RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation

James Reuther, Antony Jameson, James Farmer, Luigi Martinelli and David Saunders

RIACS Technical Report 96.02   January 1996

Presented at the AIAA 34th Aerospace Sciences Meeting and Exhibit, January 1996,
AIAA paper 96-0094

Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation

James Reuther, Antony Jameson, James Farmer, Luigi Martinelli and David Saunders

The Research Institute of Advanced Computer Science is operated by Universities Space Research
Association, The American City Building, Suite 212, Columbia, MD 21044, (410) 730-2656

Work reported herein was sponsored by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation

J. Reuther[*]
Research Institute for Advanced Computer Science
NASA Ames Research Center, MS 227-6
Moffett Field, California 94035, U.S.A.

A. Jameson[†]
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544, U.S.A.

J. Farmer
Advanced Combustion and Engineering Research Center
Brigham Young University
75J CTB P.O. Box 24214 Provo Utah 84602, U.S.A.

L. Martinelli[‡]
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544, U.S.A.

D. Saunders
Sterling Software
NASA Ames Research Center, MS 227-6
Moffett Field, California 94035, U.S.A.

ABSTRACT

This work describes the implementation of optimization techniques based on control theory for complex aircraft configurations. Here control theory is employed to derive the adjoint differential equations, the solution of which allows for a drastic reduction in computational costs over previous design methods [13, 12, 43, 38]. In our earlier studies [19, 20, 22, 23, 39, 25, 40, 41, 42] it was shown that this method could be used to devise effective optimization procedures for airfoils, wings and wing-bodies subject to either analytic or arbitrary meshes. Design formulations for both potential flows and flows governed by the Euler equations have been demonstrated, showing that such methods can be devised for various governing equations [39, 25]. In our most recent works [40, 42] the method was extended to treat wing-body configurations with a large number of mesh points, verifying that significant computational savings can be gained for practical design problems. In this paper the method is extended for the Euler equations to treat complete aircraft configurations via a new multiblock implementation. New elements include a multiblock-multigrid flow solver, a multiblock-multigrid adjoint solver, and a multiblock mesh perturbation scheme. Two design examples are presented in which the new method is used for the wing redesign of a transonic business jet.

INTRODUCTION

To allow the full realization of the potential of Computational Fluid Dynamics (CFD) to produce superior designs, there is a need not only for accurate aerodynamic prediction methods for given configurations, but also for design methods capable of creating new optimum configurations. Yet, while flow analysis has matured to the extent that Navier-Stokes calculations are routinely carried out over very complex configurations, direct CFD based design is only just beginning to be used in the treatment of moderately complex three-dimensional configurations.

Existing CFD analysis methods can be used to treat the design problem by coupling them with numerical optimization methods. The essence of these methods, which may incur heavy computational expenses, is very simple: a numerical optimization procedure is used to extremize a chosen aerodynamic figure of merit which is evaluated by the given CFD code. The configuration is systematically modified through user specified design variables. Most of these optimization procedures require the gradient of the cost function with respect to changes in the design variables. The simplest of the methods to obtain these necessary gradients is the finite difference method. In this technique, the gradient components are estimated by independently perturbing each design variable with a finite step, calculating the corresponding value of the objective function using CFD analysis, and forming the ratio of the differences. The gradient is used by the numerical optimization algorithm to calculate a

[*] Student Member AIAA

[†] James S. McDonnell Distinguished University Professor of Aerospace Engineering, AIAA Fellow

[‡] AIAA Member

search direction using steepest descent, conjugate gradient, or quasi-Newton techniques. After finding the minimum or maximum of the objective function along the search direction, the entire process is repeated until the gradient approaches zero and further improvement is not possible.

The finite difference based optimization strategy is computationally expensive because the flow must be recalculated for perturbations in every design variable to determine the gradient. Nevertheless, it is attractive when compared with other traditional design strategies such as inverse methods, since it permits any choice of the aerodynamic figure of merit. The use of numerical optimization for transonic aerodynamic shape design was pioneered by Hicks, Murman and Vanderplaats [13]. They applied the method to two-dimensional profile design subject to the potential flow equation. The method was quickly extended to wing design by Hicks and Henne [12]. Later, in the work of Reuther, Cliff, Hicks and Van Dam, the method was successfully used for the design of supersonic wing-body transport configurations [38]. In all of these cases, finite difference methods were used to obtain the required gradient information.

Recently through work by both ourselves and other groups, alternative, less expensive methods for obtaining design sensitivities have been developed which greatly reduce the computational costs of optimization. The most promising of these emerging approaches is the adjoint formulation whereby the sensitivity with respect to an arbitrary number of design variables is obtained with the equivalent of only one additional flow calculation.

FORMULATION OF THE ADJOINT EQUATIONS

The aerodynamic properties which define the cost function I are functions of the flow field variables (w) and the physical location of the boundary, which may be represented by the function $\mathcal{F}$, say. Then

$$I = I(w, \mathcal{F})$$

and a change in $\mathcal{F}$ results in a change

$$\delta I = \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F} \tag{1}$$

in the cost function. The governing equation R and its first variation express the dependence of w and $\mathcal{F}$ within the flow field domain D:

$$R(w, \mathcal{F}) = 0, \quad \delta R = \left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F} = 0. \tag{2}$$

Next, introducing a Lagrange multiplier ψ, we have

$$\begin{aligned} \delta I &= \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F} - \psi^T\left(\left[\frac{\partial R}{\partial w}\right]\delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right]\delta \mathcal{F}\right) \\ &= \left\{\frac{\partial I^T}{\partial w} - \psi^T\left[\frac{\partial R}{\partial w}\right]\right\}\delta w + \left\{\frac{\partial I^T}{\partial \mathcal{F}} - \psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right]\right\}\delta \mathcal{F} \end{aligned}$$

Choosing ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T \psi = \frac{\partial I}{\partial w} \tag{3}$$

the first term is eliminated, and we find that the desired gradient is given by

$$\mathcal{G}^T = \frac{\partial I^T}{\partial \mathcal{F}} - \psi^T\left[\frac{\partial R}{\partial \mathcal{F}}\right]. \tag{4}$$

Since (4) is independent of δw, the gradient of I with respect to an arbitrary number of design variables can be determined without the need for additional flow field evaluations. The main cost is in solving the adjoint equation (3). In general, the adjoint problem is about as complex as a flow solution. If the number of design variables is large, it becomes compelling to take advantage of the cost differential between one adjoint solution and the large number of flow field evaluations required to determine the gradient by finite differences. Once equation (4) is obtained, $\mathcal{G}$ can be fed into any numerical optimization algorithm to obtain an improved design.

ISSUES OF IMPORTANCE FOR DESIGN PROBLEMS

The development of aerodynamic design procedures that employ an adjoint equation formulation is currently being investigated by many researchers. These methods promise to allow computational fluid dynamics methods to become true aerodynamic design methods. References [1, 2, 3, 5, 4, 7, 8, 6, 32, 29, 16, 35, 30, 28, 34, 45, 31, 36, 37, 47, 15, 33, 14] represent a partial list of recent works in this developing field. However, as is the case in any new research field, many questions remain. Probably the most salient issues of concern are the following:

1. Discrete vs. continuous sensitivities
2. Choice of optimization procedures
3. Treatment of geometric and aerodynamic constraints
4. The level of coupling between design and analysis
5. The parameterization of the design space

These topics still require further investigation. With regard to the first item, it is historically interesting that Jameson in 1988 [19] first developed the equations necessary for a continuous sensitivity approach to treat the design of airfoils and wings subject to transonic flows. This technique was later implemented both by our group and independently by Lewis and Agarwal [33]. By continuous sensitivities it is implied that the steps represented by equations (1)-(4) are applied to the governing differential equations. The adjoint differential equations with the appropriate boundary conditions may then be discretized and solved in a manner similar to that used for the flow solution algorithm. One may alternatively derive a set of discrete adjoint equations directly from the discrete approximation to the flow equations by following the procedure outlined in equations (1)-(4). The resulting discrete adjoint equations are one of the possible discretizations of the continuous adjoint equations. This alternative is mentioned in [20], but was not adopted in that work because of the complexity of the resulting discrete adjoint system. The approach has been favored by Taylor et al. [29, 16, 35, 30, 28] and Baysal et al. [1, 2, 3, 5, 4, 7, 8, 6, 32].

It seems that both alternatives have some advantages. The continuous approach gives the researcher some hope for an intuitive understanding of the adjoint system and its related boundary conditions. The discrete approach, in theory, maintains perfect algebraic consistency at the discrete level. If properly implemented, it will give gradients which closely match those obtained through finite differences. The continuous formulation produces slightly inaccurate gradients due to differences in the discretization. However, these inaccuracies are of comparable magnitude to the inaccuracies in the flow solution itself, and must vanish as the mesh width is reduced.

The discrete adjoint equations form a linear system, whether derived directly by the discrete approach or by discretizaton of the continuous adjoint equation. The size and complexity of the system, however, makes the use of direct solution methods unrealistic for all but the smallest problems. The application of the continuous sensitivity analysis fosters a straightforward recycling of the flow solution algorithm for the solution of the adjoint equations, since the steps applied to the original governing differential equations can be duplicated for the adjoint differential equations. When the discrete approach is used, the adjoint equations have a complexity which makes it hard to find decompositions to facilitate their solution unless the structure from the continuous adjoint is used as a guide. The discrete method is subject, moreover, to the difficulty that the discrete flow equations often contain nonlinear flux limiting functions which are not differentiable. It also limits the flexibility to use adaptive discretization techniques with order and mesh refinement, such as the $h - p$ method, because the adjoint discretization is fixed by the flow discretization. It is crucial for the success of a gradient method that the cost function depends continuously on the design variables. However, even though the true flow solution depends continuously on the design variables, the use of adaptive discretization or nonlinear flux limiters may cause the discrete solution to cease to depend continuously on the design variables because of sudden changes in the discretization.

It turns out that the determination of items 2-4 in the above list strongly hinge on the choice for 5. In Jameson's first works in the area [19, 20, 22], every surface mesh point was used as a design variable. In three-dimensional wing design cases this led to as many as 4,224 design variables [22]. The use of the adjoint method eliminated the unacceptable costs that such a large number of design variables would incur for traditional finite difference methods. If the approach were extended to treat complete aircraft configurations, at least tens of thousands of design variables would be necessary. Such a large number of design variables also precludes the use of descent algorithms such as Newton or quasi-Newton approaches simply because of the high cost of matrix operations for such methods. The use of a simple descent procedure, such as steepest descent, has the advantage, moreover, that significant errors can be tolerated in the gradient evaluation during the early steps. Such methods therefore favor tighter coupling of the flow solver, the adjoint solver, and the overall design problem to accelerate convergence. Ta'asan et al. [31, 45] have taken advantage of this by formulating the design problem as a "one shot" procedure where all three systems are advanced simultaneously. The use of such a design space can lead to poorly conditioned design problems. This can be illustrated by the case where only one point on the surface of the geometry is moved, resulting in a highly nonlinear design response. In his original work, Jameson [20, 21] addressed this poor conditioning of the design problem by smoothing the control (surface shape) and thus attenuating the high frequency content in the developing solution shape.

Hicks and others [13, 12, 43, 38] have in the past parameterized the design space using sets of smooth functions that perturb the initial geometry. By using such a parameterization it is possible to work with considerably fewer design variables than the choice of every mesh point. And since Hicks' original works exclusively used finite difference gradients, the inherent restriction to a small number of design variables allowed for the use of more efficient search strategies. One simple choice of design variables for airfoils suggested by Hicks and Henne [12] has the following "sine bump" form:

$$b(x) = \left[\sin\left(\pi x^{\frac{\log(.5)}{\log(t_1)}}\right)\right]^{t_2}, \quad 0 \leq x \leq 1 \qquad (5)$$

Here t_1 locates the maximum of the bump in the range $0 \leq x \leq 1$ at $x = t_1$, since the maximum occurs when $x^\alpha = \frac{1}{2}$, where $\alpha = \log \frac{1}{2} / \log t_1$, or $\alpha \log t_1 = \log \frac{1}{2}$. Parameter t_2 controls the width of the bump.

When distributed over the entire configuration, such analytic perturbation functions admit a large reachable design space. They can be chosen such that symmetry, thickness, or volume can be explicitly constrained, thus avoiding the use of elaborate constrained optimization algorithms to impose geometric constraints. Further, particular choices of these variables will concentrate the design effort in regions where refinement is needed, while leaving the rest of the geometry virtually undisturbed. The disadvantage of these functions is that they are not orthogonal, and there is no simple way to form a basis from these functions which is complete for the space of continuous functions which vanish at $x = 0$ and $x = 1$. Thus, they do not guarantee that a solution, for example, of the inverse problem for a realizable target pressure distribution will be attained. Nevertheless, they have proved to be quite effective in realizing design improvements with a limited number of design variables. The design process that uses these basis spaces, can be accelerated toward convergence either by tighter coupling of the individual design elements, as was the case when using the mesh points themselves, or through the use of higher order optimization algorithms. Finally, they have the advantage that there is no need to smooth the resulting solutions as the design proceeds, since by construction higher frequencies are not admitted, and thus the design spaces are naturally well posed.

Another approach is the use of B-spline control points as design variables [5, 47, 15]. These in turn define B-spline curves and surfaces which represent the geometric configurations. Like the Hicks-Henne functions, B-splines allow for a greatly reduced number of design variables, and thus permit the use of, say, a quasi-Newton design procedure. If for example the upper and lower surfaces of an airfoil are treated separately, the method admits camber or thickness constraints explicitly within the design space. Further, local control is also possible by choosing only a limited number of control points as active design variables. This method in practice seems to have an advantage over the Hicks-Henne functions in that a more complete basis space of admitted designs is permitted for a given number of design variables. Finally, since these curves and surfaces are the natural entities used in CAD environments, they provide a straightforward way of integrating CAD and aerodynamic design. Despite these advantages, it was shown in our recent study [41] that, in contrast to Hicks-Henne functions, B-spline control points have a tendency to produce wavy solutions in problems that require a large number of control points. This difficulty can be attributed to the fact that they admit high frequency components in the design variations, rather like those which are encountered when the mesh points are used as design variables. A possible solution to this difficulty would be to use the same type of implicit smoothing procedure that proved effective when the shape was defined by the mesh points.

MULTIBLOCK FLOW SOLUTION

The extension of the methods presented in our earlier works in three-dimensions, such that they may treat complete aircraft configurations, requires the replacement of the single block flow solver used in References [22, 40, 42]. In order to use CFD in an automated design environment, the flow solver must meet fundamental requirements of accuracy, efficiency, and robust convergence. High accuracy is required since the predicted improvements in the design realized by the method can only be as good as the accuracy of the flow analysis. Efficiency of the flow solver is also critical since the optimization of the design will generally require the computation of many flow solutions or other solutions of comparable complexity. The last aspect, robust convergence, is also of significant importance. In highly refined aerodynamic design applications, the main benefit of aerodynamic optimization is in obtaining the last few percentage points in improved efficiency. In such cases the solutions must be highly converged so that the noise in the figure of merit, say of drag at a fixed lift, is well below the level of realizable improvement. Thus in contrast to flow analysis where 3 orders of magnitude convergence in RMS residual is usually considered adequate, a flow solver used in design applications must be typically able to converge 7 orders in RMS residual.

In our three-dimensional single block applications the FLO87 code written by the second author easily met all of the above criteria. FLO87 achieves fast convergence with the aid of multigridding and residual smoothing. It is normally easy to obtain solutions that converge to machine accuracy. The challenge in the present work was to meet these strict convergence requirements within the framework of a multiblock flow solver.

The general strategy in developing the multiblock flow solver is to construct and update a halo around each block such that the flow solution inside each block is transparent to the block boundaries. This task requires establishing the size and location of halo cells adjacent to block boundaries, and loading the halo cell values with appropriate flow field data at the appropriate time. To accomplish this task, a two-level halo is constructed around each block. At interior boundaries where two or more blocks meet, the values of the state vectors in the halo cells are identical to the values in the internal cells from adjacent blocks. Halo cells on the external boundary of the entire computational domain are constructed and updated by extrapolation and reflection. Once the halo configuration is set up for each block, standard methods for spatial discretization and time integration (including artificial dissipation, residual averaging, and multigridding) are employed to compute the flow solution within each individual block.

The strategy for a complete flow solution proceeds as follows: First, the blocks that comprise the flow field mesh are read from an external file. Then, the double halo configuration is established, for each individual block, by inserting into halo cell locations values for grid metrics, etc., taken from the interior cells of adjacent blocks. For the coarse grids required in the multigrid procedure, the process is repeated with coarse grid halo cells defined by the internal cells of adjacent coarse grid blocks. For block faces that lie on solid, symmetry or far field boundaries, standard single-block techniques are used to define the halo cells. As an example, consider the simple 4-block grid depicted in Figure 1. The halo cells for block I will be obtained from the internal cells of blocks II, III, and IV, and from solid or far field boundary techniques for the faces not adjacent to other blocks. Coarse grids are computed in the usual fashion, by aggregating groups of eight cells and then repeating the above halo cell process. Once the halo configuration is complete for the fine and all coarse grids, the flow solution commences.

The system of equations solved here as well as the solution strategy follows that presented in many earlier works [26, 18, 17]. The three-dimensional Euler equations may be written as

$$\frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial x_i} = 0 \quad \text{in } D, \tag{6}$$

where it is convenient to denote the Cartesian coordinates and velocity components by x_1, x_2, x_3 and u_1, u_2, u_3, and w and f_i are defined as

$$w = \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{array} \right\}, \quad f_i = \left\{ \begin{array}{c} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ \rho u_i H \end{array} \right\} \tag{7}$$

with δ_{ij} being the Kronecker delta function. Also,

$$p = (\gamma - 1)\rho \left\{ E - \frac{1}{2}\left(u_i^2\right) \right\}, \tag{8}$$

and

$$\rho H = \rho E + p \tag{9}$$

where γ is the ratio of the specific heats. Consider a transformation to coordinates ξ_1, ξ_2, ξ_3 where

$$K_{ij} = \left[\frac{\partial x_i}{\partial \xi_j}\right], \quad J = \det(K), \quad K_{ij}^{-1} = \left[\frac{\partial \xi_i}{\partial x_j}\right].$$

Introduce scaled contravariant velocity components as

$$U_i = Q_{ij} u_j$$

where

$$Q = JK^{-1}.$$

The Euler equations can now be written as

$$\frac{\partial W}{\partial t} + \frac{\partial F_i}{\partial \xi_i} = 0 \quad \text{in } D, \tag{10}$$

with

$$W = J \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{array} \right\}, \quad F_i = Q_{ij} f_j = \left\{ \begin{array}{c} \rho U_i \\ \rho U_i u_1 + Q_{i1} p \\ \rho U_i u_2 + Q_{i2} p \\ \rho U_i u_3 + Q_{i3} p \\ \rho U_i H \end{array} \right\}. \tag{11}$$

For the multiblock case, the above notation applies to each block in turn. The flow is thus determined as the steady state solution to equation (10) in all blocks subject to the flow tangency conditions on all solid boundary faces of all blocks:

$$U_\eta = 0 \quad \text{on all } B_S \tag{12}$$

where η is 1, 2, or 3 depending on the direction that is normal to face B_S where a solid surface is assumed to exist. At the far field boundary faces, B_F, freestream conditions are specified for incoming waves, while outgoing waves are determined by the solution.

The time integration scheme follows that used in the single block strategy [26]. The solution proceeds by performing the cell flux balance, updating the flow variables, and smoothing the residuals, at each stage of the time stepping scheme and each level of the multigrid cycle. The main difference in the integration strategy is the need to loop over all blocks during each stage of the integration process. The use of the double-halo configuration permits standard single-block subroutines to be used, without modification, for the computation of the flow field within each individual block. This includes the single-block subroutines for convective and dissipative flux discretization, multistage time stepping, and multigrid convergence acceleration.

The only difference between the integration strategies is in the implementation of the residual averaging. In the single-block solution strategy, a tridiagonal system of equations is set up and solved using flow information from the entire grid. Thus, each residual is replaced by an average of itself and the residuals of the entire grid. In the multiblock strategy, the support for the residual smoothing is reduced to the size of each block, in order to eliminate the need to solve tridiagonal systems spanning the blocks, which would incur a penalty in communication costs. This change has no effect on the final converged solution, and in the present application has not led to any significant reduction in the rate of convergence.

THE ADJOINT FORMULATION FOR THE EULER EQUATIONS

The application of control theory to aerodynamic design problems is illustrated by treating the case of three-dimensional design, using the Euler equations discussed above as the mathematical model for compressible flow. In our previous works, the illustrative problem most often used specified the cost function as a measure of the difference between the current and some desired pressure distribution. For variety, the development here will use drag at a fixed lift as the cost function.

$$\begin{aligned} I = \quad & C_D \\ = \quad & C_A \cos\alpha + C_N \sin\alpha \\ = \quad & \frac{1}{\mathbf{S_{ref}}} \iint_{B_S} C_P \left(S_x \cos\alpha + S_y \sin\alpha \right) d\xi_1 \, d\xi_2, \end{aligned}$$

where S_x and S_y define projected surface areas, $\mathbf{S_{ref}}$ is the reference area, and $d\xi_1$ and $d\xi_2$ are the two coordinate indices that are in the plane of the face in question. Note that the integral in the final expression above is integrated over all solid boundary faces. The design problem is now treated as a control problem where the control function is the geometry shape, which is to be chosen to minimize I, subject to the constraints defined by the flow equations (6–11). A variation in the shape will cause a variation δp in the pressure and consequently a variation in the cost function

$$\begin{aligned} \delta I = \quad & \tilde{\delta} C_A \cos\alpha + \tilde{\delta} C_N \sin\alpha \\ + \quad & \left\{ -C_A \sin\alpha + C_N \cos\alpha \right\} \delta\alpha \\ + \quad & \left\{ \frac{\partial C_A}{\partial\alpha} \cos\alpha + \frac{\partial C_N}{\partial\alpha} \sin\alpha \right\} \delta\alpha \end{aligned}$$

where $\tilde{\delta} C_A$ and $\tilde{\delta} C_N$ are variations due to changes in the design parameters with α fixed. To treat the interesting problem of practical design, drag must be minimized at a fixed lift coefficient. Thus an additional constraint is given by

$$\delta C_L = 0,$$

which gives

$$\begin{aligned} & \tilde{\delta} C_N \cos\alpha - \tilde{\delta} C_A \sin\alpha \\ + \quad & \left\{ -C_N \sin\alpha - C_A \cos\alpha \right\} \delta\alpha \\ + \quad & \left\{ \frac{\partial C_N}{\partial\alpha} \cos\alpha - \frac{\partial C_A}{\partial\alpha} \sin\alpha \right\} \delta\alpha = 0 \end{aligned}$$

Combining these two expressions to eliminate $\delta\alpha$ gives

$$\begin{aligned} \delta I \quad & = \tilde{\delta} C_A \cos\alpha + \tilde{\delta} C_N \sin\alpha \\ & + \Omega \left\{ \tilde{\delta} C_N \cos\alpha - \tilde{\delta} C_A \sin\alpha \right\}, \end{aligned} \tag{13}$$

where Ω is given by

$$\Omega = \frac{\left(C_L + \frac{\partial C_A}{\partial\alpha} \cos\alpha + \frac{\partial C_N}{\partial\alpha} \sin\alpha \right)}{\left(C_D - \frac{\partial C_N}{\partial\alpha} \cos\alpha + \frac{\partial C_A}{\partial\alpha} \sin\alpha \right)}.$$

Since p depends on w through the equation of state (8–9), the variation δp can be determined from the variation δw. If a fixed computational domain is used, the variations in the shape result in variations in the mapping derivatives. Define the Jacobian matrices

$$A_i = \frac{\partial f_i}{\partial w}, \quad C_i = Q_{ij} A_j. \tag{14}$$

Then the equation for δw in the steady state becomes

$$\frac{\partial}{\partial \xi_i} \left(\delta F_i \right) = 0,$$

where

$$\delta F_i = C_i \delta w + \delta \left(Q_{ij} \right) f_j.$$

Now, multiplying by a vector co-state variable ψ, assuming the result is differentiable, and integrating by parts over the entire domain,

$$\int_D \left(\frac{\partial \psi^T}{\partial \xi_i} \delta F_i \right) d\xi_j = \int_B \left(\bar{n}_i \psi^T \delta F_i \right) d\xi_j, \tag{15}$$

where $\bar{n}_i$ are components of a unit vector normal to the boundary. The variation in the cost function can also be expressed in terms of δp after (13) and (15) are summed to give,

$$\begin{aligned} \delta I = \quad & \frac{1}{\frac{1}{2}\gamma M_\infty^2 \mathbf{S_{ref}}} \iint_{B_S} \delta p \left\{ \left(S_x \cos\alpha + S_y \sin\alpha \right) \right. \\ & \qquad \left. + \Omega \left(S_y \cos\alpha - S_x \sin\alpha \right) \right\} d\xi_1 \, d\xi_2 \\ + \quad & \frac{1}{\mathbf{S_{ref}}} \iint_{B_S} C_p \left\{ \left(\delta S_x \cos\alpha + \delta S_y \sin\alpha \right) \right. \\ & \qquad \left. + \Omega \left(\delta S_y \cos\alpha - \delta S_x \sin\alpha \right) \right\} d\xi_1 \, d\xi_2 \\ & - \int_D \left(\frac{\partial \psi^T}{\partial \xi_i} \delta F_i \right) d\xi_j + \int_B \left(\bar{n}_i \psi^T \delta F_i \right) d\xi_j. \end{aligned} \tag{16}$$

On the solid surfaces B_S, $\bar{n}_1 = \bar{n}_2 = 0$. It follows from equation (12) that

$$\delta F_\eta = \left\{ \begin{array}{c} 0 \\ Q_{\eta 1}\delta p \\ Q_{\eta 2}\delta p \\ Q_{\eta 3}\delta p \\ 0 \end{array} \right\} + p \left\{ \begin{array}{c} 0 \\ \delta\left(Q_{\eta 1}\right) \\ \delta\left(Q_{\eta 2}\right) \\ \delta\left(Q_{\eta 3}\right) \\ 0 \end{array} \right\} \text{ on any } B_S. \tag{17}$$

Suppose now that ψ is the steady state solution of the adjoint equation

$$\frac{\partial \psi}{\partial t} - C_i^T \frac{\partial \psi}{\partial \xi_i} = 0 \quad \text{in } D. \tag{18}$$

At internal block boundaries, the face integrals cancel from the adjacent blocks. At the far field the choice of the adjoint boundary conditions depends on whether the flow is subsonic or supersonic. For subsonic flow, so long as the outer domain is very far from the configuration of interest, we may set

$$\psi_{1-5} = 0 \quad \text{on all} \quad B_F.$$

If however the flow is subsonic and the boundary is fairly close, then far field faces may be set by $\psi_{1-5} = 0$ for incoming waves, while outgoing waves are determined by the solution. It is noted that the waves in the adjoint problem propagate in the opposite direction to those in the flow problem due to the transpose in equation (18). For supersonic flows, the choice of boundary conditions at the outer domain can be developed from physical intuition as well as mathematics. For a given geometry, say a wing, a change in the surface at any particular point, $\mathcal{P}$, will incur changes in the flow field and hence the performance in the region defined by the Mach cone originating at $\mathcal{P}$. Similarly, it is possible to determine the region over which surface changes affect the flow condition at a given point. This region would also form a cone that would point roughly in the opposite direction of the Mach cone, depending on local conditions. It is the solution of this reverse problem that the adjoint represents. The contribution to, say, drag at a given point is influenced by changes to the surface at all points within the reverse cone. The correct supersonic far field boundary conditions for the adjoint equation that are consistent with this reversed character are:

$$\begin{array}{lll} \psi_{1-5} & = & 0 \quad \text{at the exit;} \\ \psi_{1-5} & & \text{extrapolated from the interior at the entrance.} \end{array}$$

Then if the coordinate transformation is such that $\delta\left(JK^{-1}\right)$ is negligible in the far field, the last integral in (16) reduces to

$$-\int\!\!\int_{B_S} \psi^T \delta F_\eta \, d\xi_1 d\xi_2. \tag{19}$$

Thus by letting ψ satisfy the boundary conditions,

$$\left(\psi_2 Q_{\eta 1} + \psi_3 Q_{\eta 2} + \psi_4 Q_{\eta 3}\right) = \mathcal{Q} \quad \text{on all } B_S, \tag{20}$$

where

$$\begin{aligned} \mathcal{Q} = \; & \frac{1}{\frac{1}{2}\gamma M_\infty^2 \mathbf{S_{ref}}} \left\{ \left(S_x \cos\alpha + S_y \sin\alpha\right) \right. \\ & \left. + \Omega\left(S_y \cos\alpha - S_x \sin\alpha\right) \right\}, \end{aligned}$$

we find after integrating by parts again that

$$\begin{aligned} \delta I = \; & \frac{1}{\mathbf{S_{ref}}} \int\!\!\int_{B_S} C_p \left\{ \left(\delta S_x \cos\alpha + \delta S_y \sin\alpha\right) \right. \\ & \left. + \Omega\left(\delta S_y \cos\alpha - \delta S_x \sin\alpha\right) \right\} d\xi_1 d\xi_2 \\ & + \int_D \psi^T \frac{\partial}{\partial \xi_i} \left(\delta Q_{ij} f_j\right) d\xi_k. \end{aligned} \tag{21}$$

MULTIBLOCK MESH VARIATIONS AND DESIGN VARIABLES

In order to construct δI in equation (21), the variation in the metric terms must be obtained in each block. One way to accomplish this is to use finite differences to calculate the necessary information. This approach avoids the use of multiple flow solutions to determine the gradient, but it unfortunately still requires the mesh generator to be used repeatedly. The number of mesh solutions required is proportional to the number of design variables. The inherent difficulty in the approach is two-fold. First, for complicated three-dimensional configurations, elliptic or hyperbolic partial differential equations must often be solved iteratively in order to obtain acceptably smooth meshes. These iterative mesh generation procedures are often computationally expensive. In the worst case they approach the cost of the flow solution process. Thus the use of finite difference methods for obtaining metric variations in combination with an iterative mesh generator leads to computational costs which strongly hinge on the number of design variables, despite the use of an adjoint solver to eliminate the flow variable variations. Second, multiblock mesh generation is by no means a trivial task. In fact no method currently exists that allows this to be accomplished as a completely automatic process for complex three-dimensional configurations.

In our earlier works [40, 39, 25, 19, 20, 22], two methods have been explored which avoid these difficulties. In the first method, a completely analytic mapping procedure was used for the mesh generation. This technique is not only fully automatic and results in smooth consistent meshes, but it also allows for complete elimination of finite difference information for the mesh metric terms. Since the mapping function fully determines the entire mesh based on the surface shape, this analytic relationship may be directly differentiated in order to obtain the required information without considering a finite step. An analytic mapping method requires the geometry topology to be built directly into the formulation, and only works for simple configurations. Nevertheless, within these limitations it has proven to be highly effective [19, 20, 22].

The second method that we have explored is the use of an analytic mesh perturbation technique. In this approach, a high quality mesh appropriate for the flow solver is first generated by any available procedure prior to the start of the design. In examples to be shown later, these meshes were created using the Gridgen software developed by the company Pointwise [44]. This initial mesh becomes the basis for all subsequent meshes which are developed by analytical perturbations. In the method that was previously developed for wing-body configurations it had been assumed that only one surface, say the wing, was perturbed during a design case. This permitted the use of a very simple algebraic mesh perturbation algorithm. New meshes are created by moving all the mesh points on an index line projecting from the surface by an amount which is attenuated as the arc length from the surface increases. If the outer boundary of the grid domain is held constant the modification to the grid has the form

$$x_i^{new} = x_i^{old} + \mathcal{S}^{old}\left(x_{s_i}^{new} - x_{s_i}^{old}\right) \tag{22}$$

where x_i represents the volume grid points, x_{s_i} represents the surface grid points and $\mathcal{S}$ represents the arc length along the radial mesh line measured from the outer domain, normalized so that $\mathcal{S} = 1$ at the inner surface. Unfortunately this simple logic breaks down in the case where multiple faces sharing common edges are allowed to move. Thus in order to use analytic mesh perturbations for the treatment of the more general problem where multiple faces of a given block may be simultaneously deformed, equation (22) had to be modified in a way that resembles transfinite interpolation (TFI) [46]. Unlike TFI, where there is no prior knowledge of the interior mesh, the perturbation algorithm developed here (WARP3D) does make use of the relative interior point distributions in the initial mesh.

WARP3D may be thought of as a two stage procedure that operates within each block. The first stage shifts the internal mesh points to produce a reference block that is determined entirely by the new locations of the 8 corner points defining the block. Corresponding to the motion of each corner point, each interior point is shifted by a displacement proportional to one minus the normalized distance along the index lines away from that corner point. The second stage checks the perturbation of each point in all six faces relative to the position of the corresponding point in the reference block. If the perturbation of the domain involves a simple translation of all boundary points, these relative changes of the face points will be zero and all the perturbation will be accomplished by the first stage. If, however, face points are perturbed relative to the reference block, then these changes are propagated to the interior points through relative arc length-based perturbations along projecting index lines as described in the original single face algorithm (22). For this second part, each interior point is dependent upon the relative motion of one point on each of the six faces that is defined by the index markers of the point in question. The idea of WARP3D is to use an initial mesh with good quality attributes as a starting point, and then systematically perturb this mesh in a manner such that the original grid quality is maintained, without the need for expensive elliptic smoothing.

Since our current flow solver and design algorithm assume a point-to-point match between blocks, each block may be independently perturbed by WARP3D, provided that perturbed surfaces are treated continuously across block boundaries. The entire method of creating a new mesh is given by the following algorithm.

1. All faces of all blocks that are coincident with perturbed surfaces are explicitly perturbed.

2. The faces in all blocks that share an edge with an explicitly perturbed face are implicitly perturbed by a quasi-3D form of WARP3D.

3. WARP3D is used on each block that has one or more explicitly or implicitly perturbed faces to determine the adjusted interior points.

Note that much of the mesh, especially away from the surfaces, will not require mesh perturbations and thus may remain fixed through the entire design process. Since this mesh perturbation algorithm is analytic it is possible to work out the analytical variations in the metric terms required for equation (21). This approach was followed in reference [40]. However since the mesh perturbation algorithm that was used in the current paper was significantly more complex, and it was discovered that the computational cost of repeatedly using the block perturbation algorithm was minimal, finite differences were used to calculate δQ_{ij} instead of deriving the exact analytical relationships. Even in cases with hundreds of design variables, the computational cost of repeatedly re-evaluating δQ_{ij} for all necessary blocks is still insignificant compared with the cost of a single flow solution. The conclusion is that the analytical mesh perturbation algorithm, WARP3D, unlike an elliptical mesh generation method, is efficient to the extent that the cost of remeshing can be neglected.

It remains to choose a set of design variables which smoothly modifies the original shape, say b_i. The gradient can then be defined with respect to these design variables as

$$\mathcal{G}(b_i) = \frac{\delta I}{\delta b_i}, \tag{23}$$

where δI is calculated by (21) and each term b_i is independently perturbed by a finite step. Therefore, to construct $\mathcal{G}$, a basis space of independent perturbation functions b_i, $i = 1, 2, \ldots, n$ (n = number of design variables) must be chosen to allow for the needed freedom of the design space. In this work, design variables were chosen as a set of Hicks-Henne functions simply for their ease of implementaion and their proven reliability. To generalize the application of the Hicks-Henne functions, which have traditionally been used for the modification of airfoil sections where the x in equation (5) refers to the chordwise position, they are applied directly to the parametric $(\tilde{u}, \tilde{v})$ surfaces that represent the mesh faces. Thus they may be applied as functions in either the $\tilde{u}$, the $\tilde{v}$, or both directions. The design code is further structured so that these variables may be applied to any subset of the parametric surface. Alternatives are provided such that these variables may be linearly lofted in the second direction as opposed to say Hicks-Henne function in both directions. All of these options may be prescribed at the input level, leading to a highly versatile design code in which one or more faces in the multiblock domain may be perturbed by the design variables. To enforce geometric constraints, each design variable may be activated on more than one face. For example, if the thickness of a wing is to be preserved and the upper and lower halves of the wing are in separate blocks, then the design variables need to be applied at the proper locations with the proper weights and on the appropriate faces in both blocks such that thickness does not change while both surfaces are allowed to be modified.

IMPLEMENTATION OF THE DESIGN ALGORITHM

With all the necessary components defined for the multiblock adjoint based design, it is now possible to outline the complete procedure:

1. Solve the flow field governing equations (6-11).

2. Solve the adjoint equations (18) subject to the boundary condition (20).

3. For each of the n design variables repeat the following:

    - Perturb the design variable by a finite step according to (5, etc.).

- Explicitly perturb all faces affected by the design variable.
- Implicitly perturb all faces that share an edge with an explicitly perturbed design variable.
- Obtain the new internal mesh point locations via WARP3D for those blocks with perturbed faces.
- Calculate all the delta metric terms, $\delta Q_{i,j}$, within those blocks that were perturbed by finite diffferencing.
- Integrate equation (21) to obtain δI for those blocks that contain nonzero $\delta Q_{i,j}$.
- Determine the gradient component by equation (23).

4. Calculate the search direction and perform a line search.
5. Return to (1) if minimum has not been reached.

The basic method here builds on that used in reference [40] with the proper extensions to treat multiblock domains. In order to implement the method, equation (18) and boundary condition (20) must be discretized on the multiblock domain. In the current implementation, a cell centered, central difference stencil that mimics the flux balancing used for the flow solution is used. Since this choice of discretization differs from the one obtained if the discrete flow equation Jacobian matrix were actually transposed to form the adjoint system, the gradients obtained by the present method will not be exactly equal to the gradients calculated by finite differencing the discrete flow solutions. However, as the mesh is refined these differences should vanish. Continuing, the adjoint system so discretized is solved on the multiblock domain in an identical fashion to that used for the flow solution. Therefore, the adjoint solver, like the flow solver, uses an explicit multistage Runge-Kutta-like algorithm accelerated by residual smoothing and multigridding. Intra-block communication is again handled through a double halo which allows for the full transfer of information across boundaries except for the stencil of support for the implicit residual smoothing.

Step (3) in the above procedure is the portion of the method that is still treated by finite differences. Fortunately, all of these steps incur only a trivial computational cost compared with even a single flow analysis time step. It is therefore possible, without significant penalty, to leave this in finite difference form even for cases where many hundreds of design variables are used.

The present implementation uses the quasi-Newton algorithm, QNMDIF, developed by Gill, Murray and Pitfield [10] and enhanced by Kennelly [27], to calculate the search direction. It is an unconstrained optimization algorithm that uses Broyden-Fletcher-Goldfarb-Shanno (BFGS) updates to the Cholesky factored Hessian matrix. A complete treatment of the quasi-Newton and other optimization strategies is given by Gill, Murray and Wright [11].

NUMERICAL TESTS AND RESULTS

All of the design test cases to be presented in this paper use a business jet configuration that is the subject of another paper at this conference (see Reference [9]). The results presented in [9] were obtained through the use of the single block design method developed by our group and presented last year [40]. Therefore, this choice allowed us to validate the qualitative results of both the multiblock flow solver and the multiblock design method.

To demonstrate the utility of the new flow solver (FLO87-MB) an initial flow solution on a 72 block mesh with a total of 750 K cells is shown in Figure (2). The solution was carried out at a Mach number of 0.80 and a C_L of 0.3. It can be seen from the figure that this is a wing-body-nacelle geometry. The actual solution was carried out on the left half of the configuration with a symmetry plane boundary condition enforced to obtain a complete solution. The empennage and nacelle pylons were not modeled here simply to ensure that results could be obtained by the conference date. The surface of the geometry shown in Figure (2) is an isometric view colored by the local Mach number. The nacelle is modeled as flow through, with a single H-block traversing its entire interior. As can be seen in Figure (3), where a cut of the solution is taken through some of the blocks (dark lines indicate block boundaries) the general lay of the 72 block mesh is C-O. This was chosen for convenience since any topology is allowed within the multiblock framework. Figure (3) also shows that the contour lines, colored with constant Mach number, traverse the block boundaries without any evidence of solution mismatches. This validates the effectiveness of the basic multiblock flow solution strategy. Four multigrid levels were used for this solution on all blocks since the limiting (smallest) block contained only 8 cells in one coordinate direction. The solution presented here took 100 multigrid cycles to converge 4.5 orders in the RMS residual from the starting residual. This required about 18 minutes of CPU time on a single processor of a Cray C90.

The first step in establishing the validity of the adjoint based design method is to perform a check of the gradients it produces as compared with those obtained by finite differences. Figure (4) shows a comparison of adjoint gradients vs. finite difference gradients for test case 1 to be discussed in detail below. The finite difference gradients were calculated using forward differencing for only 24 of the 250 design variables used for test case 1 since these calculations were quite computationally expensive. Although the adjoint method calculated all the gradients for all 250 design variables, only those corresponding to those calculated for finite differences are shown in Figure (4). For the finite difference results the flow solution was always converged 4.5 orders from the initial starting residual. In order to reduce the computational cost of these evaluations the flow solutions for the gradient components were restarted from neighboring solutions. The adjoint gradients were obtained using 4.5 orders of convergence in the flow solution and 2.0 orders in the adjoint solution. The design variables consisted of Hicks-Henne functions in the chordwise direction and linear lofting in the spanwise direction. The results in Figure (4) show 24 design variables that span from the leading to the trailing edge along the upper surface at a constant span station of 0.44. The excellent agreement between the two curves shows that the continuous sensitivity approach, if properly applied, can give very accurate results. The remaining discrepancies between the methods could be produced by the limited convergence in either the flow or the adjoint solutions for either of the two methods, or simply from the discretization mismatch for the continuous adjoint approach. The key point in this comparison is that the calculation of the adjoint gradient for all 250 design variables took 37 minutes of Cray C90 single processor CPU time. The comparable finite difference calculation would have taken an estimated minimum of 1,000 minutes. Thus there is a factor of 27 between the computational costs of the two approaches.

The first test case used the mean square deviation from a target pressure as the objective function. Wing stations in the original geometry shown in Figures (2) and (3) were replaced by thickness-scaled NACA 0012 airfoils in all areas except for the tip and the root. The solution from Figures (2) and (3) was then used as a target pressure on the entire wing surface. The design therefore attempted to recover the original wing starting far from the solution. The design was run in constant α as opposed to constant C_L mode with a Mach number of 0.80. 250 design variables were distributed over the entire wing surface to allow for possible design changes. These variables assumed the form of Hicks-Henne functions in the chordwise direction and linear lofting in the spanwise direction. 50 variables spread over both the upper and lower surfaces were specified at each of 5 defining stations across the wing. The 72 block mesh has the property that three blocks abut to the wing upper surface and three blocks abut to the wing lower surface. Thus some of the design variables traversed more than one face in separate blocks. Figure (5) shows the pressure distributions and the targets at various stations along the wing for the starting point in the design. Figure (6) shows the solution and the target at the same stations after 12 design cycles. Note that the design almost fully recovered the original pressure distributions. This represents a drop in the calculated cost function from 6.02 to 0.10. The design algorithm was stopped after 12 cycles to conserve computer budget, though further reductions in the cost function were within reach. It must also be remembered that the Hicks-Henne functions do not admit a complete design space, so that even if convergence to a minimum is achieved, slight mismatches will remain.

The second test case attempts to improve the original configuration by reducing its drag at a higher Mach number than the original design Mach number. In this case the Mach number was set at 0.82 and C_L at 0.30. The design was run in fixed lift mode with C_D as the cost function to be minimized. Since the flow solver was inviscid, this drag consisted of form drag and induced drag. In this case 105 design variables of the Hicks-Henne form in the chordwise direction and linear lofting in the spanwise direction were used. To keep the wing from violating realistic constraints, the design variables were chosen so that the thickness distribution over the entire wing was preserved. This required some of the design variables to operate on up to four faces in different blocks simultaneously. Again the design variables were allowed to modify most of the wing except for near the tip and near the root. The initial and final designs after 6 cycles are shown in Figure (7). Note that the shock on the wing upper surface has been largely eliminated over the entire span. This is even achieved near the root where no geometric changes were allowed. To explain this phenomenon it must be remembered that the design was run in fixed lift mode. It turns out that the incidence of the entire geometry is reduced by the redesign, thus causing the upper surface not to have to work so hard. As a consequence it can be seen that the strength of the shock on the wing lower surface is slightly increased over much of the span. Apparently this trade-off produces a drag reduction. Most of the upper surface pressure distribution has been modified into near flat roof-top designs with weak to no shocks. The final configuration drag was reduced by 19%, most of which was due to the reduction of the drag on the wing.

CONCLUSIONS AND RECOMMENDATIONS

In the period since this approach to optimal shape design was first proposed by the second author [19], the method has been verified by numerical implementation for both potential flow and flows modeled by the Euler equations [20, 39, 25, 23]. It has been demonstrated that it can be successfully used with a finite volume formulation to perform calculations with arbitrary numerically generated grids [39, 25]. Further, results have been presented for three-dimensional calculations using both the analytic mapping and general finite volume implementations [40]. In the last year the technique has been adopted by some industry participants to perform the aerodynamic design of future configurations [9]. Now with the extension to multiblock meshes developed here, the technology has advanced to the degree that aerodynamic shape design of complete aircraft configurations is possible. To achieve this capability the flow solver, adjoint solver, and mesh perturbation algorithm were all extended to treat multiblock meshes. The use of the adjoint based design method reduced the computational time over that required for the finite difference based method by at least a factor of 27. An accompanying paper discusses the implementation of the method for parallel computer architectures [24]. In future efforts the techniques will be extended to address both unstructured meshes and flows governed by the Navier-Stokes equations.

ACKNOWLEDGMENTS

This research has benefited greatly from the generous support of the AFOSR under grant number AFOSR-91-0391, ARPA under grant number N00014-92-J-1976, USRA through RIACS, the High Speed Research branch of NASA Ames Research Center, and IBM. Considerable thanks also goes to Mark Rimlinger of Carnegie Mellon University for his assistance in the initial multiblock grid generation.

References

[1] O. Baysal and M. E. Eleshaky. Aerodynamic sensitivity analysis methods for the compressible Euler equations. *Journal of Fluids Engineering*, 113(4):681–688, 1991.

[2] O. Baysal and M. E. Eleshaky. Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. *AIAA Journal*, 30(3):718–725, 1992.

[3] O. Baysal and M. E. Eleshaky. Airfoil shape optimization using sensitivity analysis on viscous flow equations. *Journal of Fluids Engineering*, 115:75–84, 1993.

[4] G. W. Burgreen and O. Baysal. Aerodynamic shape optimization using preconditioned conjugate gradient methods. *AIAA paper 93-3322*, July 1993.

[5] G. W. Burgreen and O. Baysal. Three-dimensional aerodynamic shape optimization of wings using sensitivity analysis. *AIAA paper 94-0094*, 32nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1994.

[6] G. W. Burgreen, O. Baysal, and M. E. Eleshaky. Improving the efficiency of aerodynamic shape optimization procedures. *AIAA paper 92-4697*, September 1992.

[7] M. E. Eleshaky and O. Baysal. Preconditioned domain decomposition scheme for three-dimensional aerodynamic sensitivity analysis. In *Proceedings of of the 12th AIAA computational fluid dynamics conference*, pages 1055–1056, July 1993.

[8] M. E. Eleshaky and O. Baysal. Shape optimization of a 3-D nacelle near a flat plate wing using multiblock sensitivity analysis. *AIAA paper 94-0160*, 32nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1994.

[9] J. Gallman, J. Reuther, N. Pfeiffer, W. Forrest, and D. Bernstorf. Business jet wing design using aerodynamic shape optimization. *AIAA paper 96-0554*, 34th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1996.

[10] P. Gill, W. Murray, and R. Pitfield. The implementation of two revised quasi-Newton algorithms for unconstrained optimization. *NAC 11*, National Physical Laboratory, Division of Numerical Analysis and Computing, 1972.

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

[12] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[13] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats. An assessment of airfoil design by numerical optimization. *NASA TM X-3092*, Ames Research Center, Moffett Field, California, July 1974.

[14] J. Huan and V. Modi. Design of minimum drag bodies in incompressible laminar flow. Technical report, The Forum on CFD for Design and Optimization, (IMECE 95), San Francisco, California, November 1995.

[15] W.P. Huffman, R.G. Melvin, D.P. Young, F.T. Johnson, J.E. Bussoletti, M.B. Bieterman, and C.L. Hilmes. Practical design and optimization in computational fluid dynamics. *AIAA paper 93-3111*, AIAA 24th Fluid Dynamics Conference, Orlando, Florida, July 1993.

[16] A. C. Taylor III, G. W. Hou, and V. M. Korivi. Sensitivity analysis, approximate analysis, and design optimization for internal and external viscous flows. *AIAA paper 91-3083*, September 1991.

[17] A. Jameson. Solution of the Euler equations for two dimensional transonic flow by a multigrid method. *Applied Mathematics and Computations*, 13:327–356, 1983.

[18] A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[19] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[20] A. Jameson. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics, Tel Aviv*, pages 5–17, February 1990.

[21] A. Jameson. Aerodynamic design methods. In *International Workshop on Solution Techniques for Large-Scale CFD Problems*, Montreal, September 1994. CERCA.

[22] A. Jameson. Optimum aerodynamic design via boundary control. In *AGARD-VKI Lecture Series, Optimum Design Methods in Aerodynamics*. von Karman Institute for Fluid Dynamics, 1994.

[23] A. Jameson. *Optimum Aerodynamic Design Using Control Theory, Computational Fluid Dynamics Review 1995*. Wiley, 1995.

[24] A. Jameson and J.J. Alonso. Automatic aerodynamic optimization on distributed memory architectures. *AIAA paper 96-0409*, 34th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1996.

[25] A. Jameson and J. Reuther. Control theory based airfoil design using the Euler equations. *AIAA paper 94-4272*, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL, September 1994.

[26] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes. *AIAA paper 81-1259*, January 1981.

[27] R. Kennelly. Improved method for transonic airfoil design-by-optimization. *AIAA paper 83-1864*, AIAA Applied Aerodynamics Conference, Danvers, Massachusetts, July 1983.

[28] V. M. Korivi, A. C. Taylor III, G. W. Hou, P. A. Newman, and H. E. Jones. Sensitivity derivatives for three-dimensional supersonic Euler code using incremental iterative strategy. In *Proceedings of of the 12th AIAA computational fluid dynamics conference*, pages 1053–1054, July 1993.

[29] V. M. Korivi, A. C. Taylor III, and P. A. Newman. Aerodynamic optimization studies using a 3-D supersonic Euler code with efficient calculation of sensitivity derivatives. *AIAA paper 94-4270*, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, Florida, September 1994.

[30] V. M. Korivi, A. C. Taylor III, P. A. Newman, G. W. Hou, and H. E. Jones. An incremental strategy for calculating consistent discrete CFD sensitivity derivatives. *NASA TM 104207*, Langley Research Center, Hampton, VA, February 1992.

[31] G. Kuruvila, S. Ta'asan, and M. D. Salas. Airfoil optimization by the one-shot method. In *AGARD-VKI Lecture Series, Optimum Design Methods in Aerodynamics*. von Karman Institute for Fluid Dynamics, 1994.

[32] J.M. Lacasse and O. Baysal. Design optimization of single- and two-element airfoils on multiblock grids. *AIAA paper 94-4273*, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, Florida, September 1994.

[33] J. Lewis and R. Agarwal. Airfoil design via control theory using the full-potential and Euler equations. Technical report, The Forum on CFD for Design and Optimization, (IMECE 95), San Francisco, California, November 1995.

[34] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.

[35] P. A. Newman, G. W. Hou, H. E. Jones, A. C. Taylor III, and V. M. Korivi. Observations on computational methodologies for use in large-scale gradient-based, multidisciplinary design incorporating advanced CFD codes. *NASA TM 104206*, Langley Research Center, Hampton, VA, February 1992.

[36] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.

[37] O. Pironneau. Optimal shape design for aerodynamics. In *AGARD-VKI Lecture Series, Optimum Design Methods in Aerodynamics*. von Karman Institute for Fluid Dynamics, 1994.

[38] J. Reuther, S. Cliff, R. Hicks, and C.P. van Dam. Practical design optimization of wing/body configurations using the Euler equations. *AIAA paper 92-2633*, 1992.

[39] J. Reuther and A. Jameson. Control theory based airfoil design for potential flow and a finite volume discretization. *AIAA paper 94-0499*, 32nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1994.

[40] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA paper 95-0123*, 33rd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1995.

[41] J. Reuther and A. Jameson. A comparison of design variables for control theory based airfoil optimization. Technical report, 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, Nevada, September 1995.

[42] J. Reuther and A. Jameson. Supersonic wing and wing-body shape optimization using an adjoint formulation. Technical report, The Forum on CFD for Design and Optimization, (IMECE 95), San Francisco, California, November 1995.

[43] J. Reuther, C.P. van Dam, and R. Hicks. Subsonic and transonic low-Reynolds-number airfoils with reduced pitching moments. *Journal of Aircraft*, 29:297–298, 1992.

[44] J.P. Steinbrenner, J.R. Chawner, and C.L. Fouts. The GRIDGEN 3D multiple block grid generation system. Technical report, Flight Dynamics Laboratory, Wright Research and Development Center, Wright-Patterson Air Force Base, Ohio, July 1990.

[45] S. Ta'asan, G. Kuruvila, and M. D. Salas. Aerodynamic design and optimization in one shot. *AIAA paper 92-0025*, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1992.

[46] J.F. Thompson, Z.U.A Warsi, and C.W. Mastin. *Numerical Grid Generation, Foundations and Applications*. Elsevier Science Publishing Company, New York, NY, 1985.

[47] D.P. Young, W.P. Huffman, R.G. Melvin, M.B. Bieterman, C.L. Hilmes, and F.T. Johnson. Inexactness and global convergence in design optimization. *AIAA paper 94-4286*, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, Florida, September 1994.

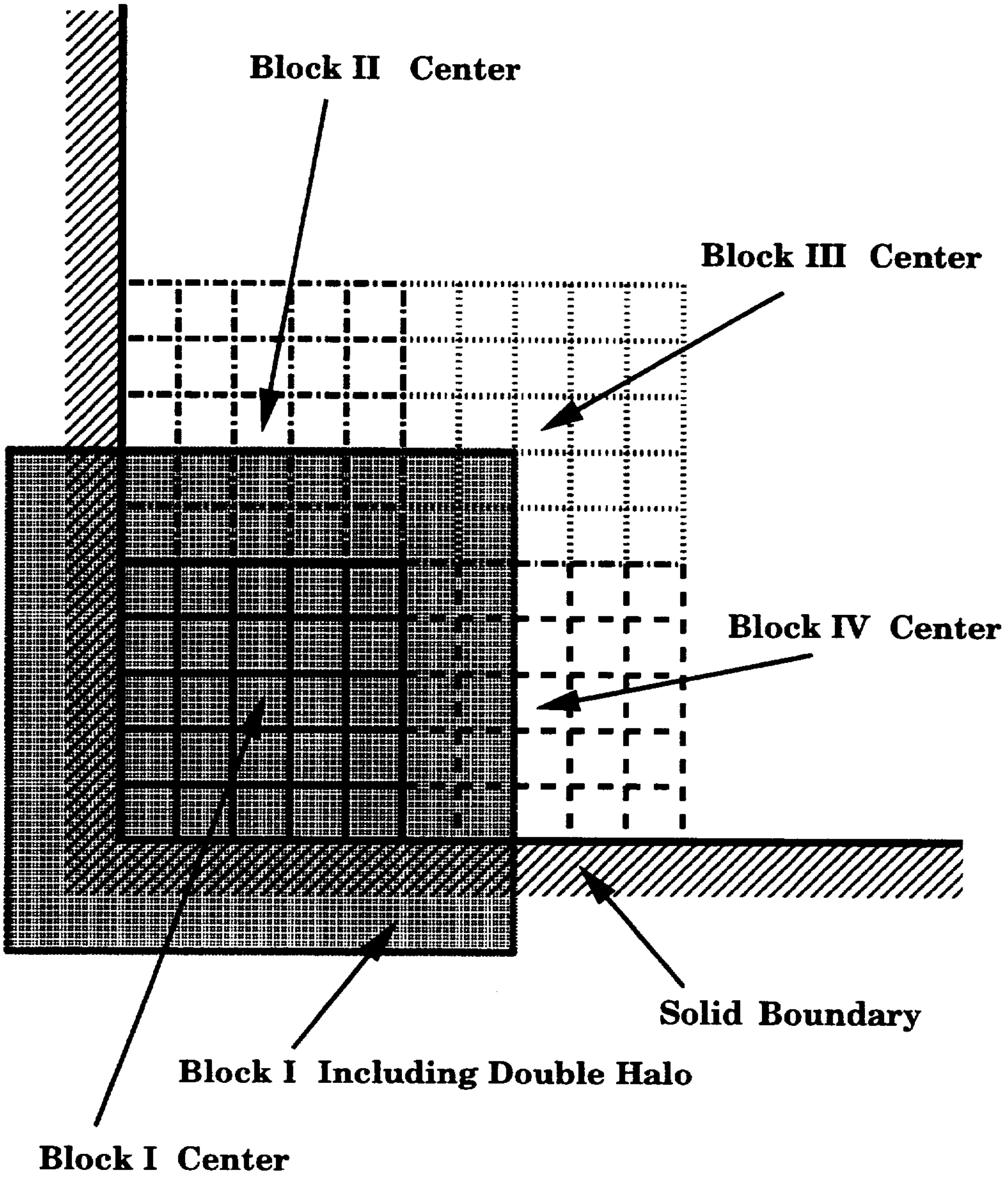


Figure 1: 4 Block interface using a double halo of cells around each block. Each block's double halo of cells contains information from internal cells in adjacent blocks.

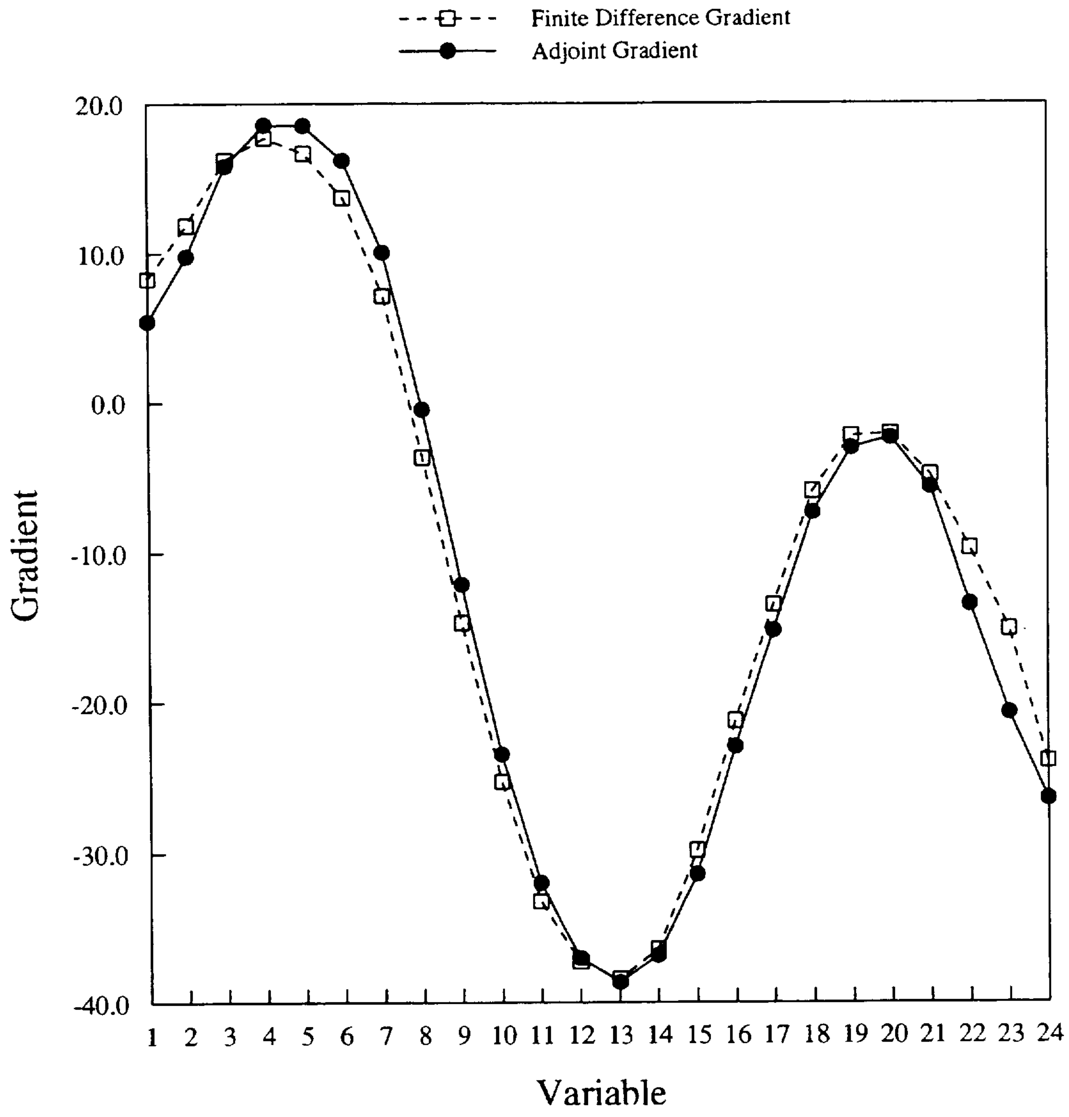


**Figure 4: Comparison of Gradients for 24 Discrete Design Variables
Adjoint vs. Finite Differences
Design Variables Span the Upper Surface at a Span Staion of 0.44
Beginning at the Leading Edge and Ending at The Trailing Edge**

SYN87-MB Solution on a typical business jet

72 Blocks - 750 k mesh points - Mach=0.80 - CL=0.30

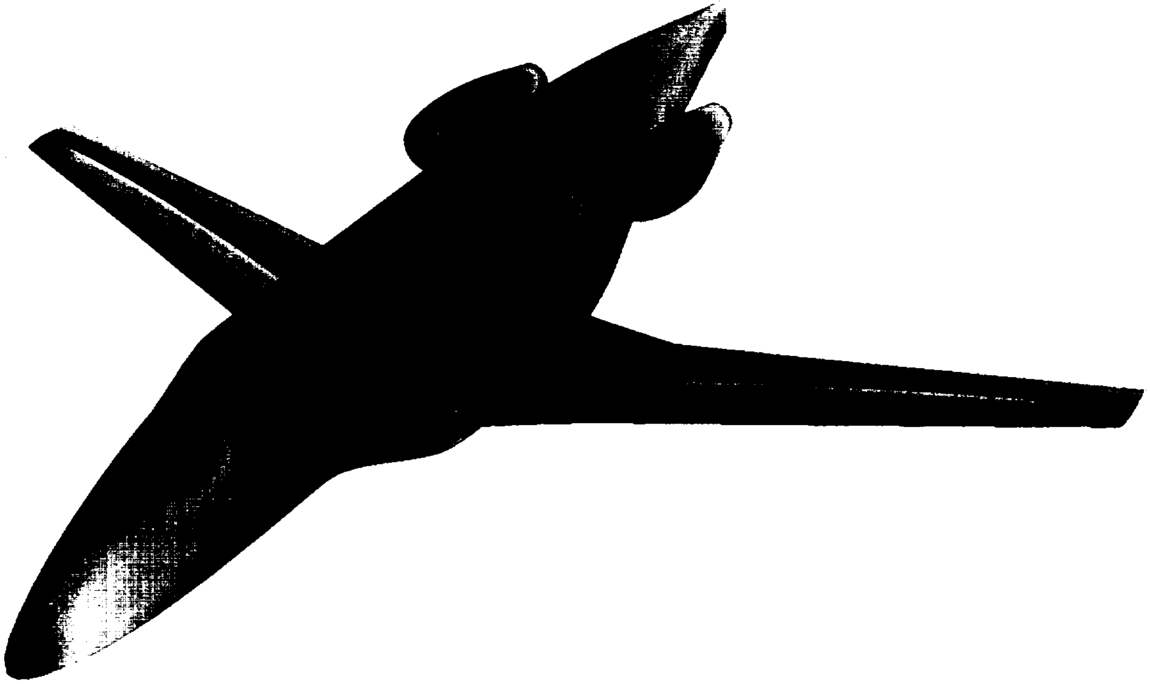


Figure 2: Initial flow solution for wing-body-nacelle geometry.
Surface colored by Mach number.

RIACS/Reuther

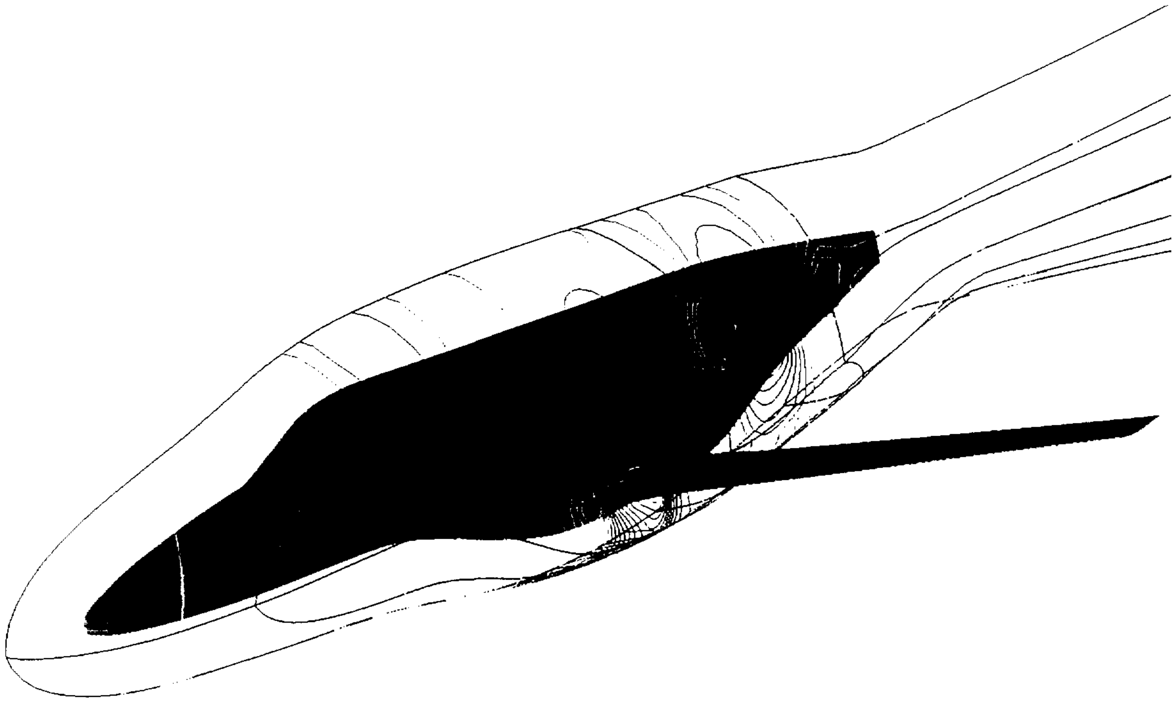
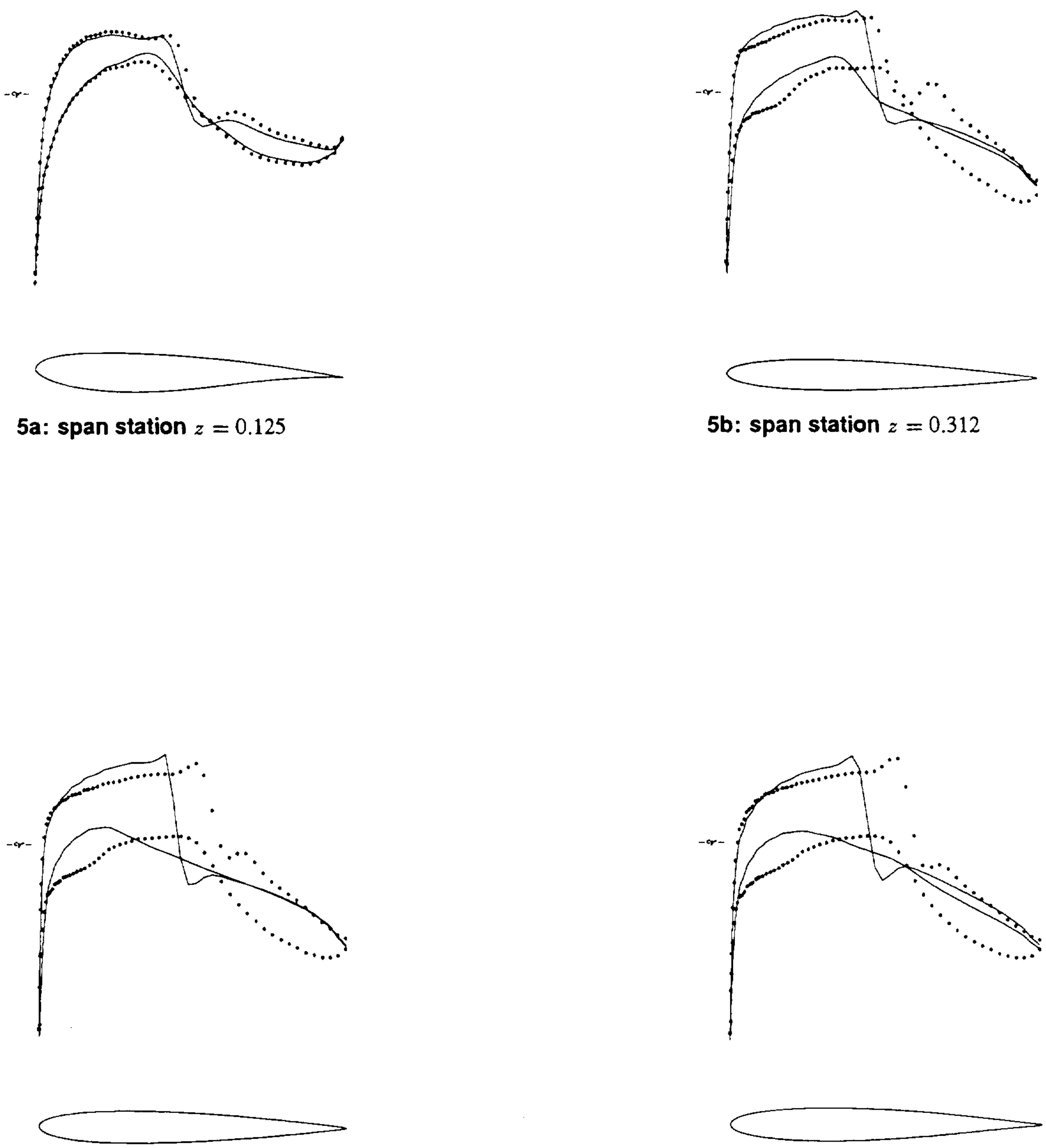


Figure 3: Initial flow solution showing block boundaries as dark lines
and contours colored by Mach number.



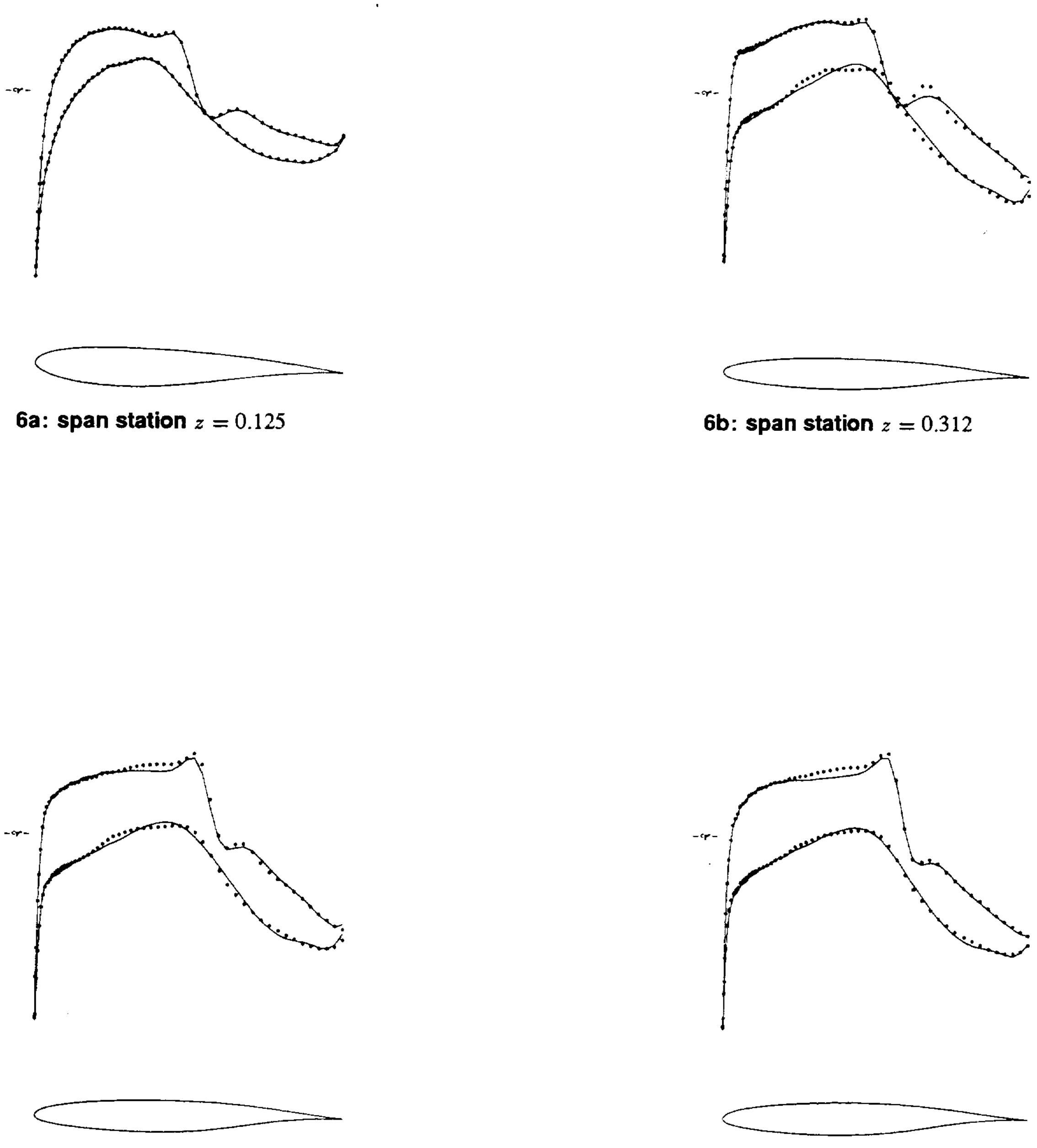
5a: span station $z = 0.125$

5b: span station $z = 0.312$

5c: span station $z = 0.559$

5d: span station $z = 0.764$

Figure 5: SYN87-MB Inverse Target Pressure Design.
72 Block Mesh, 750 K mesh cells, $M = 0.8$
250 Hicks-Henne variables.
$*$**, Target Pressures**
—, Initial Pressures.



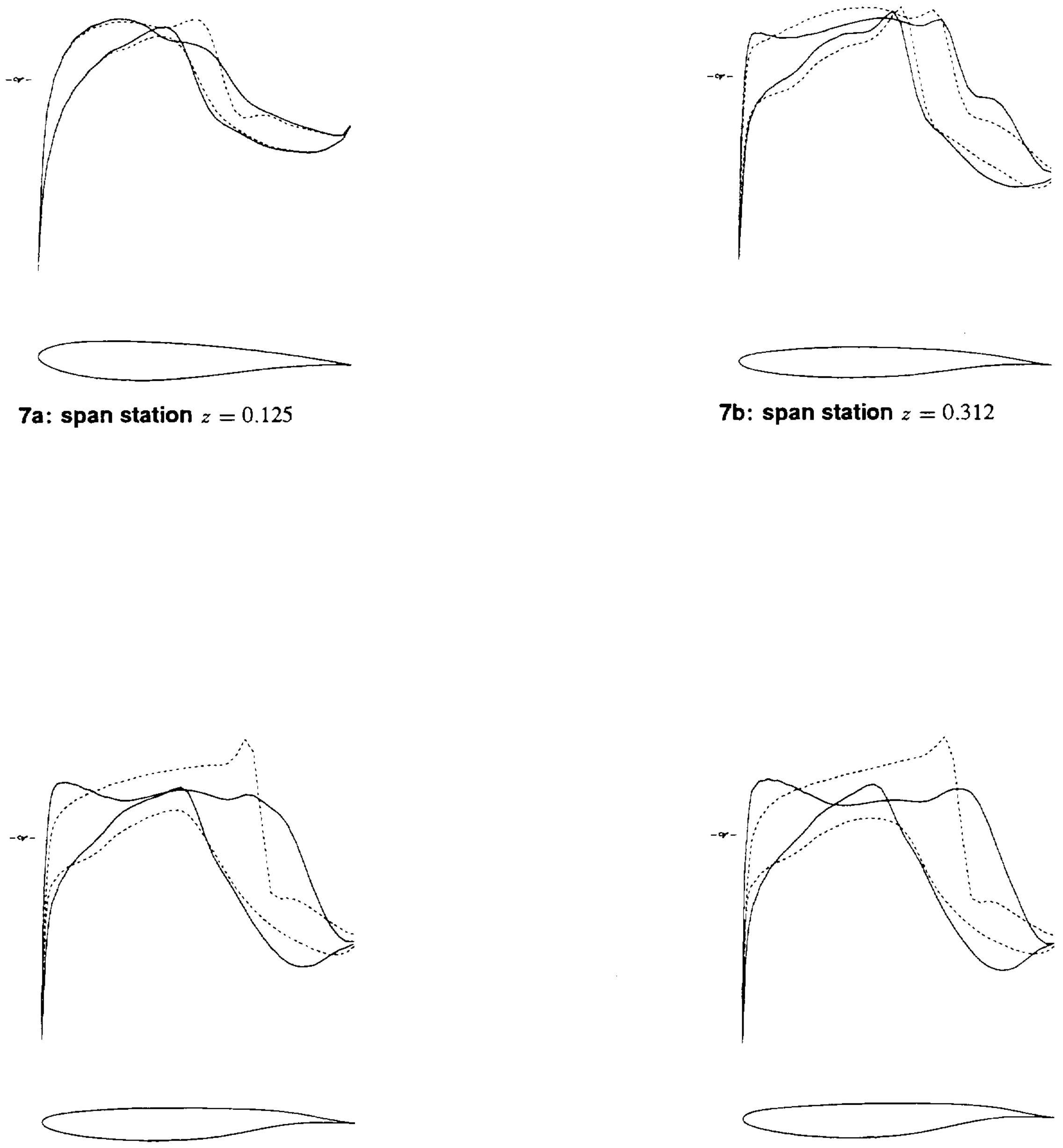
6a: span station $z = 0.125$

6b: span station $z = 0.312$

6c: span station $z = 0.559$

6d: span station $z = 0.764$

Figure 6: SYN87-MB Inverse Target Pressure Design.
72 Block Mesh, 750 K mesh cells, $M = 0.8$
250 Hicks-Henne variables.
$*$, Target Pressures
—, Design Pressures After 12 Design Cycles.



7a: span station $z = 0.125$

7b: span station $z = 0.312$

7c: span station $z = 0.559$

7d: span station $z = 0.764$

Figure 7: SYN87-MB, Fixed Lift Drag Minimization.
72 Block Mesh, 750 K mesh cells, $M = 0.82$, $C_L = 0.3$
105 Hicks-Henne variables.
- - -, Initial Pressures
—, Pressures After 6 Design Cycles.