cost function, due to the nonparabolic nature of the cost function variation for example, the design change was continually reduced by an amount given by the golden section factor $\Delta\beta^{i+1} = \frac{3-\sqrt{5}}{2}\Delta\beta^i$ until a decrease in the cost function was found.

The evolutions of surface pressure distribution and geometry in the course of the direct-method-based optimization process are shown in Figures 5 and 6. Figure 8 shows the variation of cost function and design variables as a function of design iteration. Convergence to the target is reached in 3 steps.

With the adjoint-method-based sensitivities, we can no longer make such a good initial estimate of the Hessian matrix as we do with the direct-method-based sensitivities. Therefore, we use the BFGS method in which an increasingly accurate estimate of H is made based on changes in the gradient vector along the optimization path. Here, (and wherever the BFGS method is used herein), the quasi-Newton scheme was started by taking one step using steepest descent $\Delta\beta = -k\nabla I$ and using an initial guess for $\mathbf{H}$ proportional to the identity matrix $\mathbf{I}$. As expected, this required more iterations (10 in this case) to reach convergence. With only three design variables, the direct method is about twice as fase as the adjoint method. However for a larger number of design variables the adjoint is expected to be much more efficient. Furthermore, for drag optimization in which the direct method loses its advantage of an accurate Hessian [5], the adjoint method is expected to be superior.

3D Infinite straight wing optimization For this case, the baseline geometry was a straight wing formed by lofting two NACA0012 sections separated by a spanwise distance of 70% chord. The computational domain is bounded by eight surfaces including two symmetry planes at the spanwise extremities of the wing. Freestream conditions are $M_\infty = 0.8$ and $\alpha = 1.25°$. Figure 11 and 12 show baseline grid and C_p contours, respectively. Two design variables were chosen to be vertical surface perturbations with the same x variation as the NACA 4-series meanline and thickness distributions with the meanline maximum located at 40% of chord.

Figure 9 shows a comparison of surface density sensitivity to the camber design variable as calculated by direct and finite difference methods (using $\Delta\beta = .0001c$). It can be seen that the agreement is quite good, with some of the noisiness in the finite difference plot being due to traversal of discontinuous portions of the limiting function distribution for some edges. Further validation of the sensitivity calculations was provided by performing the full cost function sensitivity calculation for adjoint, direct and finite difference methods. For the camber design variable, the values are printed in row 2 of Table 1. (The cost function is calculated based on a target generated by perturbing the design variables by such an amount as to reproduce the NACA1410 airfoil – or a close replica of it since thickness was applied vertically from the meanline rather than normal to it).

Next, modal design optimization was performed based on the sensitivities found using the adjoint method. For this case, the BFGS or quasi-Newton method was used. Resulting geometry and surface pressure evolutions are shown in Figures 13 and 14 while the evolution of design variables and cost function is shown in Figure 10. Convergence is found at about 6 design iterations.

3D 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, $\Lambda_{LE} = 25°$, aspect ratio, $A_r = 9$, taper ratio, $\lambda = 5$. Freestream conditions were chosen to be $M_\infty = 0.8$ and $\alpha = 1.25°$. The baseline grid and pressure contours are shown in Figures 15 and 16. The shock in this case is weaker than for the infinite wing due to its obliqueness caused by the wing sweep. Design variables were chosen to be products of linear functions in y in the spanwise direction and NACA 4-series meanline, $f_1(x/c)$ (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 $h_{ij}(x,y) = f_i(x/c)g_j(y)$.

Figure 19 shows a comparison of analytic versus finite difference sensitivity at $\eta = 66\%$ for design variable $h_{11}(x,y)$. Similar agreement is found at other span stations and for other state vector variable sensitivities.

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 twice as much change in camber and thickness as was used for the straight infinite wing case and is therefore a more challenging target to design to. Using the resulting pressure distribution as p_{targ} in the expression for the cost function, cost function sensitivities were calculated by direct, adjoint and finite difference methods. Row 3 in Table 1 shows the resulting comparison.

Finally, modal design optimization was performed based on the sensitivities found using the adjoint method and the BFGS method. Resulting geometry and surface pressure evolutions are shown in Figures 17 and 18 while the evolution of design variables and cost function is shown in Figure 20. Due to time constraints, this optimization exercise has not been carried to completion. Nonetheless, it can be seen that the "design space march" has entered somewhat level territory, if not a local minimum. While the two camber design variables and the inboard thickness variable are close to the target, the outboard thickness variable is only 20% of the way to the target value. Part of the reason for the flatness here is that the outboard thickness design variable (as well as the outboard camber) have far less impact on the cost function than their corresponding inboard counterparts. This is because of the reduced area of the wing in the outboard half due to taper. A more effective approach would be to redefine the cost function or the design variables such that there is a more equitable impact on the cost function by each design variable.

Another wing/body optimization exercise was performed with 9 design variables. In this case, the three linear spanwise functions $g_1(y), g_2(y), g_3(y)$ had peaks at $\eta = 36\%$, $\eta = 64\%$ and $\eta = 100\%$, respectively. Chordwise variations were provide by a linear shear variable $f_1(x/c) = x/c$, and two Hicks-Henne type design variables,

$$f_2(x) = \sin(\pi(\frac{x}{c})^{2.40942})$$
$$f_3(x) = \sin^2(\pi(1 - \frac{x}{c}))$$

shown in Figure (22). The target was generated by perturbing all 9 design variables. Freestream conditions in this case are $M_\infty = 0.85$ and $\alpha = 1.25°$. Once again, time constraints precluded carrying the design march to convergence. However it can be seen that significant progress towards the minimum has been made, as can be deduced by the low level of the cost function at the final iteration. Furthermore, this example shows how powerful the adjoint method is since at each iteration only one additional inversion of $\partial\mathbf{R}/\partial\mathbf{U}$ was required to calculate all nine cost function sensitivities.

4 Conclusion

We have demonstrated for the 2D Euler equations, the capability to accurately calculate "exact" sensitivities for the discrete flow equations in per-iteration times that twice as fast as analysis times. It was found, that although flux limiting functions in the analysis scheme are discontinuous with respect to $\mathbf{U}$ at some points, the functions are piecewise differentiable, and no problems were found in this regard. We have used exactly the same explicit multistage Runge-Kutta algorithm to solve both direct (23), adjoint (25) and flow (22) equations.

The potential of this method is that, due to the very low memory requirements (the adjoint equation requires about the same amount of memory as the flow solver), it readily extends to 3D and Navier Stokes. This extension to 3D Euler has been readily performed with an analysis code very similar to FELISA [14], which is in widespread use. Due to the larger required memory, the 3D sensitivity calculation procedure is not as fast vis-a-vis the analysis code as the 2D version, since flux jacobian terms must be recalculated every iteration. However, it remains a practical means of calculating accurate sensitvities and potential exists for realizing some of the gain found in 2D by storing parts of the $\partial\mathbf{R}/\partial\mathbf{U}$ matrix. Furthermore, use of the adjoint method practically *eliminates* the dependence of the cost function gradient calculation on the number of design variables.

Further work will concentrate on the development of more general geometry parameterization procedures and increasing the computational efficiency by parallelizing the code.

Acknowledgements

The authors would 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.

Application	Design Variable	AD (Adjoint)	AD (Direct)	FD
2D Multi	flap rot.	1.93e-4	1.91e-4	1.88e-4
3D Single	camber	-0.634	-0.635	-0.656
3D W/B	camber 1	-3033.	-3020.	-2807.

Table 1: Comparison of Various Sensitivities

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] M. Drela. Design and Optimization Method for Multi-Element Airfoils. *AIAA*, 93-0969, 1993.

[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] M. Giles. An Energy Stability Analysis of Multi-step Methods on Unstructured Meshes. Technical Report 87-1, MIT CFD Lab, 1987.

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

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

[10] 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.

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

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

[13] 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.

[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.

[22] A. Von Doenhoff and A. Abbot. *Theory of Wing Sections.* Dover Publications, Inc, Mineola, NY, 1959.

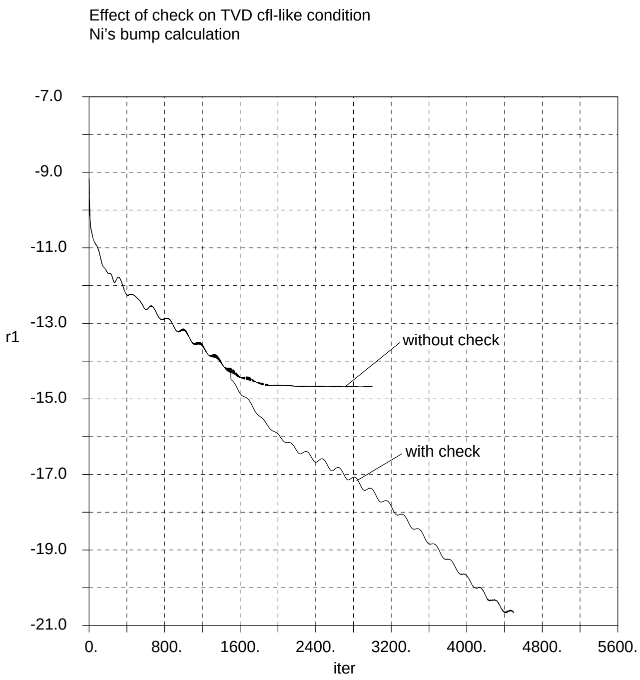

Figure 2: Convergence history for calculations with and without TVD CFL-like check

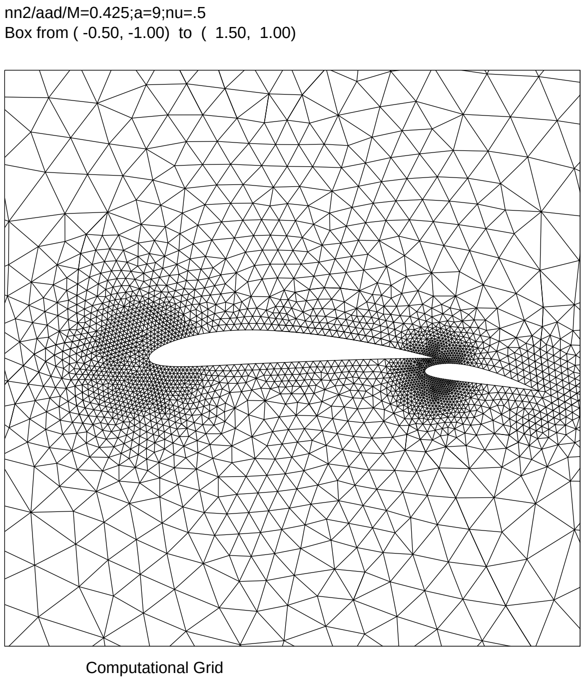

Figure 3: Grid used for seed 2D multielement airfoil calculation

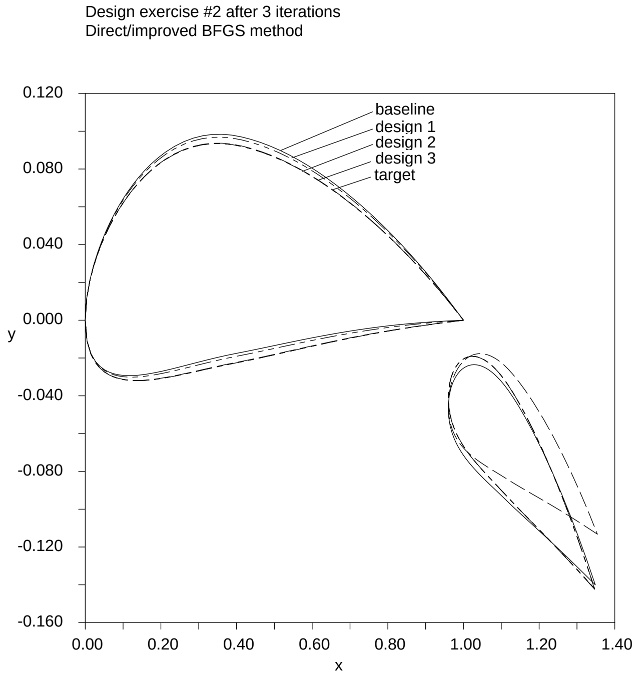

Figure 5: Geometry evolution for 2D multi element optimization exercise

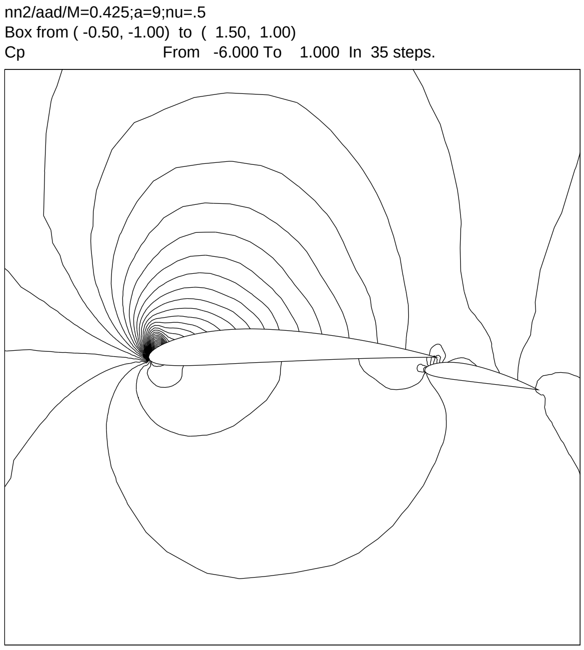

Figure 4: Pressure contours for 2D multielement airfoil at M=0.425, aoa=9deg

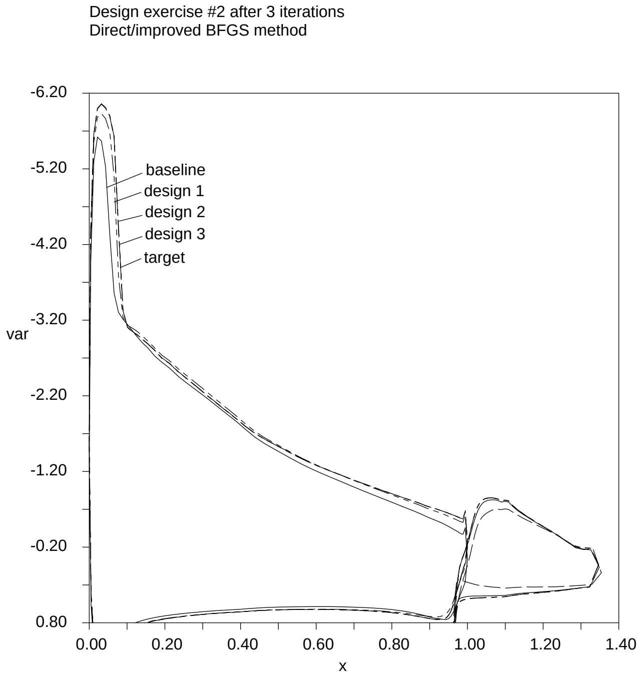

Figure 6: Pressure evolution for 2D multi element optimization exercise

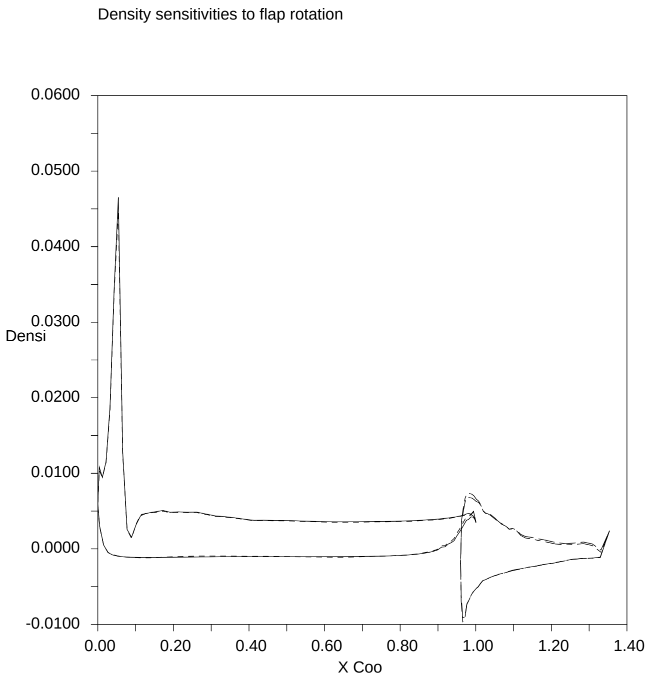

Figure 7: Analytic and FD density sensitivity v. x

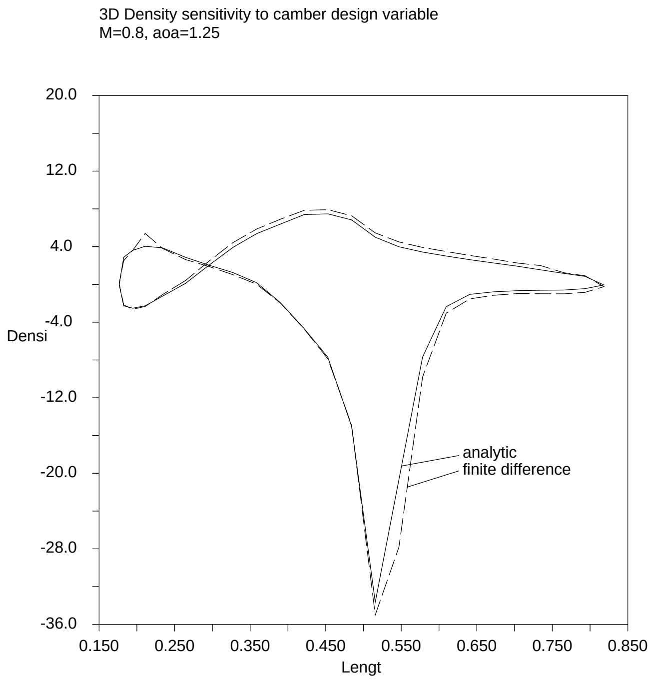

Figure 9: 3D Analytic and FD density sensitivity v. x at y=-.35 symmetry plane

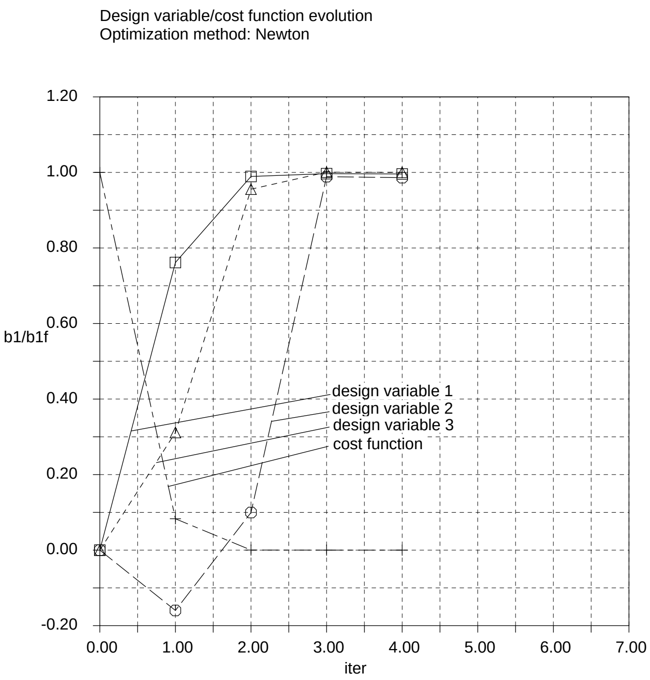

Figure 8: Cost function and design variable evolution for 2D multi element optimization exercise

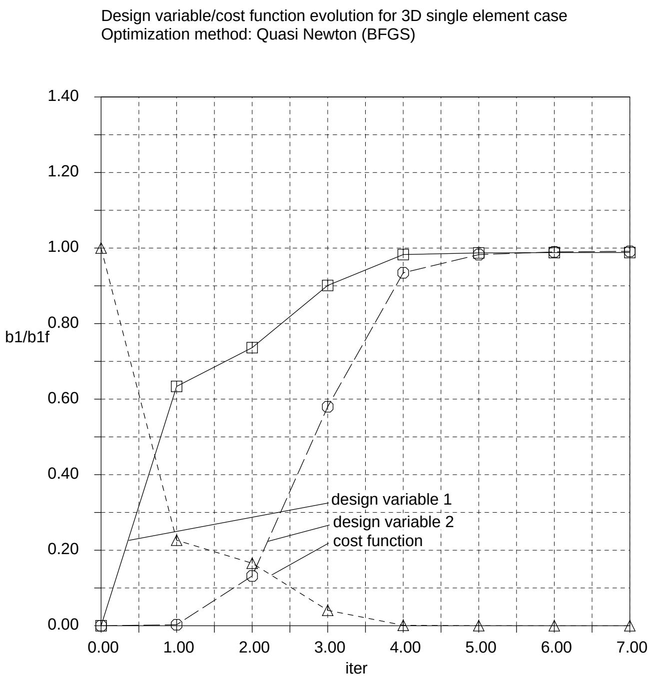

Figure 10: Cost function and design variable evolution for 3D infinite wing optimization exercise

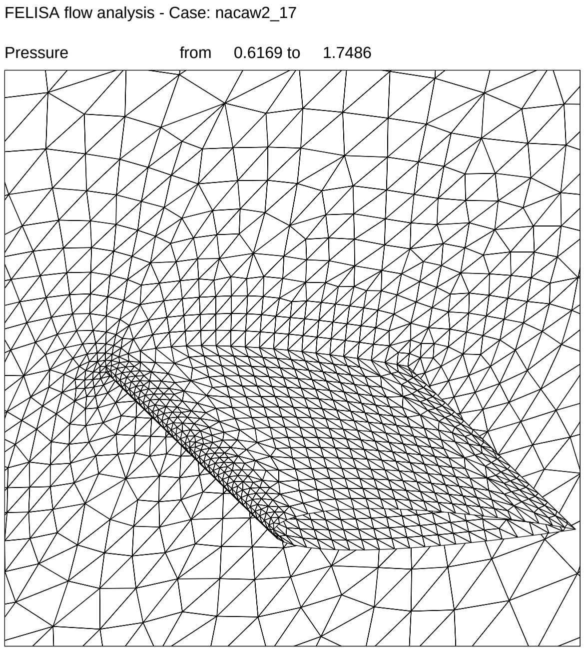

Figure 11: Grid used for baseline 3D infinite wing calculation

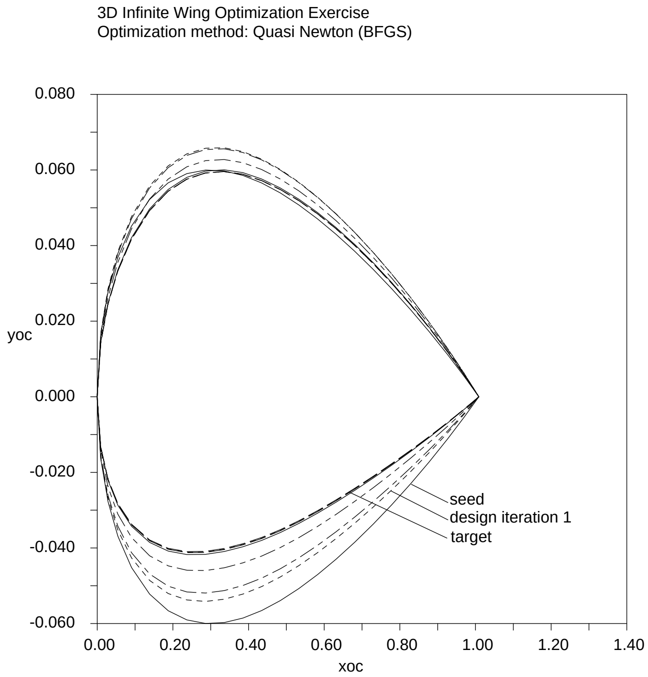

Figure 13: Geometry evolution for infinite wing optimization exercise

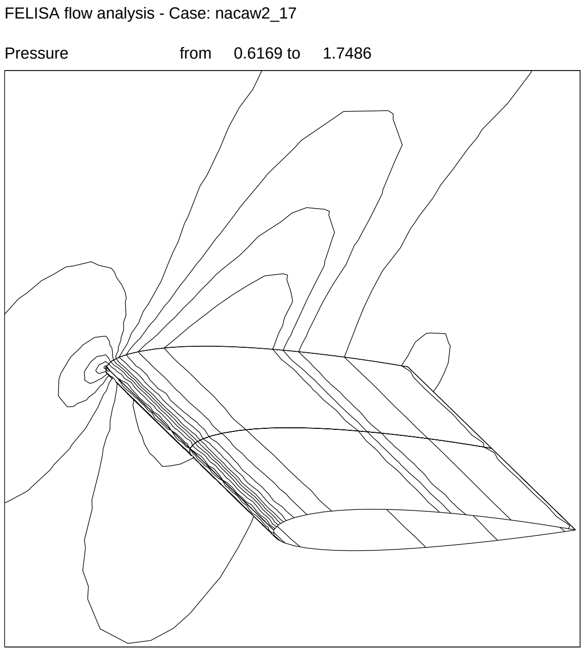

Figure 12: Pressure contours for 3D infinite wing at M=0.8, aoa=1.25deg

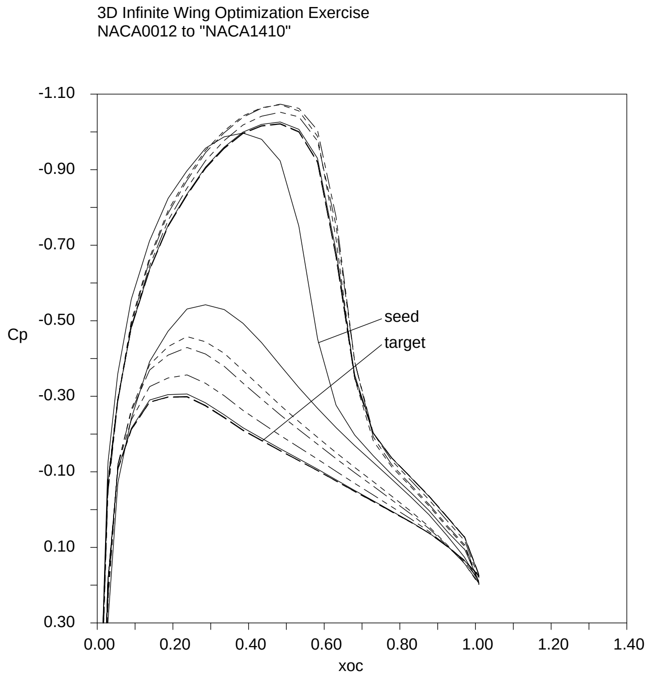

Figure 14: Pressure evolution for 3D infinite wing optimization exercise

FELISA flow analysis - Case: case

Density from 0.6525 to 1.2757

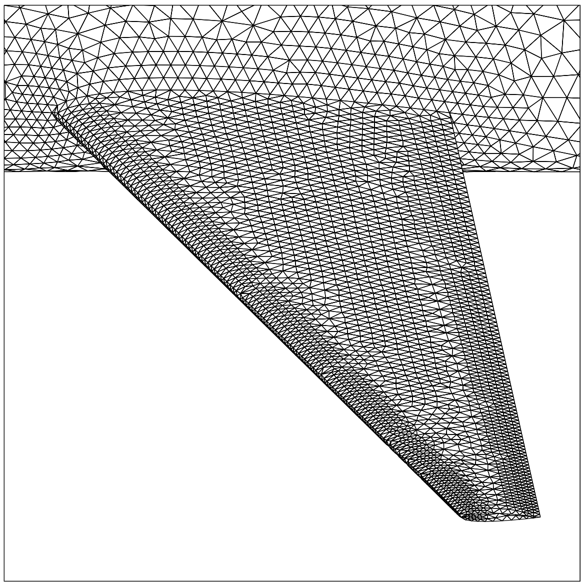

Figure 15: Surface grid for baseline 3D wing/body calculation

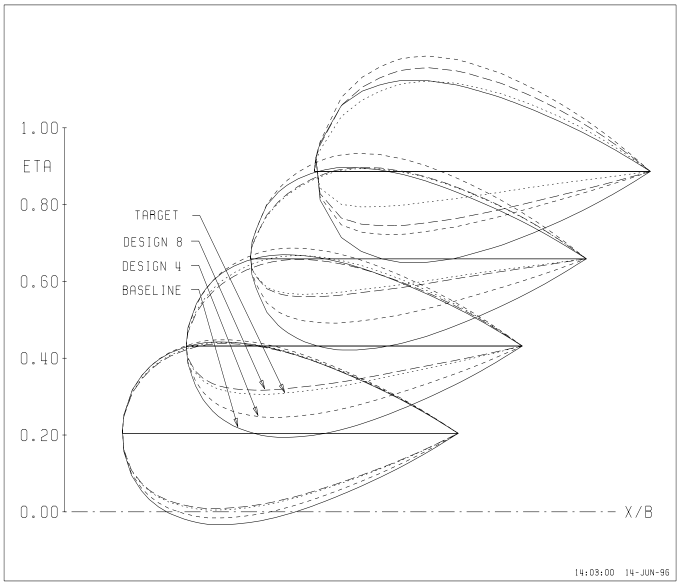

Figure 17: Geometry evolution for wing/body optimization exercise

FELISA flow analysis - Case: case

Pressure from 0.6464 to 1.7296

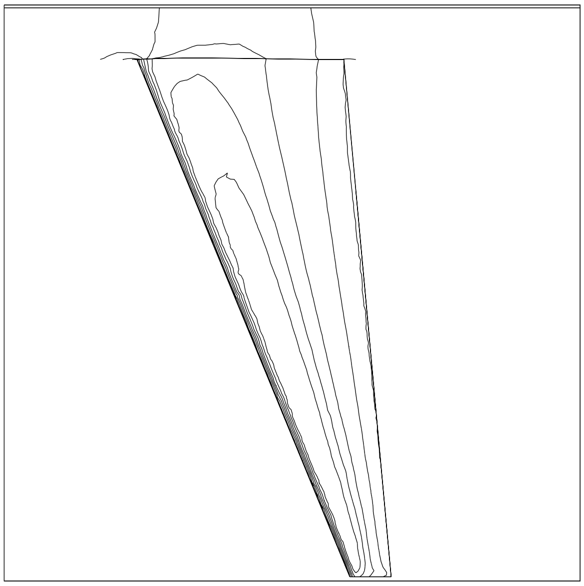

Figure 16: Surface pressure contours for 3D wing/body at M=0.8, aoa=1.25deg

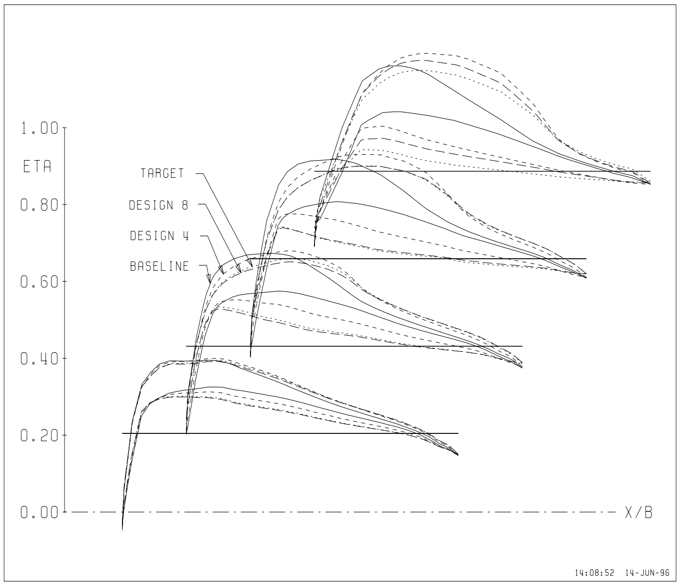

Figure 18: Pressure evolution for wing/body optimization exercise

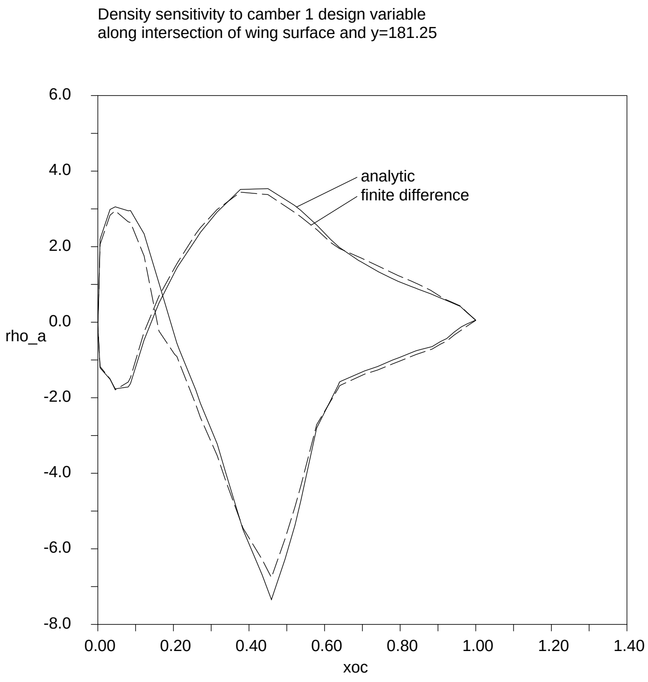

Figure 19: 3D Wing/body Analytic and FD density sensitivity v. x/c at 67.5% semispan

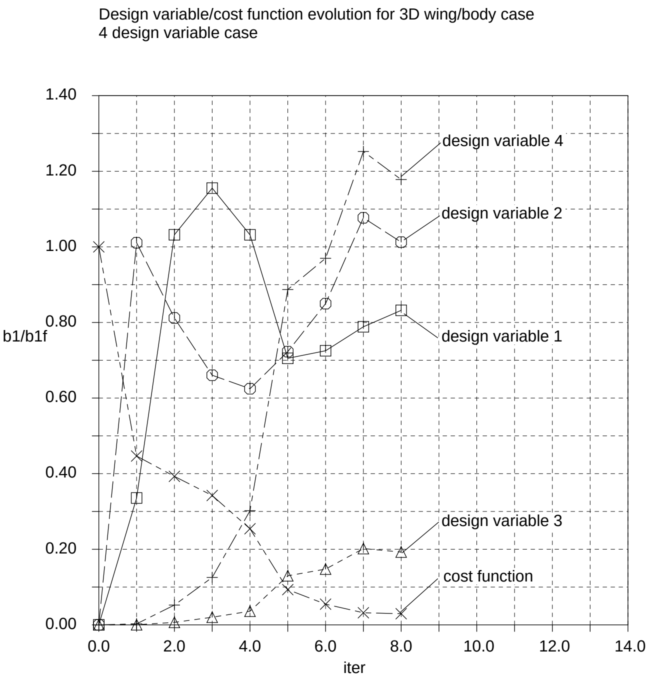

Figure 20: Cost function and design variable evolution for wing/body optimization exercise

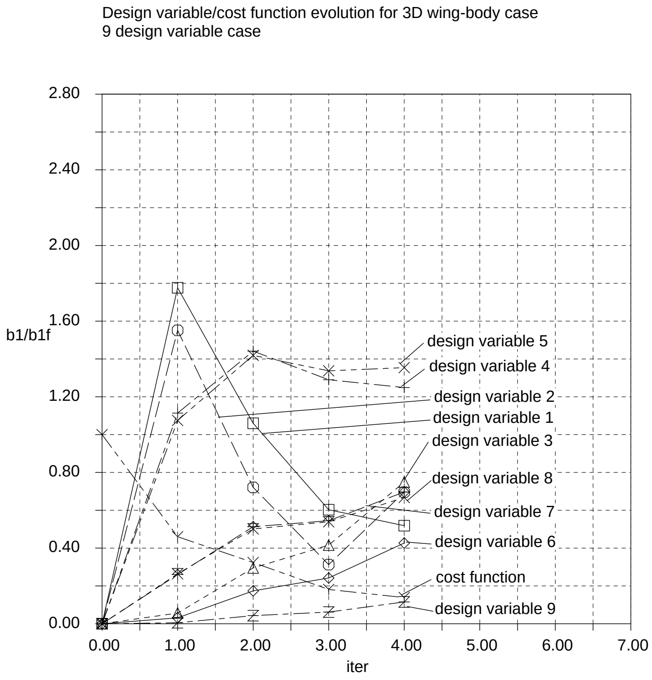

Figure 21: Cost function and design variable evolution for 2nd wing/body optimization exercise

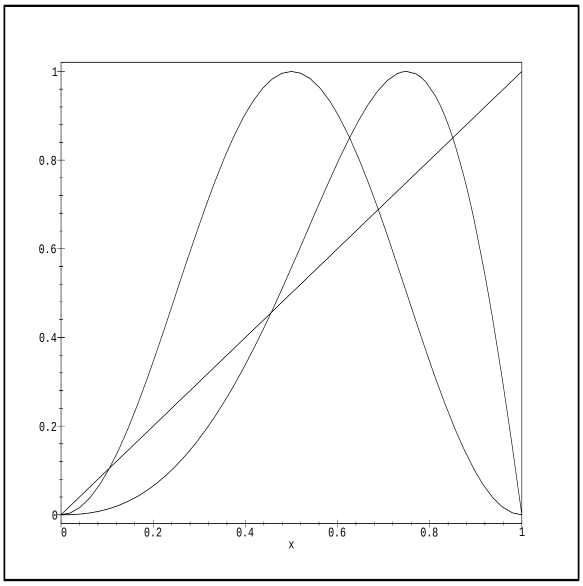

Figure 22: Chordwise components of design variables for 2nd wing/body optimization exervise