AERODYNAMIC OPTIMIZATION ON UNSTRUCTURED MESHES WITH VISCOUS EFFECTS *

Jonathan Elliott† and Jaime Peraire‡
Fluid Dynamics Research Laboratory
MIT Department of Aeronautics and Astronautics

Abstract

A method is presented for performing lift-constrained drag minimization based on the 3D Euler equations and the 2D laminar Navier-Stokes equations. For the 3D Euler equations, the optimization system is based on a flow analysis scheme which uses gradients calculated on an elemental basis. For the 2D Navier-Stokes system, a change was made in the flow analysis scheme to one based on gradients calculated using a finite volume approach due to the fact that this scheme has drastically reduced memory costs associated with the storage of the residual jacobian, and allows direct extension to 3D Navier Stokes. Optimization exercises are presented which demonstrate the effectiveness of the optimization system in finding credible optimal geometries.

1 Introduction

Optimization based on the Euler equations is a useful capability for performing inverse pressure design of aerodynamic bodies especially for high Reynolds number flows where boundary layers are thin and the decambering associated with their displacement surfaces causes only small shock movements and lift changes. However almost any drag minimization exercise based on the Euler equations and applied to modern supercritical wings in cruise condition is *doomed to failure*. This is because the baseline wings have been designed to maximize L/D by pushing the upper surface shock as far aft as possible - at least over the highly-loaded outboard portion of the wing where the transonic effects are the most severe and where the wave drag is usually the highest. The beneficial lift improvement associated with this aft movement of the shock is counterbalanced by an increasingly severe adverse pressure gradient in the recovery region which typically, at shock positions of around $60\% - 70\%$, causes trailing edge separation and the associated detrimental effects of drag increase and lift loss. Hence it is clear that any drag minimization exercise applied to supercritical transport wings must include viscous effects.

Other flow regimes in which viscous effects play a large role are those associated with high lift takeoff and landing configurations. In these cases, accurate modelling of the outer flow is often not possible *even for fully attached flow*, without inclusion of viscous effects . Typical regimes feature boundary layers whose displacement thickness form a significant proportion of the gaps (between elements) through which they pass. Hence these boundary layers have a large impact on the beneficial lift improvement often found by the interaction between elements. (This interelement interaction is heavily exploited by aerodynamic design engineers – but typically using unsatisfactory tools such as 2D viscous or 3D inviscid CFD tools, supported by copious expensive wind tunnel testing.) Furthermore local areas of separation are often present in these regimes, and indeed the maximum lift is often set by the onset of trailing edge separation on one or more elements. Further complications include off the surface slat wake flow reversal which can have a large impact on the lift achieved by landing configurations.

One effective way of including viscous effects is to solve a coupled viscous/inviscid system incorporating, for example, a viscous solver which solves the integral boundary layer equations [8]. This can result in a system which generates solutions more efficiently than Navier-Stokes solvers which typically require many points to resolve the boundary layer, resulting in both higher cost per iteration due to more mesh degrees of freedom and more iterations to convergence due to the requirement of smaller timesteps for stability, for both explicit and implicit time relaxation schemes. There are three factors which stack up against this option. Firstly, for separated flow, use of the direct coupling method (which most naturally fits into our current algorithm) results in an unstable scheme. Fully simultaneous [8] and semi-inverse schemes are possible alternatives which are stable but each would require significant amounts of work to incorporate into our current (explicit or point implicit) relaxation scheme. Secondly, while impressive work has been done on fully 3D integral boundary layer solvers, [22, 20], confidence in the aerospace industry in their results has not yet reached the levels found for their 2D counterparts and some theoretical issues remain to be resolved such as the imposition of hyperbolic boundary conditions. Application of the 2D results via strip theory may introduce a further source of error. Thirdly, use of displacement surfaces to implement the effect of the boundary layer on the inviscid flow leaves open the difficulty of applying this surface at geometric intersections such as wing/body intersections or wing/strut intersections.

On the other hand solving the Navier-Stokes equations throughout the flowfield is a more natural extension of the Euler optimization capability developed by the authors [9, 10]. Also, this capability can serve as a benchmark for future work towards coupled viscous/inviscid solvers.

*AIAA-97-1849: presented at 13th AIAA CFD Conference, Snowmass, Colorado, June 1997
†Graduate Research Assistant, Member AIAA
‡Associate Professor

Although the algorithm presented herein has only been implemented for the 2D Navier-Stokes equations so far, it represents a significant preliminary step towards our ultimate goal of 3D Navier-Stokes optimization. The features of the 2D algorithm allow direct extension to a practical 3D algorithm. In particular, the residual Jacobian storage scheme implemented here leads to an efficient implementation of a 3D Navier-Stokes optimization algorithm.

2 Algorithms

In the course of extending the Euler optimization capability discussed in [9, 10, 11] to Navier-Stokes, it was found that switching to a spatial discretization scheme which can be described as mixed finite volume/finite element [24, 25] allowed a far more memory-efficient sensitivity calculation scheme than one based on the original scheme with viscous contributions. Since we also present 3D Euler results here and since we discuss the relative merits of either scheme vis-à-vis the sensitivity calculation, we find it appropriate to devote some space to discussion of the main components of either spatial discretization scheme.

Flow Analysis

The first scheme can be considered to be a finite volume scheme directly extended from the inviscid solver discussed in [9, 10]. In this case, exact gradients required by both viscous and artificial dissipation fluxes are found on an elemental basis. This scheme will henceforth be referred to as "Scheme I". The second scheme can be considered to be a mixed finite element/finite volume scheme directly extended from the fundamental algorithm underlying FELISA [25], in which the average gradients over each nodal control volume are used for the artificial dissipation and viscous fluxes. We will henceforth refer to this scheme as "Scheme II". The only difference between the two schemes is the way in which the gradients are calculated.

For illustration purposes, the algorithms are described based on the 2D algorithm. The 3D algorithm is a direct extension thereof. Both schemes begin with the integral form of the Navier-Stokes equations:-

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t} d\Omega = \int_\Omega \frac{\partial}{\partial x_j}(\mathbf{G}^j - \mathbf{F}^j) d\Omega = \oint_\Gamma (\mathbf{G}^j - \mathbf{F}^j) n_j d\Gamma \quad (1)$$

where $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho E)^T$,

$$\mathbf{F}^j = \begin{bmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ u_j(p + \rho E) \end{bmatrix}, \mathbf{G}^j = \begin{bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ u_i \tau_{ij} + k\frac{\partial T}{\partial x_j} \end{bmatrix} \quad (2)$$

and Ω is a closed control volume with boundary, Γ. Here τ_{ij} is the tensor of viscous stresses and is given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\delta_{ij}\frac{\partial u_k}{\partial x_k} \quad (3)$$

We first assume that $\mathbf{G}^j = 0$ and present the underlying inviscid scheme which is almost identical for both. The discretization of $\mathbf{U}$ and $\mathbf{F}^j$ 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 associated with each node i consisting of all triangles having vertex i. The line integration given by Equation (1) is performed exactly around the outer boundary of this control volume. (It is noted in passing that an identical discretization can be achieved by forming a Galerkin weighted residual statement using the same piecewise linear functions as the weight functions, and lumping the mass matrix entries onto the diagonal.) Using this spatial discretization, the following set of semi-discrete equations results:-

$$\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0} \quad (4)$$

where, through the use of an edge-based formulation, the residual at node i can be written as

$$\mathbf{R}_i = \sum_{ik=1}^{K} [\mathbf{R}]_{ik}. \quad (5)$$

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

$$[\mathbf{R}]_{ik} = \sum_j \frac{\mathbf{F}_i^j + \mathbf{F}_k^j}{2} S_{ik}^j - \mathbf{D}_{ik} \quad (6)$$

where S_{ik}^j is the area vector associated with edge ik and is calculated as follows:-

$$(S_{ik}^1, S_{ik}^2) = \left(\frac{x_{k+1}^2 - x_{k-1}^2}{3}, \frac{x_{k-1}^1 - x_{k+1}^1}{3}\right) \quad (7)$$

where $k+1$ and $k-1$ are the 3rd nodes in either element containing edge ik. The second term on the right hand side in Equation (6) is an artificial dissipation flux added to stabilize the scheme and is given by:-

$$\mathbf{D}_{ik} = \frac{S}{2}\mathbf{P}|\mathbf{\Lambda}|(\mathbf{P}^{-1}\Delta\mathbf{U}_{ik} - L(\mathbf{P}^{-1}\Delta\mathbf{U}^-, \mathbf{P}^{-1}\Delta\mathbf{U}^+)) \quad (8)$$

Here $\Delta\mathbf{U}^-$ and $\Delta\mathbf{U}^+$ represent changes in $\mathbf{U}$ along an imaginary line segment extended past the positive and negative ends of the edge, of equal length and direction to those of the edge itself, $x_k^j - x_i^j$. $L(u,v)$ in Equation (8) is a nonlinear averaging function, constructed to retain coefficient positivity for the characteristic differences, $\Delta\mathbf{W}^\pm = \mathbf{P}^{-1}\Delta\mathbf{U}^\pm$, and therefore the LED character of the numerical scheme [19, 10]. It should be noted that if an exact average is returned by $L(u,v)$, then the combination of characteristic differences inside the paranthesis in Equation (8) can be seen to form a third order dissipative flux. (Note that $\mathbf{P}$ is the matrix of eigenvectors and $\mathbf{\Lambda}$ is the matrix of eigenvalues for the Roe matrix, $\mathbf{A}(\mathbf{U}_i, \mathbf{U}_k)$ based on the unit normal associated with S_{ik}^j and $\mathbf{A}$ incorporates the Harten correction [16] to avoid zero eigenvalues.)

The method of calculation of $\Delta\mathbf{U}^-$, $\Delta\mathbf{U}^+$ in Equation (8) is different for Scheme I and Scheme II. For Scheme I,

$\Delta\mathbf{U}^-$, $\Delta\mathbf{U}^+$ are formed exactly from the gradients in the elements that either end of the edge points into [3, 19]. For example,

$$\Delta\mathbf{U}^+ = \sum_j (\frac{\partial \mathbf{U}}{\partial x^j})^+ (x_k^j - x_i^j) = \epsilon_{ir}\Delta\mathbf{U}_{ir} + \epsilon_{is}\Delta\mathbf{U}_{is} \quad (9)$$

where the nodes in the adjacent element have been labelled i,r,s and $\epsilon_{ir}, \epsilon_{is}$ are, by construction, always positive. This feature is also critical in ensuring the LED nature of the scheme.

For Scheme II, $\Delta\mathbf{U}^-$, $\Delta\mathbf{U}^+$ are also calculated from the gradients, but here the gradients are calculated as averages over the control volumes:-

$$\frac{\partial \mathbf{U}_i}{\partial x^j} = \frac{1}{V_i}(\sum_{il=1}^{\mu_i} \frac{\mathbf{U}_i + \mathbf{U}_l}{2} S_{il}^j + \sum_{il=1}^{n_{bou}} \frac{5\mathbf{U}_i + \mathbf{U}_l}{4} S_{il}^l) \quad (10)$$

where μ_i is the number of nodes neighbouring node i, n_{bou} is the number of boundary edges contributing to the residual at node i and subscript il represents the edge connecting nodes i and l. The rigourous LED character of the scheme is lost but Scheme II tends to produce smoother and less dissipative solutions than Scheme I for transonic and low supersonic flows.

For both schemes, the means for inclusion of viscous terms recycles the nodal gradients used for the dissipative fluxes. For Scheme II, the edge-based data structure can be retained, and $\mathbf{F}_i^j$, for example, in Equation (6) is simply replaced by $\mathbf{F}_i^j - \mathbf{G}_i^j$. For Scheme I, on the other hand, inclusion of viscous effects demands that we return to an element-based data structure, whether the viscous terms are included via the Finite Volume or the Finite Element method [26]. This has a detrimental impact on the memory requirements for both flow analysis and sensitivity analysis (as pointed out in the next section).

Also note that for the Scheme II problems presented here, which are all Navier-Stokes problems, it was found desirable to use a smoother limiter function in Equation (8), to avoid the limiter switching to first order in regions of high curvature, such as the edge of the boundary layer. The limiter used was that employed in [15] and is

$$L(u,v) = minmod(2u, 2v, \frac{u+v}{2}). \quad (11)$$

Note that this limiter is more likely to return the third argument, and by examination of Equation (8), results in a scheme with a desirable third order dissipative flux. For the 3D Euler optimization problem, the standard minmod function was used.

Mesh Movement

For the 3D Euler calculations presented here, the interconnected springs algorithm was used to accomplish mesh movement. For the Navier-Stokes calculations, we reverted to a scheme in which the grid movement at a location x_j is simply given by the sum of the airfoil surface modal perturbations corresponding to that value of x_1. This eliminates the potential problem of negative volume elements in the perturbed grid in the boundary layer due to the small mesh spacing in the normal direction. This will obviously have to be dealt with in a more satisfactory manner when extending to multi-element airfoils and 3D Navier-Stokes.

Sensitivity Analysis

Consider, for the purposes of simplicity of notation, that the design variable β contains only one component and so can be written as a scalar β. Furthermore, assume that for each condition j, the drag coefficient is given by

$$C_{D_j} = C_{D_j}(\mathbf{U}_j(\beta), \beta). \quad (12)$$

Differentiating the steady state version of Equation (4) and Equation (12) produce, respectively,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \beta} = -\frac{\partial \mathbf{R}}{\partial \beta} \quad (13)$$

and

$$\left.\frac{dC_{D_j}}{d\beta}\right| = \frac{\partial C_{D_j}}{\partial \beta} + \frac{\partial C_{D_j}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \beta} \quad (14)$$

The first term on the right-hand side of (14) can be calculated relatively easily. The second term is far more difficult. Two approaches are the direct and the adjoint method.

The direct method involves *direct* solution of Equation (13) and substitution of the resulting vector $\partial\mathbf{U}/\partial\beta$ into Equation (14). The means by which Equation (13) is solved is discussed below. Note that Equation (13) must be solved once for each component of β.

The adjoint method is based on the recognition that Equations (13) and (14) can be combined to give:

$$\left.\frac{dC_{D_j}}{d\beta}\right|_{\mathbf{R}=0} = \frac{\partial C_{D_j}}{\partial \beta} - \left[\frac{\partial C_{D_j}}{\partial \mathbf{U}}\right]^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^{-1} \left[\frac{\partial \mathbf{R}}{\partial \beta}\right] \quad (15)$$

Due to associativity of matrix multiplication, the right-hand-side double product calculation may be performed either by first multiplying the Jacobian inverse by the term to its right – which gives the *direct* method – or by the term to its left – which gives the *adjoint* method. It is convenient to write the result of the intermediate calculation for the adjoint method as $-\psi$ and therefore ψ satisfies

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^T \psi = -\left\{\frac{\partial C_{D_j}}{\partial \mathbf{U}}\right\}^T \quad (16)$$

and

$$\left.\frac{dC_{D_j}}{d\beta}\right|_{\mathbf{R}=0} = \frac{\partial C_{D_j}}{\partial \beta} + \psi^T \left\{\frac{d\mathbf{R}}{d\beta}\right\} \quad (17)$$

where ψ is called the *adjoint* variable.

Time Integration

It is possible to adopt exactly the same relaxation algorithm to solve equations (4), (13) and (16). This is done by introducing an artificial unsteady term into, for example, the adjoint equation.

$$\frac{d\psi}{dt} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^T \psi - \left[\frac{\partial C_{D_j}}{\partial \mathbf{U}}\right]^T = 0 \tag{18}$$

These schemes possess the same stability properties because the spectral radii corresponding to the direct and adjoint sensitivity analysis schemes are identical to that of the linearized flow analysis scheme which governs its asympotic convergence behaviour. The spectral radius of the error-mode amplification matrix implied by Equation (18) is obviously the same as that for the linearized flow analysis.

In the current research we have used a multi-stage explicit time-stepping scheme for the Euler calculations. For the Navier-Stokes calculations it was felt that an implicit scheme was needed due to the small sizes of the elements in the boundary layer in the normal direction. For preliminary studies a point implicit time-stepping algorithm scheme was used. It was found to allow at least a doubling of the asymptotic convergence rate for one fourth of the per-iteration CPU cost of the multistage scheme for the adjoint calculation. For a given problem we always use the same time marching algorithm for both flow and adjoint analysis.

The point implicit algorithm follows the ideas elucidated in [15, 14], but differs in that the exact block diagonal entries of the $\partial \mathbf{R}/\partial \mathbf{U}$ matrix are used on the left hand side as opposed to only the first order terms. In brief, the relaxation scheme is

$$\mathbf{L}_i \Delta \mathbf{U}_i = (\Omega_i + \Delta t_i \left.\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right|_{ii}) \Delta \mathbf{U}_i = \Delta t_i \mathbf{R}_i \tag{19}$$

where

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \Delta \mathbf{U}_i. \tag{20}$$

To extend this timestepping scheme to the adjoint system, it is necessary to replace $\mathbf{L}$ with $\mathbf{L}^T$, since this ensures that the eigenvalues of the error amplification matrix remain the same as for linearized flow analysis. Justification for taking this approach was verified when it was observed that flow, direct sensitivity and adjoint analysis calculations exhibited identical asymptotic convergence behaviour, as can be seen in Figure 5.

The effect of the no-slip boundary conditions is to eliminate the momentum equation residuals at the wall from the system to be solved. Therefore the adjoint variables (which are in essence Lagrange multipliers for each nodal residual equation[11, 17]) corresponding to the momentum equations at the wall should have no impact on the final adjoint solution elsewhere. This was recently pointed out in reference [2]. This has been accomplished for our adjoint solver by performing an à posteriori sweep on the wall boundary nodes in which the momentum adjoint variables are zeroed out.

Finally, a viscous correction was made to the Δt_i in an effort to ensure that the viscous time step limit is not exceeded, thereby compromising stability. This correction is given by

$$\Delta t_i = \frac{\Delta t_i^{inv}}{1 + 4/Re_{\Delta s_i}} \tag{21}$$

where Δt_i^{inv} is the time step limit found using linearized Fourier analysis for the inviscid scheme [12] and $Re_{\Delta s_i}$ is the Reynolds number based on the length of the smallest edge in the elements surrounding node i. It has been found by many researchers that this stability limit can have a large influence for well-resolved, low Reynolds number flows. For the 3D solutions presented here, Δt_i^{inv} also included a factor to ensure satisfaction of the TVD CFL-like condition [9].

Jacobian storage

One of the main obstacles to sensitivity calculation based on the Euler and Navier-Stokes equations using the direct and adjoint approaches is that associated with the large memory cost of storing the residual Jacobian matrix, $\partial \mathbf{R}/\partial \mathbf{U}$. The matrix is usually sparse due to the fact that the finite volume or finite element residual statements based on piecewise linear variation of the state vector, $\mathbf{U}$, result in only the immediate neighbours having an influence on $\mathbf{R}_i$, the residual at node i (an exception would be a spectral scheme). Dissipative fluxes required for stability and shock capturing can extend the stencil to include the neighbours of the neighbours depending on the formulation used. Nonetheless, it is obviously wasteful to store the whole sparse matrix, $\partial \mathbf{R}/\partial \mathbf{U}$.

One obvious improvement is to store the Jacobian entries on an edge basis. This results in the storage costs scaling as $O(N)$ rather than $O(N^2)$ for the full matrix. However the costs are still large and prohibitive for 3D cases unless a parallel architecture is used.

Several approaches have been used in the past to circumvent these memory cost problems. In some cases [5], schemes have been limited to first order accuracy in order that $\partial \mathbf{R}/\partial \mathbf{U}$ may be more easily stored due to the smaller stencil. An alternative option is to resort to schemes in which the entries in $\partial \mathbf{R}/\partial \mathbf{U}$ are recalculated "on-the-fly" [21, 9]. The authors found that although this obviously provides enormous memory savings, it results in large CPU cost increases [11] – about a factor of four increase for our 2D and 3D Euler schemes.

Use of the continuous sensitivity analysis approach [18, 27] avoids these problems since the relaxation scheme used for solving the flow analysis system can be recycled to solve the adjoint problem. Stability can be provided either by reverse biasing of the difference operators due to the reversed direction of the zone of dependence for the adjoint problem or simply by addition of a dissipative flux of the same form as that for the flow analysis scheme. Of course, one of the drawbacks of the continuous sensitivity analysis approach is that calculated sensitivities cannot be exact except at the limit of an infinitely fine grid.

The extension to Navier-Stokes introduces further complications. The rationale for using Scheme I [9, 10] originally was that a very small stencil results for the Euler equations with the resulting low memory costs for $\partial \mathbf{R}/\partial \mathbf{U}$. Unfortunately, when viscous stresses and heat fluxes are included when extending to the Navier-Stokes equations, Scheme I loses its small stencil advantage. (This is because exact integration of, for example, the viscous fluxes results in dependencies on the 3rd (and 4th in 3D) node in the 2 elements containing a given edge.) Switching to an element-based data structure because of this, causes an increase in memory for both flow analysis and a large increase in the size of $\partial\mathbf{R}/\partial\mathbf{U}$ for sensitivity analysis.

At first glance, reverting to the control-volume-average gradients in Scheme II would appear to have a still higher memory cost for $\partial\mathbf{R}/\partial\mathbf{U}$ storage especially in 3D. One might erroneously arrive at this conclusion because the residual contribution for a given edge is dependent on the nodal gradients at either end of the edge, and the gradient at node i is in turn dependent on its immediately adjacent neighbours as shown in Equation (10). Therefore the residual contribution for edge ik can be expressed as:-

$$[\mathbf{R}]_{ik} = [\mathbf{R}]_{ik}(\mathbf{U}_i, \mathbf{U}_k, \mathbf{U}_{i1}, \cdots, \mathbf{U}_{i\mu_i}, \mathbf{U}_{k1}, \cdots, \mathbf{U}_{k\mu_k}) \quad (22)$$

$[\mathbf{R}_{ik}]$ can be dependent on as many as $(2 \times 10 + 2)$ nodes in 2D and $(2 \times 40 + 2)$ nodes in 3D. Direct calculation of each dependency obviously results in a prohibitive memory cost for $\partial\mathbf{R}/\partial\mathbf{U}$.

This problem can be avoided by calculating $\partial\mathbf{R}/\partial\mathbf{U}$ in two stages. It can be seen from Equations (2), (6) and (8) that the residual contribution for a given edge can be written as

$$\mathbf{R}(\mathbf{U}) = \mathbf{R}(\mathbf{U}, \mathbf{U}_{x_i}; \mathbf{U}) \quad (23)$$

Following numerous previous researchers, e.g.[4], we apply the chain rule and obtain,

$$\begin{aligned} \frac{d\mathbf{R}}{d\mathbf{U}} &= \frac{\partial\mathbf{R}}{\partial\mathbf{U}_I} + \frac{\partial\mathbf{R}}{\partial\mathbf{U}_{II}} \\ &= \frac{\partial\mathbf{R}}{\partial\mathbf{U}_I} + \frac{\partial\mathbf{R}}{\partial\mathbf{U}_{x_j}}\frac{\partial\mathbf{U}_{x_j}}{\partial\mathbf{U}} \end{aligned}$$

where we have used the shorthand $\mathbf{U}_{x_j} = \partial\mathbf{U}/\partial x_j$. This doesn't appear to be much of a savings since $\partial\mathbf{U}_{x_j}/\partial\mathbf{U}$, at first glance, appears to be quite large. However, it is quickly found after examining the expression used to calculate $\mathbf{U}_{x_j}$, Equation (10), that each conservative variable component is only dependent on neighbouring nodal values of that same conservative variable component. Therefore $\partial\mathbf{U}_{x_j}/\partial\mathbf{U}$ is only nonzero if the same conservative variable components are being compared. Also $\partial\mathbf{U}_{x_j}/\partial\mathbf{U}$ is the same for all conservative variable components. Therefore $\partial\mathbf{U}_{x_j}/\partial\mathbf{U}$ is quite small, and $\partial\mathbf{R}/\partial\mathbf{U}$ can be stored using about the same amount of memory as was required to store the Euler residual jacobian based on the Scheme I discretization.

Optimization Strategy

A subspace BFGS algorithm is used to minimize drag with constraints on lift for all three examples discussed herein. The algorithm is elucidated in [11], but we will give a brief description. An increasingly accurate estimate to the Hessian (of the cost function with respect to the design variables) is developed in the subspace, $\mathbf{Z}$, orthogonal to the constraint gradients. This estimate is based on the usual BFGS formula, but uses variables which are all projected into the subspace [13]. The matrix $\mathbf{Z}_k$ which is used to perform the projection is found by performing an LQ decomposition of the constraint gradients, $\partial C_L/\partial\beta_j$. This formulation requires the solution of two adjoint problems per line search – one for C_L and one for C_D. The method appears to be quite robust since, as for the unconstrained BFGS algorithm, the Hessian approximation is guaranteed to remain positive definite provided the exact Hessian is positive definite and provided sufficient progress is made towards the minimum for each line search [13].

3 Validation

Flat plate flow analysis validation The grid used is shown in Figure 2. No slip boundary conditions are applied along the lower boundary from $x = 0$ to $x = 1.5$; free slip boundary conditions are applied on the lower boundary from $x = -1.5$ to $x = 0$; non-reflecting Riemann boundary conditions are applied at the left boundary; pressure is specified while the appropriate characteristic variables are extrapolated from the interior along the top and right boundaries. Flow is from left to right.

This test case or variations of it have been extensively tested by the second of the authors and other researchers [24, 1, 28]. At sufficiently large distances from the leading edge, the boundary layer profile is expected to closely approximate the Blasius profile – provided sufficient spatial resolution is used. The profiles found at $x = 1.215$ using Scheme II for successively coarser grids (coarsened by removing every other vertical and horizontal in multigrid fashion [1]) are shown in Figure 1. It can be seen that the solution spatially converges to a close approximation of the Blasius solution.

Airfoil flow validation A calculation was performed, once again using Scheme II, for the NACA0012 with freestream conditions of $M = 0.8, \alpha = 3.5°, Re = 2000$. The grid is shown in Figure 27. The C_p and M distributions are shown in Figures 28 and 29. There are 7081 points in the grid and 17 points in the boundary layer at the trailing edge, although only the first 9 elements are "structured" [23]. This small number of "structured" elements in the boundary layer is the cause of the wiggles observable in the C_p distribution. In spite of these low level errors, surface C_p and C_f distributions agree quite well with the corresponding distributions as found along the displacement surface by MSES, a coupled Euler/integral boundary layer code [6]. This can be seen in Figures 3-4. On the lower surface, agreement is quite good, while on the upper surface, the differences are due to the non-zero normal pressure gradient through the boundary layer. The C_p difference between displacement surface and the wall was sampled at several locations in the Navier-Stokes solution and found to agree quite closely with the differences observed in Figure

3. Further validation work at higher Reynolds numbers is currently in progress to confirm this hypothesis.

Airfoil sensitivity validation Sensitivity calculations were performed based on the solution described in the previous paragraph. As expected, and as found for the corresponding Euler calculations, asymptotic convergence rates for flow, adjoint and sensitivity analyses were very close as can be seen in Figure 5. Comparison of the resulting $\partial \rho u/\partial \alpha$ distribution (Figure 6) with that found using the finite difference method based on $\Delta\alpha = 0.01$ deg (Figure 8), reveals good agreement. Comparison of $\partial C_l/\partial \beta$ and $\partial C_d/\partial \beta$ as found by finite difference, adjoint and direct methods are shown in Table 1 below for design variables of α and a NACA 4-series meanline camber mode. Quite good agreement is found for α. The discrepancies for the camber mode are believed to be due to the finite difference step size used in making that estimate. $\Delta\beta_c = 0.001$ was used corresponding to a movement in the airfoil surface of 0.1% of chord.

To put a proper perspective on the source of this discrepancy, much can be learned by examining the sensitivities of the only flow quantities that contribute to the lift and drag. Surface pressure sensitivities are compared in Figure 7 revealing good agreement. Surface values of $\partial \tau_{11}/\partial \beta$ are shown in Figure 9 and also reveal good agreement. Similarly good agreement is observed for τ_{12} and τ_{22}. Since these 4 scalar values are the only ones that contribute to C_l and C_d, $\partial C_l/\partial \beta$ and $\partial C_d/\partial \beta$ are expected to show similarly good agreement. The larger than expected discrepancies shown in Table 1 are presumably due to the effect of small errors when the integration is performed. The difference of two large numbers is taken to obtain a small number, a scenario which is conducive to error magnification. Nonetheless it would probably be concluded at this juncture that the analytic derivatives are the ones with higher accuracy after examination of Figures 7 and 9. Further tests in the course of the optimization exercises by comparison of $\Delta\beta_j \partial C_l/\partial \beta_j$ with ΔC_l and $\Delta\beta_j \partial C_d/\partial \beta_j$ with ΔC_d provided further definitive evidence in this regard.

β	$\partial C_l/\partial \beta\|_{FD}$	$\partial C_l/\partial \beta\|_{adj}$	$\partial C_l/\partial \beta\|_{dir}$
camber	-0.9086	-1.0075	-1.0075
α	0.04707	0.04811	0.04811
	$\partial C_d/\partial \beta\|_{FD}$	$\partial C_d/\partial \beta\|_{adj}$	$\partial C_d/\partial \beta\|_{dir}$
camber	-0.0385	-0.04902	-0.04917
α	0.00627	0.00634	0.00634

Table 1: Comparison of sensitivities to α and NACA 4-series meanline mode

4 Results

3D Euler, Multipoint, lift-constrained drag minimization exercise For this case, Scheme I was used. This 3D case [11] is a double point drag-minimization exercise in which the cost function is given by $F = 0.5(C_{D_1} + C_{D_2})$. The two flight conditions are $(M_{\infty_1} = 0.9, C_{L_1} = 0.450)$ and $(M_{\infty_2} = 1.6, C_{L_2} = 0.125)$. The baseline geometry is a wing-body configuration with an area-ruled body. The wing has a low aspect ratio $(AR = 2.67)$, with leading edge sweep, $\Lambda_{LE} = 45°$, trailing edge sweep, $\Lambda_{TE} = 0°$. It has uniform airfoil maximum thickness of 5% of chord, has uniform twist of $\phi = 0°$ and is uniformly uncambered across the span. The baseline grid contains 1.13 million elements and 211,000 points and is shown in Figure 10.

The optimization was performed using twelve design variables. Two of these were α_1 and α_2, the angles of attack at each condition. The other ten were combinations of four spanwise functions and three chordwise functions (twist and two camber modes). The adjoint method based on discrete sensitivity analysis was used to calculate sensitivities. These sensitivities and the more-easily calculated objective functions were fed into a BFGS optimization algorithm. After 7 BFGS iterations, the optimization process has practically converged, with a reduction of around 5.5% in the cost function.

Baseline and final C_p distributions are shown in Figures 11-14 for either condition while sectional geometry and C_p evolution are plotted in Figures 16-18. As expected, the nose has pitched down, the angle of attack has been reduced (resulting in a lower leading edge peak and a weakened or vanished leading edge shock), while the lift has been recovered by a camber increase along the aft part of the wing which is accompanied by an aft movement of the shock. It should be noted however, that the positioning of the shock this far aft would spell trouble in the physically realistic situation due to the presence of viscous effects. At normal Reynolds numbers, one would expect that either shock-induced boundary layer separation would occur or that separation would occur in the severe adverse pressure gradient in the recovery region.

The supersonic drag actually increases slightly but by a far smaller amount than the transonic drag decrease. The reader is referred to reference [11] for a more thorough exposition of the details of this optimization exercise, including design variable definitions, etc.

2D Attached Viscous Optimization Case Scheme II was used for flow analysis for this problem. The baseline geometry was a very thin airoil (with about 1% maximum thickness), with a thickness distribution that had already undergone considerable design and optimization for low Reynolds number viscous flow, albeit at a slightly higher Re=10000. The original airfoil is under consideration for use in micro unmanned aerial vehicles (μ-UAVs). It had significant camber, but it was decided that a well behaved zero-camber airfoil was required as an initial test for the optimization system, so the camber was removed from the airfoil by a simple geometric operation. At $\alpha = 3.5°$, $M = 0.8$, and $Re = 2000$ it was found that the boundary layer C_f distribution was well above zero everywhere. The high Mach number was chosen, not for its realism in representing the flight conditions of μ-UAVs, but because it allows quicker convergence of the flow analysis calculations. It was decided that drag minimization with a lift constraint was an appropriate initial test, even though not much of a drag reduction was expected. The design variables were chosen to be the modal amplitudes of Hicks-Henne camber functions of the form $\Delta y(x) = \sin(\pi x^p)$ or $\Delta y(x) = \sin(\pi(1-x)^p)$with p chosen to be such that peaks occur at 20%, 40%, 60% and 80% of chord. In addition, the angle of attack was chosen as a design variable.

The baseline grid, pressure distribution and Mach distribution are shown in Figures 19-21. The baseline grid contains 8506 nodes and at the trailing edge there are approximately 20 points in the boundary layer. However for this case, the elements with "structure" in the boundary layer number about 16, which is the probable cause of the reduced presence of wiggles in the solution compared to the NACA0012 described in the previous section. In spite of the low level error associated with these wiggles, the surface C_p distribution was found to match that found by MSES even more closely than the match shown in Figures 3-4. Note that although there is locally supersonic flow, no shocks are present, although the ability to capture these shocks is an inherent part of the flow analysis algorithm.

The final C_p distribution is shown in Figure 22 while the evolution of design variables, C_f distribution, C_p distribution and geometry are shown in Figures 23-26. It can be seen that the minimum appears to have been almost reached after about 4 line searches which is about what should be expected for a well behaved design surface. The cost function has only decreased by 18 counts or about 2.2% of the baseline C_d. This modest decrease was expected since the baseline solution exhibited quite healthy boundary layer behaviour. Note that the decrease has come about partially by a reduction of the leading edge pressure peak. The resulting reduction in the net pressure increase and adverse pressure gradient along the upper surface causes the momentum thickness to be lower at the upper surface trailing edge, indicating a more healthy boundary layer and lower drag.

Also it should be noted that this optimization exercise was reproduced using MSES and and LINDOP, its associated optimization driver [7]. A similar reduction in C_d was observed, although the final geometry was slightly different. It is believed that this is due to the relatively shallow minimum.

2D Separated Viscous Optimization Case Scheme II is also used for flow analysis for this problem. This case is a more challenging one due to the presence of separation over the aft 60% of the upper surface. However, more of a drag reduction is expected due to the poor initial health of the upper surface boundary layer. It was felt that thickness design variables would possess powerful leverage over setting the separation point on the upper surface, so two thickness design variables with peaks at 20% and 40% were chosen as well as one camber design variable with a peak at 50%, along with the angle of attack. In addition to the lift constraint, for this case, it was decided that an area constraint should also be imposed. Otherwise, the airfoil would be driven uselessly to zero thickness. The baseline grid, C_p and M distributions are shown in Figures 27-29.

It should also be noted that this optimization exercise did not proceed nearly as smoothly as the one discussed in the previous paragraph. The presence of the separated flow seemed to cause two major undesirable effects. Firstly, as soon as the flow separates on both lower and upper surface, the flow becomes unsteady due to the onset of vortex shedding, and no fully converged flow or adjoint solution can be found. The way this undesirable feature was dealt with in the optimization algorithm was to assume a very large value of the objective function at that point and thereafter to trace back up the line search direction until a steady solution could be found, and then to search for a new direction from there. The second undesirable effect was that the presence of separation caused larger deviations from quadratic behaviour than usual. This appears to have slowed down the convergence of the approximate Hessian to the value found at the minimum. It is not clear that a good approximation has been made for the final design point depicted. The optimization process was stopped after 3 line searches when successive iterations resulted in the search direction not changing significantly with the design poised to enter the part of design space in which the above-mentioned vortex-shedding unsteadiness occurs.

The final C_p distribution is shown in Figure 30 while the evolution of design variables, C_f distribution, C_p distribution and geometry are shown in Figures 31-34. The cost function for this case has decreased a far larger 120 counts or about 10.6% of the baseline C_d. This larger decrease was expected due to the presence of separation in the baseline solution. Note that the decrease has been accomplished partially by a reduction of the leading edge pressure peak. Also the thickness distribution has been redistributed such that the maximum is about 15% further aft. Like the early natural laminar flow airfoils [29], this delays the start of the adverse pressure gradient to aft of that maximum thickness point. Consequently the upper surface separation point moves from 45% to about 65%. This movement appears to be limited by the appearance of separation on the lower surface and the resulting above-mentioned unsteadiness as can be seen in Figure 32.

Agreement with a similar optimization exercise performed using LINDOP and MSES revealed similar initial behaviour although the LINDOP geometry evolved to one with negative camber – a configuration not allowed by the current Navier-Stokes optimizer due to the intervening onset of vortex shedding as the maximum camber passes through zero.

5 Conclusion

The problem addressed was the implementation of a capability to perform lift-constrained drag minimization on unstructured grids. To this end a previously described Euler optimization system was extended to perform this task in a multipoint fashion for 2 flight conditions. This study produced credible optimal geometries although it pointed to the need to include viscous effects in order to avoid unrealistic adverse pressure gradients.

Subsequently, a 2D laminar Navier-Stokes optimization capability was implemented and tested on two cases. One involved fully attached flow while the other contained an extensive region of separated flow. In spite of some complications unearthed in the course of the separated flow case – due to nonsmoothness of the design space surface and unsteadiness for some regions of design space – these studies confirm the effectiveness of the optimization system by also producing credible optimal geometries.

As a result of this study, it now appears that 2D laminar optmization is a capability that is well in hand. Extension of the capability to one based on 3D turbulent flow analysis is the problem area that should now be addressed to allow a practical industrial tool to evolve. It should be noted that most features of the overall approach extend directly as is to 3D Navier-Stokes. For example, the residual jacobian storage system developed here removes this as an obstacle to 3D Navier-Stokes optimization. However, grid generation, time integration and turbulence modelling are key areas on which attention will need to be focussed in the immediate future.

Acknowledgements

We would like to express our thanks to Professor Mark Drela for continuing invaluable assistance at various stages of this project and to Bill Herling and Fritz Roetman of the Boeing Company for many helpful discussions and for providing the 3D geometry. 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 partially supporting this research.

References

[1] S. R. Allmaras. Contamination of Laminar Boundary Layers by Artificial Dissipation in Navier-Stokes Solutions. In *Proc. of the ICFD Conference on Numerical Methods for Fluid Dynamics*, Reading, England, UK, 1992.

[2] W. Anderson and V.Venkatakrishnan. Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation. AIAA-97-0643, 1997.

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

[4] R. Beam and R. Warming. An Implicit Factored Scheme for the Compressible Navier-Stokes Equations. *AIAA Journal*, pages 393–402, April 1978.

[5] 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.

[6] M. Drela. Newton Solution of Coupled Viscous/Inviscid Multielement Airfoil Flows. AIAA-90-1470, 1990.

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

[8] M. Drela and M. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.

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

[10] J. Elliott and J. Peraire. Practical 3D Aerodynamic Design and Optimization using Unstructured Meshes. *AIAA Journal*, accepted for publication, 1997.

[11] J. Elliott and J. Peraire. Three-Dimensional, Multipoint, Lift-Constrained Drag Minimization using Unstructured Meshes. *Aeronautical Journal of the Royal Aeronautical Society*, submitted for publication, 1997.

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

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

[14] P. Gnoffo, R. Candless, and H. Yee. Enhancements to Program Laura for Computation of Three-Dimensional Hypersonic Flow. AIAA Paper 87-0280, January 1987.

[15] P. A. Gnoffo. An Upwind-Biased Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect Gas Flows. TP 2953, NASA, February 1990.

[16] A. Harten and J. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50:235–269, 1983.

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

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

[19] A. Jameson. Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows. AIAA-93-3359, 1993.

[20] B. H. Mughal. A Calculation Method for the Three-Dimensional Boundary-Layer Equations in Integral Form. Master's thesis, Dept. of Aeronautics and Astronautics, MIT, September 1992.

[21] 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.

[22] B. A. Nishida. *Fully Simultaneous Coupling of the Full Potential Equation and the Integral Boundary Layer Equations in Three Dimensions*. PhD thesis, Dept. of Aeronautics and Astronautics, MIT, 1995.

[23] J. Peraire and K. Morgan. Unstructured Mesh Generation Including Directional Refinement for Aerodynamic Flow Simulation. *Finite Elements in Analysis and Design*, 25:343–356, 1997.

[24] 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*, Reading, England, UK, 1992.

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

[26] J. Peraire, J. Peiro, and K. Morgan. Multigrid Solutions of the 3D Compressible Euler Equations on Unstructured Tetrahedral Grids. *International Journal for Numerical Methods in Engineering*, 36:1029–1044, 1993.

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

[28] S. Tatsumi, L. Martinelli, and A. Jameson. Flux-Limited Schemes for the Compressible Navier-Stokes Equations. *AIAA Journal*, 33:252–261, 1995.

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

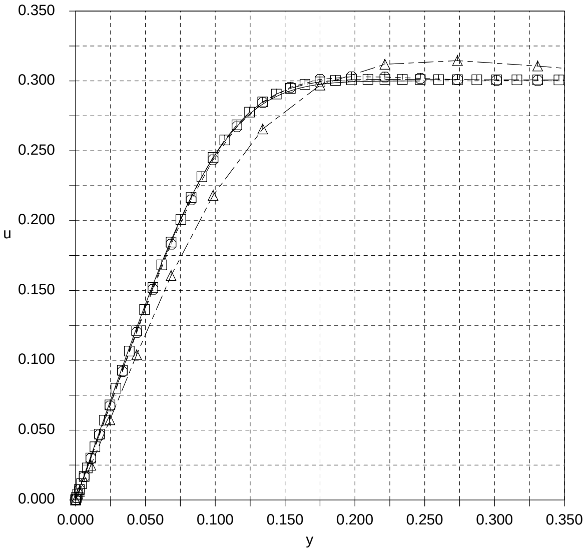

Figure 1: Spatial convergence of boundary layer profile at x=1.2 with local Blasius profile

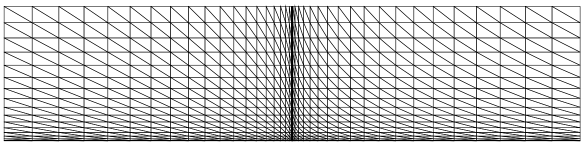

Figure 2: 41x17 grid used for flat plate calculations

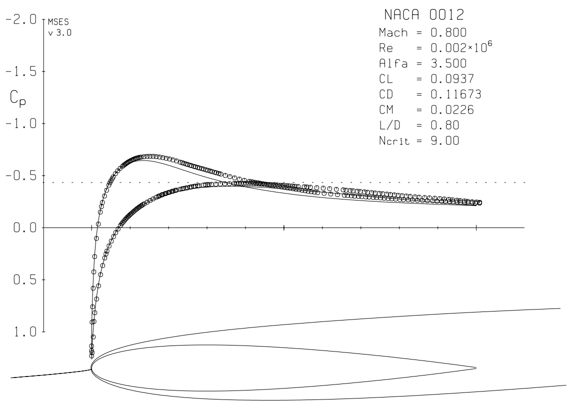

Figure 3: Surface C_p distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,$deg: Scheme II and MSES solutions

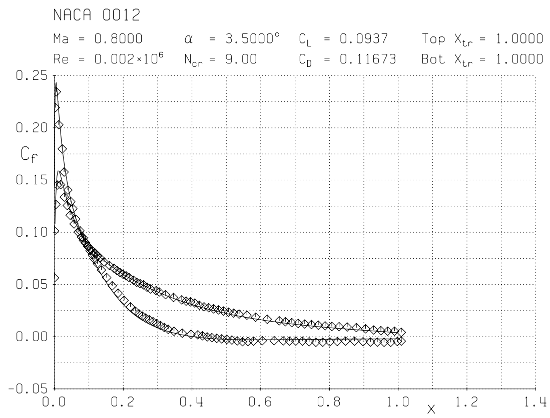

Figure 4: Surface C_f distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,$deg: Scheme II and MSES solutions

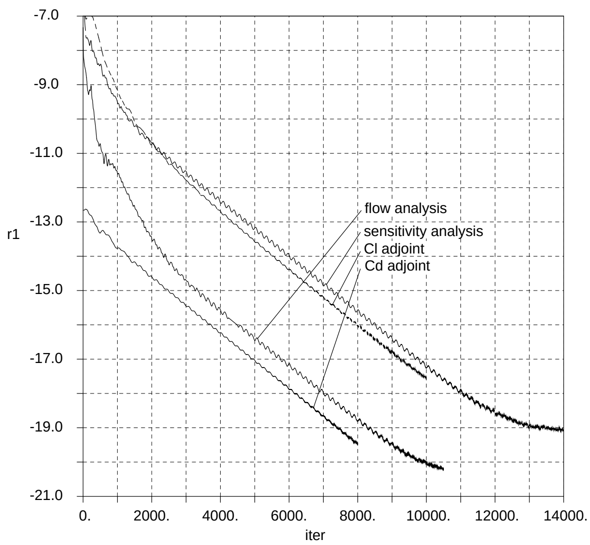

Figure 5: Convergence histories for NACA0012 airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,$deg

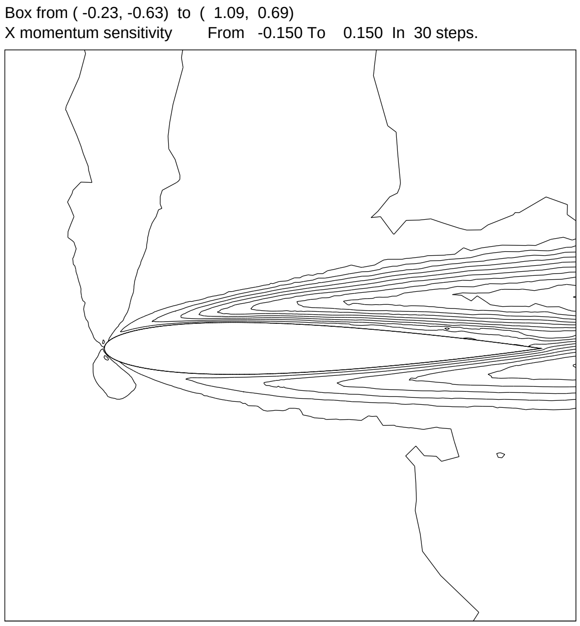

Figure 6: $\partial \rho u / \partial \alpha$ distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$ (Scheme II) found by direct method

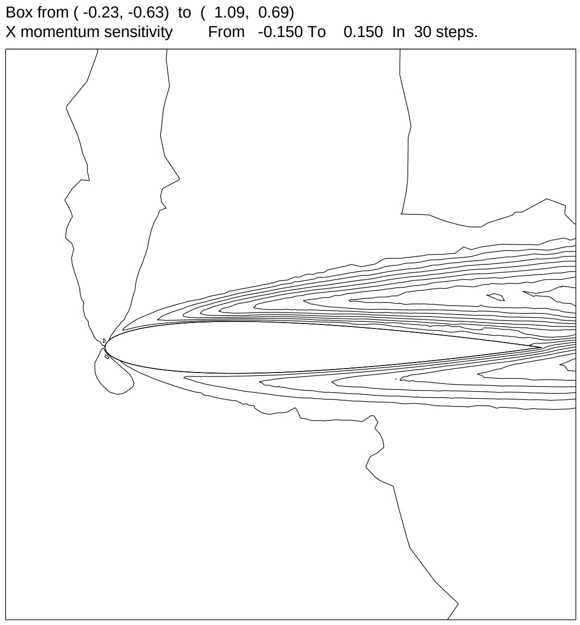

Figure 8: $\partial \rho u / \partial \alpha$ distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$ (Scheme II)from finite difference with $\Delta\alpha = 0.01$

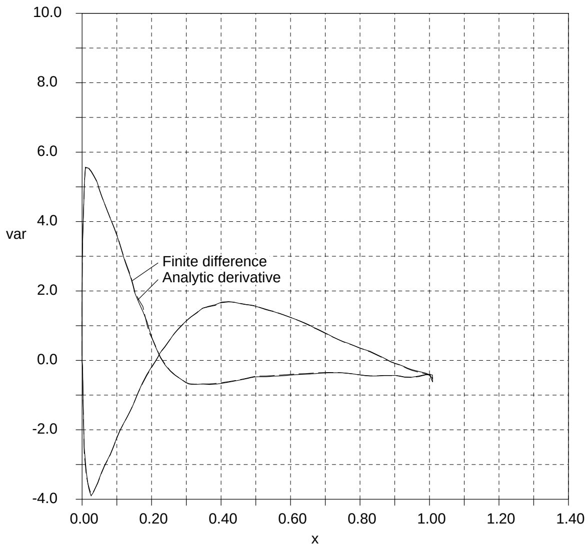

Figure 7: Surface $\partial p / \partial \beta_c$ distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$ (Scheme II v. Finite Difference)

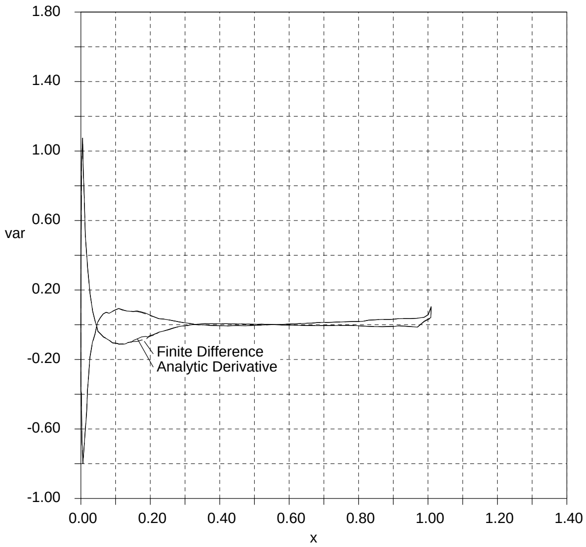

Figure 9: Surface $\partial \tau_{11} / \partial \beta_c$ distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$ (Scheme II v. Finite Difference)

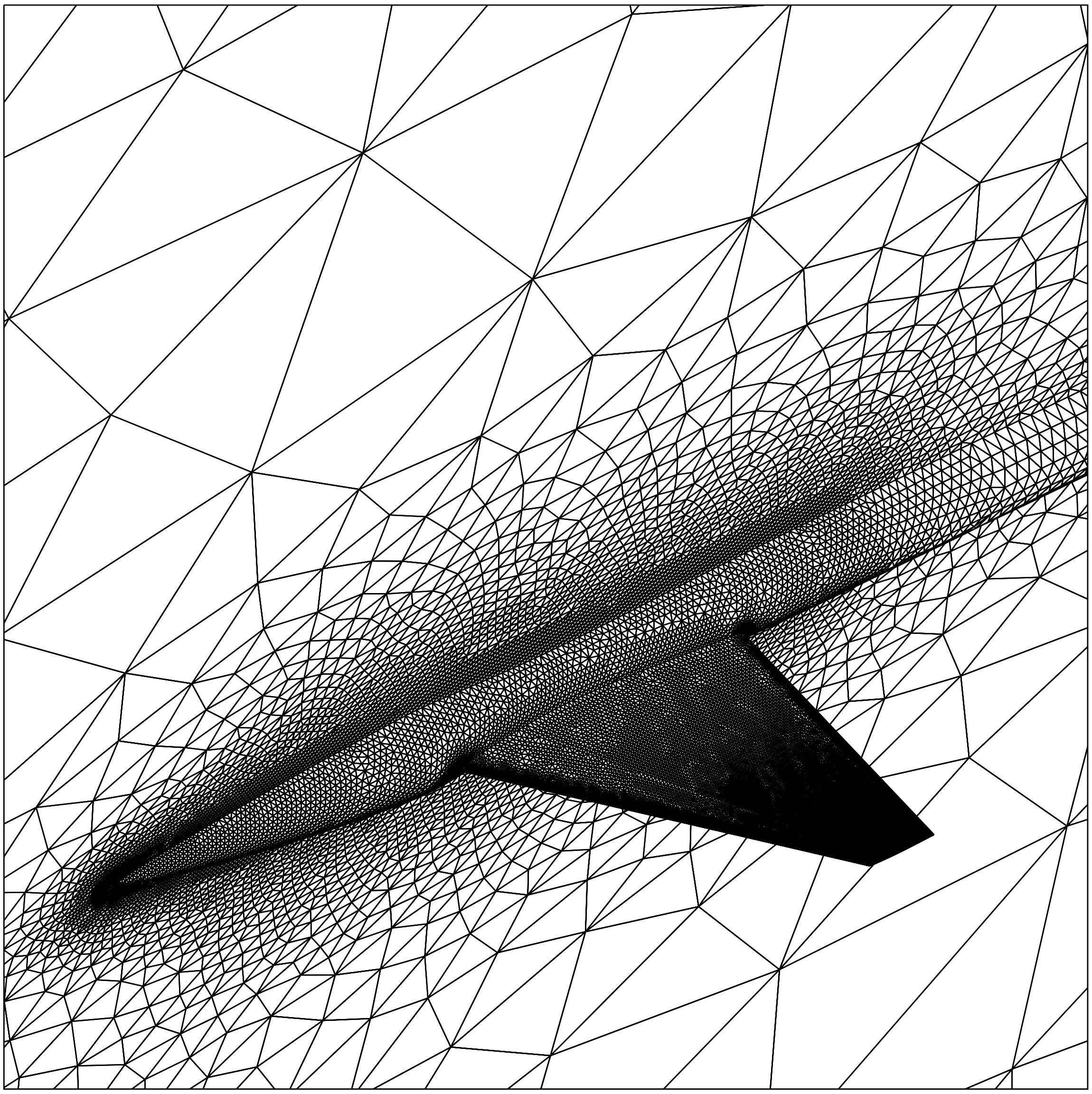

Figure 10: Baseline grid for 3D optimization exercise

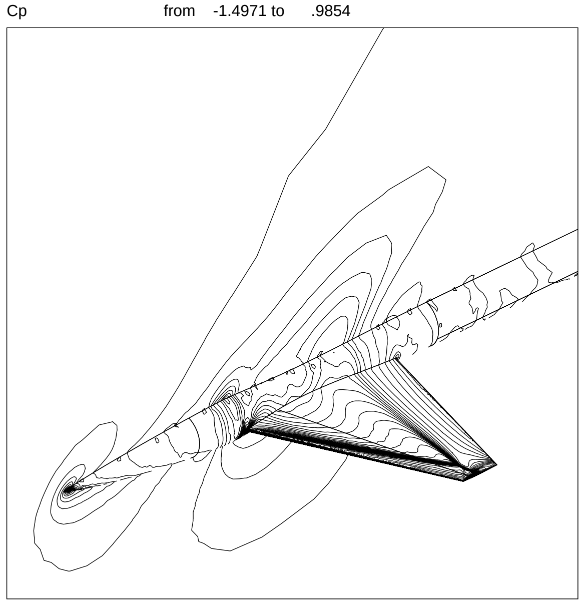

Figure 11: Condition 1 (M=0.9) baseline C_p distribution for 3D optimization exercise

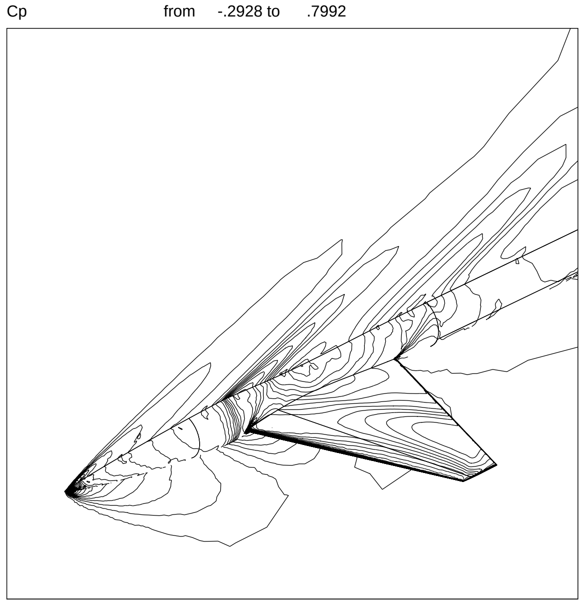

Figure 12: Condition 2 (M=1.6) baseline C_p distribution for 3D optimization exercise

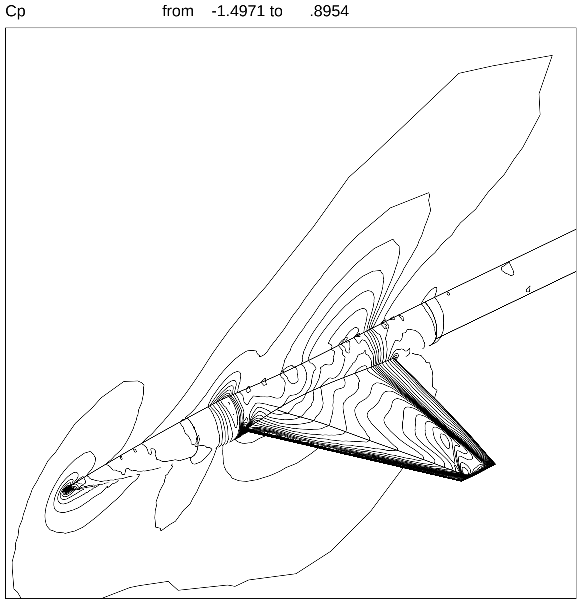

Figure 13: Condition 1 (M=0.9) “Design 7” C_p distribution for 3D optimization exercise

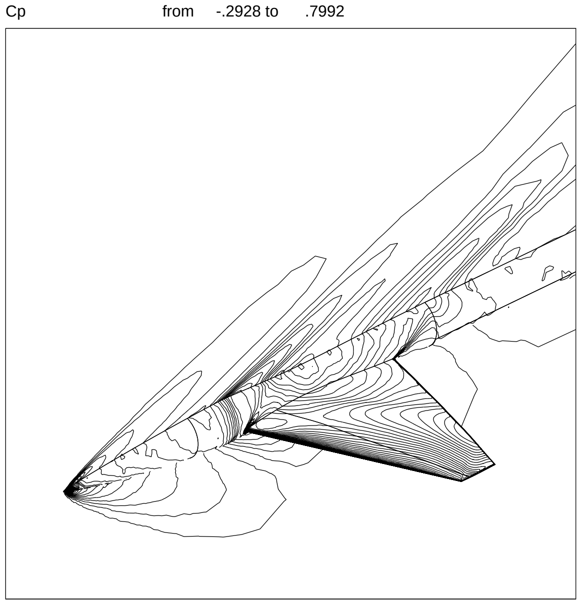

Figure 14: Condition 2 (M=1.6) “Design 7” C_p distribution for 3D optimization exercise

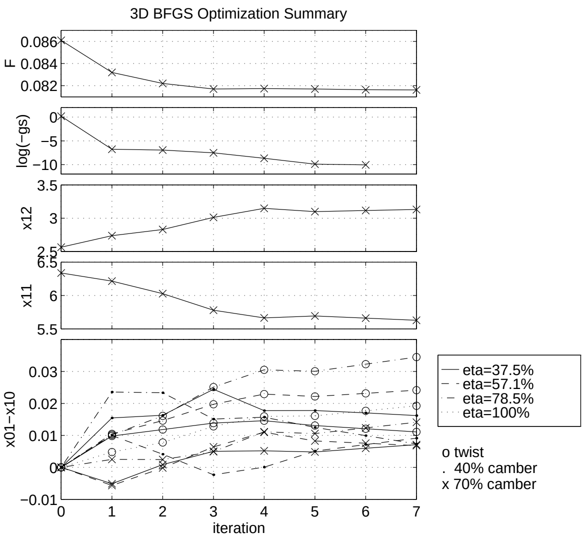

Figure 15: Cost function and design variable evolution for 2D BFGS optimization exercise

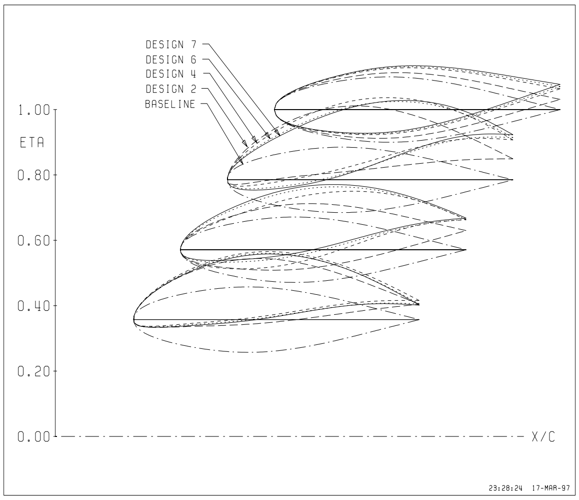

Figure 17: Geometry evolution for 3D BFGS optimization exercise

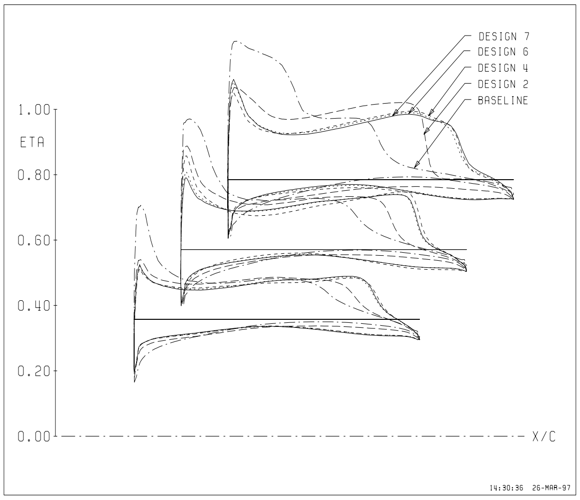

Figure 16: Condition 1 (M=0.9) C_p evolution for 3D optimization exercise

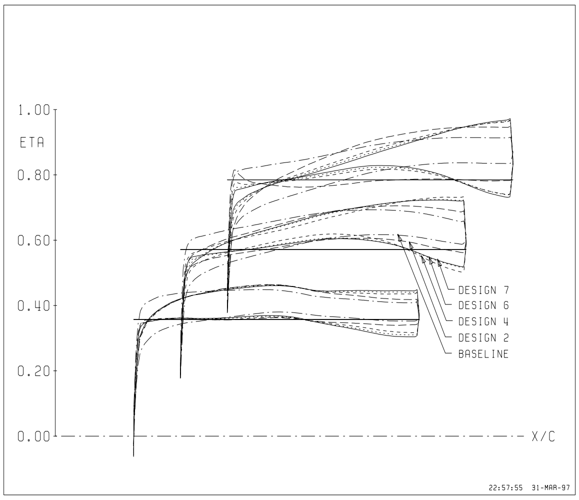

Figure 18: Condition 2 (M=1.6) C_p evolution for 3D optimization exercise

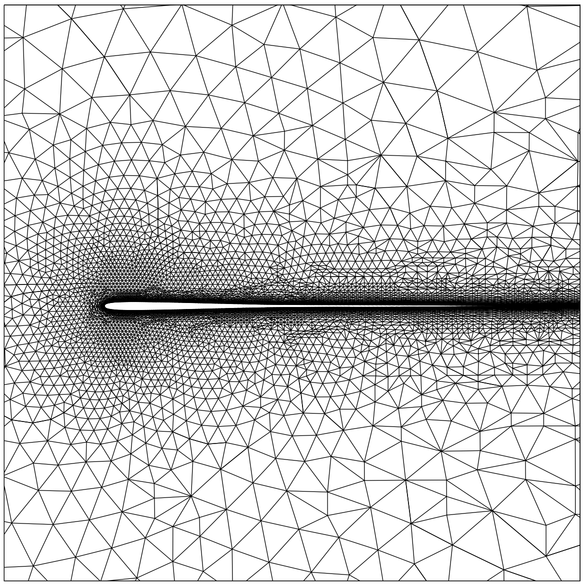

Figure 19: μ-UAV airfoil grid

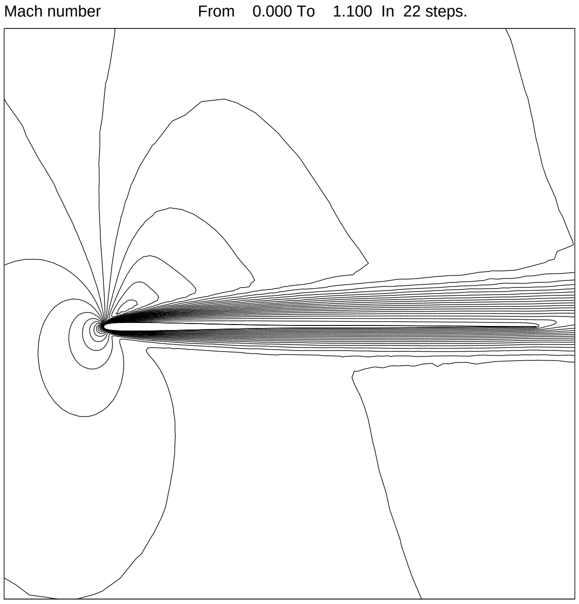

Figure 21: Baseline M distribution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$

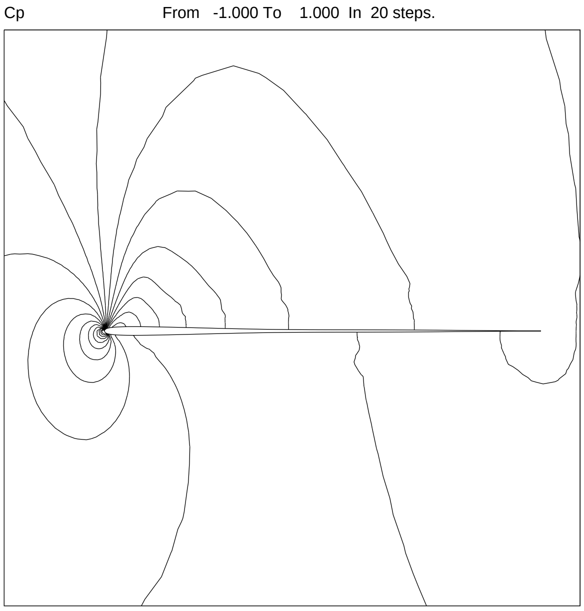

Figure 20: Baseline C_p distribution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$

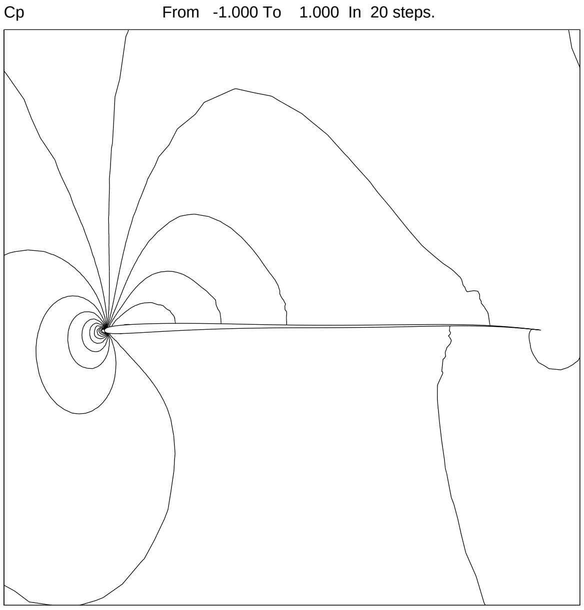

Figure 22: Final C_p distribution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\text{deg}$

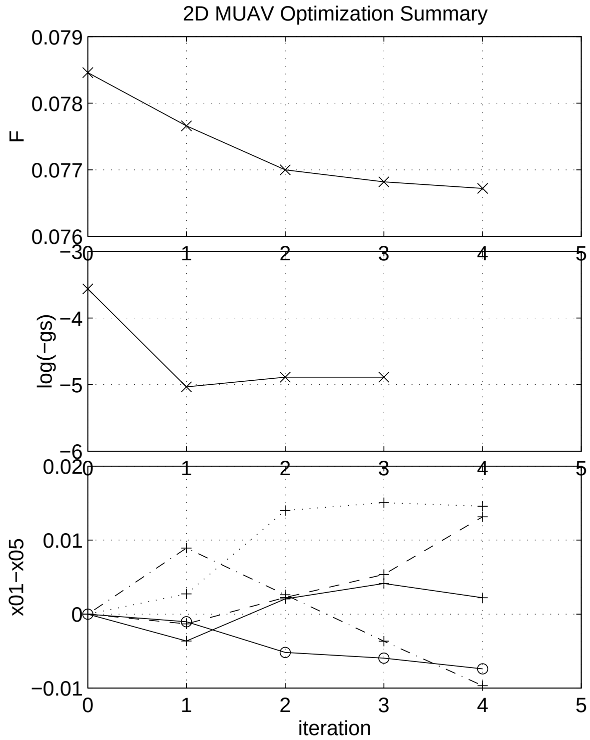

Figure 23: Design variable evolution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

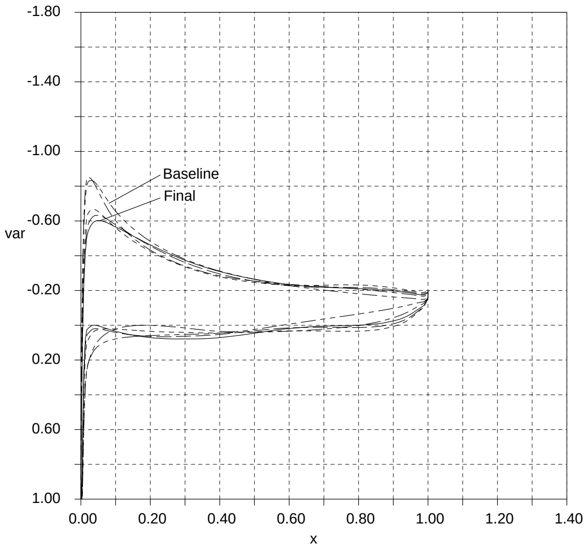

Figure 25: Surface C_p evolution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

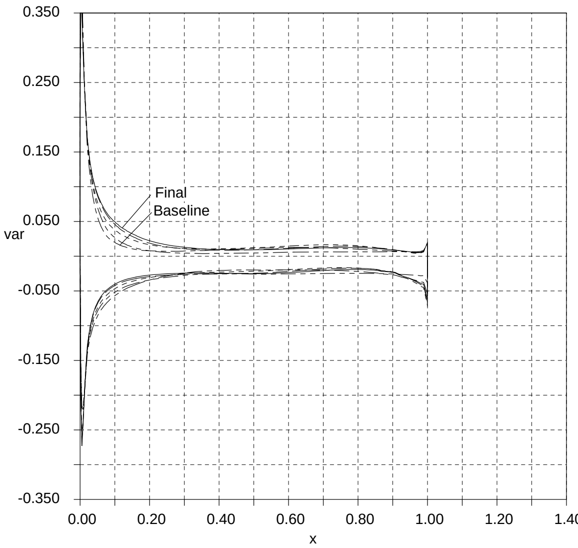

Figure 24: Surface C_f evolution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

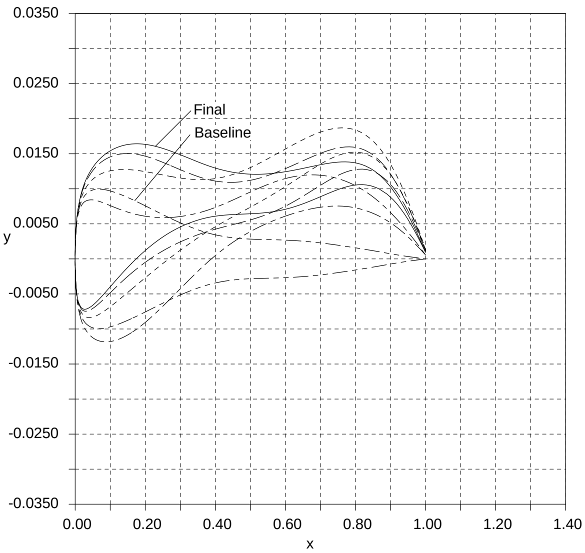

Figure 26: Geomety evolution for μ-UAV airfoil, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

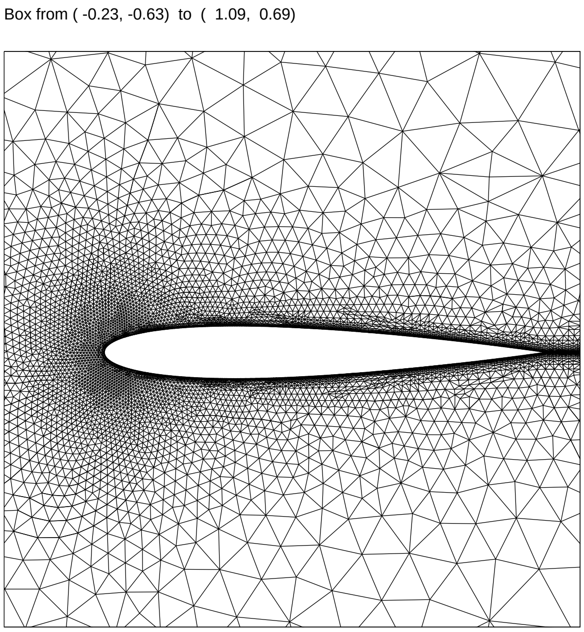

Figure 27: Re=2000 NACA0012 grid

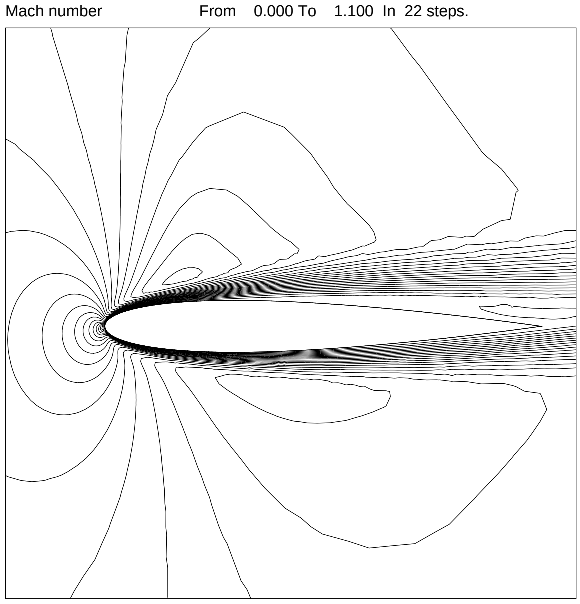

Figure 29: Baseline M distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

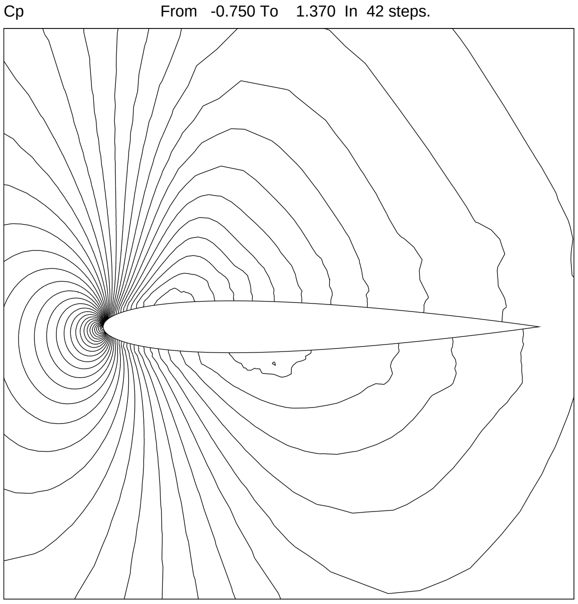

Figure 28: Baseline C_p distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

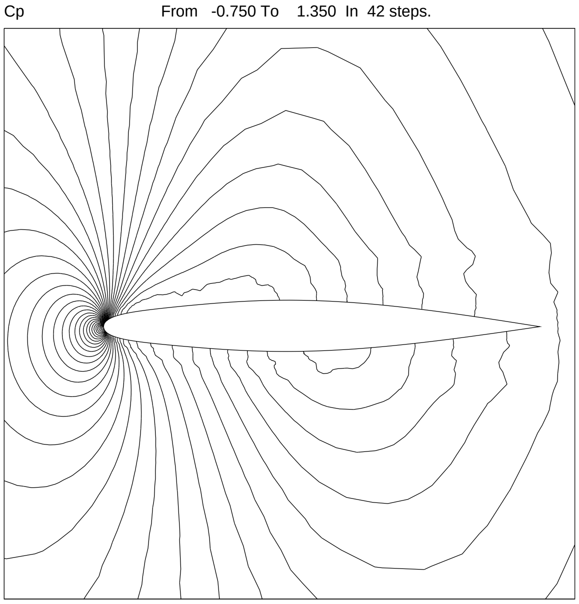

Figure 30: Final C_p distribution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

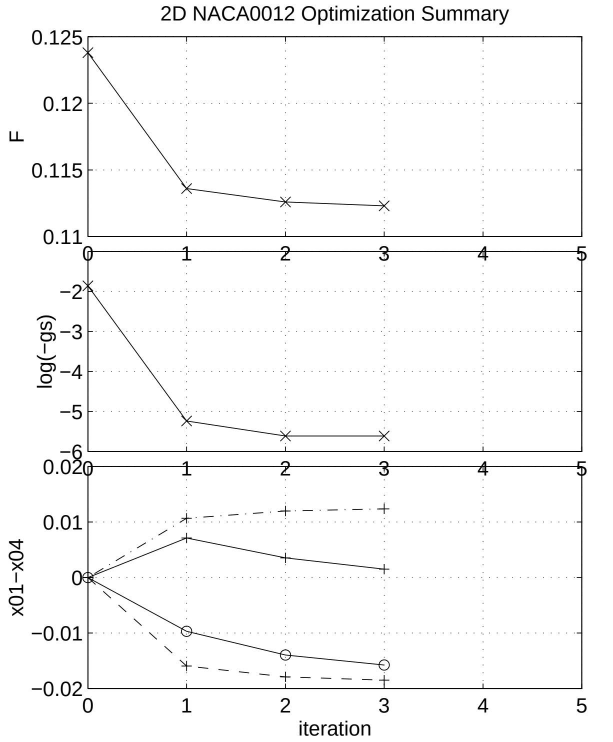

Figure 31: Design variable evolution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

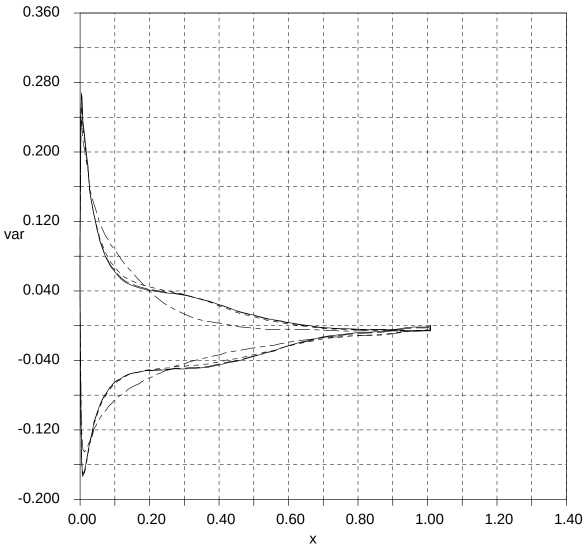

Figure 32: Surface C_f evolution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

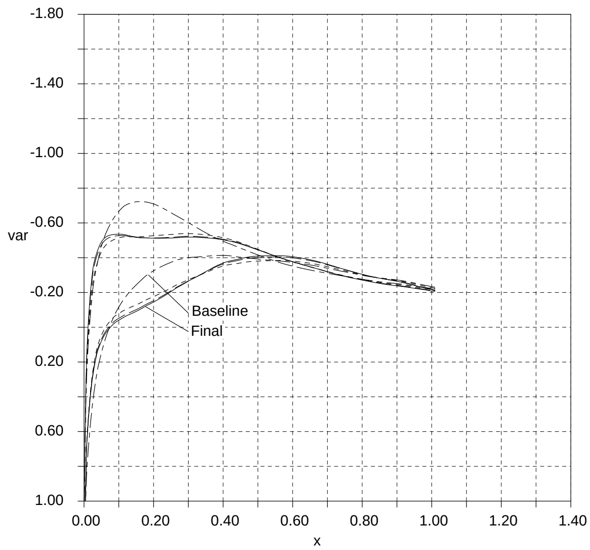

Figure 33: Surface C_p evolution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$

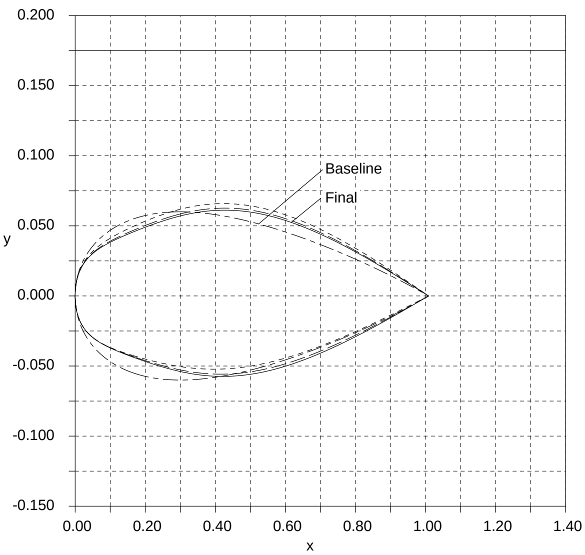

Figure 34: Geometry evolution for NACA0012, Re=2000, $M_\infty = 0.8$, $\alpha = 3.5\,\mathrm{deg}$