RIACS

NASA-CR-204486

Research Institute for Advanced Computer Science
NASA Ames Research Center

CFD Analysis and Design Optimization Using Parallel Computers

Luigi Martinelli, Juan Jose Alonso, Antony Jameson, ans James Reuther

RIACS Technical Report 97.04 January 1997

CFD Analysis and Design Optimization Using Parallel Computers

Luigi Martinelli, Juan Jose Alonso, Antony Jameson, ans James Reuther

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 2-96027 between NASA and the Universities Space Research Association (USRA).

CFD ANALYSIS AND DESIGN OPTIMIZATION USING PARALLEL COMPUTERS

L. Martinelli, J.J. Alonso, A. Jameson
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544, U.S.A.

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

A versatile and efficient multi-block method is presented for the simulation of both steady and unsteady flow, as well as aerodynamic design optimization of complete aircraft configurations. The compressible Euler and Reynolds Averaged Navier-Stokes (RANS) equations are discretized using a high resolution scheme on body-fitted structured meshes. An efficient multigrid implicit scheme is implemented for time-accurate flow calculations. Optimum aerodynamic shape design is achieved at very low cost using an adjoint formulation. The method is implemented on parallel computing systems using the MPI message passing interface standard to ensure portability. The results demonstrate that, by combining highly efficient algorithms with parallel computing, it is possible to perform detailed steady and unsteady analysis as well as automatic design for complex configurations using the present generation of parallel computers.

1 INTRODUCTION

The essential requirements for the industrial use of Computational Fluid Dynamics (CFD) are: (1) reliable solution accuracy, (2) acceptable computational cost, (3) complex geometry treatment and (4) rapid solution/design turnaround.

Advances in both algorithms and computing hardware have been and will be necessary to fulfill these requirements. During the eighties the development of vector processors allowed the aeronautical engineer to analyze steady-state flow problems. In the last three years parallel computing has begun the transition from research laboratories to industrial environments. Today, parallel computing promises to enhance our flow prediction capability by allowing the analysis of time-dependent flow and by enabling the the automatic design optimization of complete aircraft. There remains the challenge of developing application software which can take full advantage of the computing power of these new architectures.

With these goals in mind, much effort by our research group has been placed on the development of accurate and efficient methods for the calculation of steady and time dependent three-dimensional inviscid and viscous flows. Not only are these analysis methods necessary purely for evaluating candidate designs, but they also form the core of any CFD based design approach. The pursuit of high accuracy has focused on the implementation of refined artificial dissipation algorithms which provide the necessary upwind bias without corrupting the physical phenomena at hand. Efficiency has been achieved through the application of multigrid algorithms and the utilization of high performance scalable parallel computing platforms.

For time-resolved flow calculations we have continued the development of a very efficient multigrid implicit scheme originally presented by Jameson for the compressible Euler equations [1]. This method has proved effective for the calculation of unsteady flows that might be associated with wing flutter [2,3], and for the calculation of unsteady incompressible flows [4]. It has also been applied recently to simulate helicopter rotor flows [5]. The method has the advantage that it can be added as an option to a computer program which uses an explicit multigrid scheme, allowing it to perform efficient calculations of both steady and unsteady flows.

Three alternative approaches are available for the discretization of complex configurations: (1) Cartesian meshes, (2) unstructured tetrahedral meshes, and (3) body-fitted meshes. These basic techniques can be also combined into a variety of hybrid mesh strategies. Each of these approaches has advantages and disadvantages. In this work we use body-fitted hexahedral meshes. These are particularly well suited for the treatment of viscous flow because they readily allow the mesh to be refined in the region near the body and in the direction normal to the surface. However, in order to use body-fitted meshes for very complex configurations it generally proves necessary to use a multi-block procedure [6,7], whereby multiple structured meshes are pieced together to form the entire flow-field domain. From our perspective, the major advantage of a multi-block approach is that it allows a straightforward extension to complex geometries of a family of well validated computer codes originally written for single-block meshes.

With currently available computers the turnaround for numerical simulations is becoming so rapid that it is feasible to examine an extremely large number of variations. However, it is not at all likely that interactive analysis and a design approach involving significant user intervention will lead to truly optimum designs. To examine a larger design space and realize substantial improvements in aerodynamic efficiency of new designs, CFD simulations need to be combined with automatic search and optimization procedures. Thus, concurrently with the development of improved analysis methods, we have made considerable efforts toward the development of automatic design methodologies. The problems of drag minimization and inverse design can both be systematically treated within the mathematical theory for the control of systems governed by partial differential equations [8]. The control theory approach to optimal aerodynamic design, whereby the boundary shape becomes the control, and the gradient of the cost function with respect to shape changes is obtained by solving the adjoint problem for the given set of governing equations, was first applied to transonic flow by Jameson[9,10]. The adjoint approach has been recently implemented in the inviscid version of the multi-block flow solver [11,12,13], thus allowing for the optimization of complete configurations.

The mathematical models governing compressible flow are discussed in the next section. Section 3 presents the numerical algorithms for flow simulation. Section 5 presents the results of some numerical calculations for steady and time resolved flows on complex configurations. Section 5.3 discusses automatic design procedures which can be used to produce optimum aerodynamic designs, and presents the results for the aerodynamic optimization of a typical transonic business jet configuration and a supersonic transport configuration.

2 MATHEMATICAL MODEL

The dynamics of compressible fluid flows are governed by the Navier-Stokes equations. Let (x_1,x_2,x_3) be a Cartesian coordinate system. By adopting the convention of indicial notation where a repeated index "i" implies summation over $i = 1$ to 3, the three-dimensional Navier-Stokes equations take the form

$$\frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial x_i} = \frac{\partial f_{vi}}{\partial x_i} \quad \text{in } \mathcal{D}, \tag{1}$$

where the state vector w, inviscid flux vector f, and viscous flux vector f_v are described by

$$w = \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{array} \right\}, \quad f_i = \left\{ \begin{array}{c} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ \rho u_i H \end{array} \right\}, \quad f_{vi} = \left\{ \begin{array}{c} 0 \\ \sigma_{ij}\delta_{j1} \\ \sigma_{ij}\delta_{j2} \\ \sigma_{ij}\delta_{j3} \\ u_j\sigma_{ij} + k\frac{\partial T}{\partial x_i} \end{array} \right\}. \tag{2}$$

In these definitions, ρ is the density, u_1, u_2, u_3 are the Cartesian velocity components, E is the total energy and δ_{ij} is the Kronecker delta function. The pressure is determined by the equation of state

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

and the stagnation enthalpy is given by

$$H = E + \frac{p}{\rho},$$

where γ is the ratio of the specific heats. The viscous stresses may be written as

$$\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\delta_{ij}\frac{\partial u_k}{\partial x_k}, \tag{3}$$

where μ and λ are the first and second coefficients of viscosity. The coefficient of thermal conductivity and the temperature are defined by

$$k = \frac{\gamma\mu}{Pr}, \qquad T = \frac{p}{(\gamma - 1)\rho}. \tag{4}$$

When using a discretization on a body-conforming structured mesh, it is useful to consider a transformation to computational coordinates (ξ_1,ξ_2,ξ_3) defined by the metrics

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

The Navier-Stokes equations can then be rewritten in computational space as

$$\frac{\partial (Jw)}{\partial t} + \frac{\partial (F_i - F_{vi})}{\partial \xi_i} = 0 \text{ in } \mathcal{D}, \tag{5}$$

where the inviscid and viscous flux contributions are now defined with respect to the computational cell faces by $F_i = S_{ij} f_j$ and $F_{vi} = S_{ij} f_{vj}$, and the quantity $S_{ij} = J K_{ij}^{-1}$ is used to represent the projection of the ξ_i cell face along the x_j axis. In obtaining equation (5) we have made use of the property that

$$\frac{\partial S_{ij}}{\partial \xi_i} = 0, \tag{6}$$

which represents the fact that the sum of the face areas over a closed volume is zero, as can be readily verified by a direct examination of the metric terms.

When the mesh is non stationary, the calculation of the flux must take into account the motion of the mesh. For a moving mesh, the conservation equations are obtained by computing the convective flux based on the fluid velocity relative to the moving mesh. If the mesh deforms, the time variation of the control volumes must also be accounted for.

Many critical phenomena of fluid flow, such as shock waves and turbulence, are highly nonlinear and exhibit extreme disparities of scales. While the actual thickness of a shock wave is of the order of the mean free path of the gas particles, on a macroscopic scale its thickness is virtually zero. In turbulent flows, energy is transferred from large scale motions to progressively smaller eddies until the scale becomes so small that the motion is dissipated by viscosity. The ratio of the length scale of the global flow to that of the smallest persisting eddies is of order $\mathrm{Re}^{\frac{3}{4}}$, where Re is the Reynolds number (typically in the range of 30 million for a transport aircraft). In order to resolve such scales in all three spatial directions, a computational grid with order $\mathrm{Re}^{\frac{9}{4}}$ cells would be required. This is beyond the range of any current or foreseeable computer.

Accurate modeling of multi-scale phenomena has presented the CFD research community with a challenge that has yet to be fully resolved. With regards to shock waves, the development of new high resolution schemes has resulted in very accurate models for shock capturing. To simulate turbulent flows, simplified models must be constructed. In the limit of infinite Reynolds number, the contributions due to viscosity and heat conduction vanish. Thus equation (5) may be reduced under such assumptions to the Euler equations. This inviscid model may be suitable for describing the flow on most aircraft at cruise conditions. However, viscous effects must be ultimately be taken into account since shock waves and boundary layers often interact with a dramatic effect on the flow field.

When viscous effects and turbulence play a salient role, a common approach is to time average the Navier-Stokes equations. This produces the Reynolds Averaged Navier-Stokes (RANS) system which governs the dynamics of the mean flow. The use of the RANS equations brings viscous flow calculations within the threshold of feasibility on modern computers. Unfortunately, the averaging process results in additional terms and unknowns which require a turbulence model for closure of the system of equations. In this work a very simple algebraic closure model, originally developed by Baldwin and Lomax [14], is used. This model has proved satisfactory for the calculation of attached and slightly separated wing flows [15], and with appropriate modifications has been successfully applied to vortical flows [16,17]. Closure models based on the solution of transport equations for the turbulent kinetic energy k and the dissipation rate ϵ, or for a pair of equivalent quantities [18,19,20,21,22,23], will be implemented in our multi-block solver in the near future.

3 NUMERICAL METHOD

3.1 SPATIAL DISCRETIZATION

The discretization of the spatial operators is accomplished by using a cell-center finite volume method. The flow domain is divided into a large number of small subdomains, and the integral form of the conservation laws

$$\frac{\partial}{\partial t}\int_{\mathcal{D}} \mathbf{w} dV + \int_{\mathcal{B}} \mathbf{F}\cdot d\mathbf{S} = 0$$

is applied to each subdomain. Here $\mathbf{F}$ is the flux appearing in equation (5) and $d\mathbf{S}$ is the directed surface element of the boundary $\mathcal{B}$ of the domain $\mathcal{D}$. The use of the integral form has the advantage that no assumption of the differentiability of the solution is implied, with the result that it remains a valid statement for a subdomain containing a shock wave. In general the subdomains could be arbitrary, but in this work we use the hexahedral cells of a body-conforming curvilinear mesh. Discretizations of this type reduce to central differences on a regular Cartesian grid, and in order to eliminate possible odd-even decoupling modes allowed by the discretization some form of artificial dissipation must be added. Moreover, when shock waves are present, it is necessary to upwind the discretization to provide a non-oscillatory capture of discontinuities. In the present work this goal is achieved by using a Convective Upstream Split Pressure (CUSP) approach, coupled with an Essential Local Extremum Diminishing (ELED) formulation. Details on this technique and an extensive validation of the scheme for both inviscid and viscous flow, can be found in [24,25,26].

To include the viscous terms of the Navier-Stokes equations into the spatial discretization scheme it is necessary to approximate the velocity derivatives $\frac{\partial u_i}{\partial x_j}$ which constitute the stress tensor σ_{ij}. These derivatives may be evaluated by applying Gauss' formula to a control volume V with the boundary S:

$$\int_V \frac{\partial u_i}{\partial x_j} dV = \int_S u_i n_j dS \ ,$$

where n_j is the outward normal. For a hexahedral cell this gives

$$\overline{\frac{\partial u_i}{\partial x_j}} = \frac{1}{\mathrm{V}} \sum_{\text{faces}} \ \overline{u}_i \ n_j \ S \tag{7}$$

where $\overline{u}_i$ is an estimate of the average of u_i over the face, n_j is the $j-th$ component of the normal, and S is the face area.

3.2 TIME STEPPING SCHEME FOR STEADY-STATE SOLUTIONS

If the space discretization procedure is implemented separately from the discretization in time, it leads to a set of coupled ordinary differential equations which can be written in the form

$$\frac{d\mathbf{w}}{dt} + \mathbf{R}(\mathbf{w}) = \mathbf{0}, \tag{8}$$

where $\mathbf{w}$ is the vector of the flow variables at the mesh locations, and $\mathbf{R}(\mathbf{w})$ is the vector of the residuals, consisting of the flux balances defined by the spatial discretization together with the added dissipative terms. If the objective is simply to reach the steady state and details of the transient solution are immaterial, the time-stepping scheme may be designed solely to maximize the rate of convergence.

Throughout this work we use a multistage explicit scheme, belonging to the general class of Runge-Kutta schemes[27]. Schemes of this type have proved very effective for a wide variety of problems, and they have the advantage that they can be applied equally easily on both structured and unstructured meshes [28,29,30,31,32].

If one reduces the linear scalar model problem corresponding to (8) to an ordinary differential equation by substituting a Fourier mode $\hat{w} = e^{ipx_j}$, the resulting Fourier symbol has an imaginary part proportional to the wave speed, and a negative real part proportional to the diffusion. Thus, the time stepping scheme should have a stability region which contains a substantial interval of the negative real axis, as well as an interval along the imaginary axis. To achieve this we treat the convective and dissipative terms in a distinct fashion. Thus the residual is split as

$$R(w) = Q(w) + D(w),$$

where $Q(w)$ is the convective part and $D(w)$ the dissipative part. Denote the time level $n\Delta t$ by a superscript n. Then the multistage time stepping scheme is formulated as

$$\begin{aligned} w^{(n+1,0)} &= w^n \\ &\dots \\ w^{(n+1,k)} &= w^n - \alpha_k \Delta t \left(Q^{(k-1)} + D^{(k-1)} \right) \\ &\dots \\ w^{n+1} &= w^{(n+1,m)}, \end{aligned}$$

where the superscript k denotes the k-th stage, $\alpha_m = 1$, and

$$\begin{aligned} Q^{(0)} &= Q\left(w^n\right), \quad D^{(0)} = D\left(w^n\right) \\ &\dots \\ Q^{(k)} &= Q\left(w^{(n+1,k)}\right) \\ D^{(k)} &= \beta_k D\left(w^{(n+1,k)}\right) + (1-\beta_k) D^{(k-1)}. \end{aligned}$$

The coefficients α_k are chosen to maximize the stability interval along the imaginary axis, and the coefficients β_k are chosen to increase the stability interval along the negative real axis.

The coefficients of a five-stage scheme [33] which has been found to be particularly effective are tabulated below.

$$\begin{array}{ll} \alpha_1 = \frac{1}{4} & \beta_1 = 1 \\ \alpha_2 = \frac{1}{6} & \beta_2 = 0 \\ \alpha_3 = \frac{3}{8} & \beta_3 = 0.56 \\ \alpha_4 = \frac{1}{2} & \beta_4 = 0 \\ \alpha_5 = 1 & \beta_5 = 0.44 \end{array} . \tag{9}$$

3.3 ACCELERATION OF STEADY FLOW CALCULATIONS

Radical improvements in the rate of convergence to a steady-state solution can be realized by the multigrid time-stepping technique. The concept of acceleration by the introduction of multiple grids was first proposed by Fedorenko [34]. There is by now a fairly well-developed theory of multigrid methods for elliptic equations based on treating the updating scheme as a smoothing operator on each grid [35,36]. This theory does not hold for hyperbolic systems. Nevertheless, it seems that it ought to be possible to accelerate the evolution of a hyperbolic system to a steady state by using large time steps on coarse grids so that disturbances can be more rapidly expelled through the outer boundary. Various multigrid time-stepping schemes designed to take advantage of this effect have been proposed [37,38,39,40,41,42,43,44,45].

In our work we implement a multigrid scheme, originally developed by Jameson [38], which uses a sequence of coarser meshes generated by eliminating alternate points in each coordinate direction. In order to give a precise description of the multigrid scheme, subscripts may be used to indicate the grid level. Several transfer operations need to be defined. First the solution vector on grid k must be initialized as

$$w_k^{(0)} = T_{k,k-1} w_{k-1},$$

where w_{k-1} is the current value on grid $k-1$, and $T_{k,k-1}$ is a transfer operator. Next it is necessary to transfer a residual forcing function such that the solution on grid k is driven by the residuals calculated on grid $k-1$. This can be accomplished by setting

$$P_k = Q_{k,k-1} R_{k-1} \left(w_{k-1} \right) - R_k \left[w_k^{(0)} \right],$$

where $Q_{k,k-1}$ is another transfer operator. Then $R_k(w_k)$ is replaced by $R_k(w_k) + P_k$ in the time- stepping scheme. Thus, the multistage scheme is reformulated as

$$\begin{aligned} w_k^{(1)} &= w_k^{(0)} - \alpha_1 \Delta t_k \left[R_k^{(0)} + P_k \right] \\ \dots & \quad \dots \\ w_k^{(q+1)} &= w_k^{(0)} - \alpha_{q+1} \Delta t_k \left[R_k^{(q)} + P_k \right]. \end{aligned}$$

The result $w_k^{(m)}$ then provides the initial data for grid $k+1$. Finally, the accumulated correction on grid k has to be transferred back to grid $k-1$ with the aid of an interpolation operator $I_{k-1,k}$. With properly optimized coefficients, multistage time-stepping schemes can be very efficient drivers of the multigrid process. A W-cycle of the type illustrated in Figure 1 proves to be a particularly effective strategy for managing the work split between the meshes. In a three-dimensional case the number of cells is reduced by a factor of eight on each coarser grid. On examination of the figure, it can therefore be seen that the work measured in units corresponding to a step on the fine grid is of the order of

$$1 + 2/8 + 4/64 + \ldots < 4/3,$$

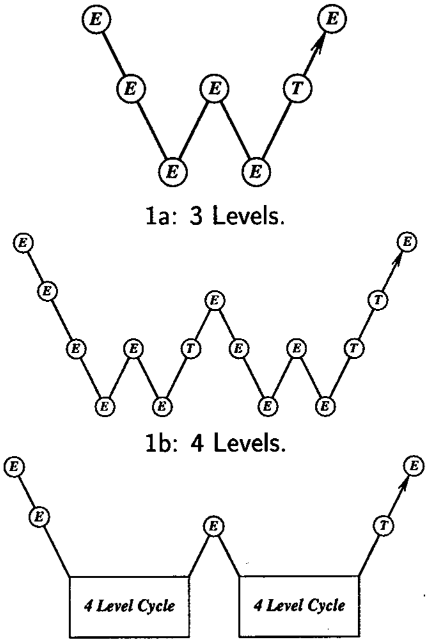
and consequently the very large effective time step of the complete cycle costs only slightly more than a single time step in the fine grid.

3.4 A MULTIGRID IMPLICIT SCHEME FOR UNSTEADY FLOW

Time dependent calculations are needed for a number of important applications, such as flutter analysis or the simulation of the flow past a helicopter rotor, in which the stability limit of an explicit scheme forces the use of much smaller time steps than would be needed for an accurate simulation. In this situation a multigrid explicit scheme can be used in an inner iteration to drive the solution of a fully implicit time discretization [1].

Suppose that (8) is approximated as

$$D_t w^{n+1} + R(w^{n+1}) = 0.$$



1a: 3 Levels.

1b: 4 Levels.

1c: 5 Levels.

Figure 1: Multigrid W-cycle for managing the grid calculation. E, evaluate the change in the flow for one step; T, transfer the data without updating the solution.

Here D_t is a k^{th} order accurate backward difference operator of the form

$$D_t = \frac{1}{\Delta t} \sum_{q=1}^{k} \frac{1}{q} (\Delta^-)^q,$$

where

$$\Delta^- w^{n+1} = w^{n+1} - w^n.$$

Applied to the linear differential equation

$$\frac{dw}{dt} = \alpha w,$$

the schemes with $k = 1, 2$ are stable for all $\alpha \Delta t$ in the left half plane (A-stable). Dahlquist has shown that A-stable linear multi-step schemes are at best second order accurate[46]. Gear however, has shown that the schemes with

$k \leq 6$ are stiffly stable[47], and one of the higher order schemes may offer a better compromise between accuracy and stability, depending on the application.

Equation (8) is now treated as a modified steady state problem to be solved by a multigrid scheme using variable local time steps in a fictitious time t^*. For example, in the case $k = 2$ one solves

$$\frac{\partial w}{\partial t^*} = R^*(w),$$

where

$$R^*(w) = \frac{3}{2\Delta t} w + R(w) + \frac{2}{\Delta t} w^n - \frac{1}{2\Delta t} w^{n-1},$$

and the last two terms are treated as fixed source terms. The first term shifts the Fourier symbol of the equivalent model problem to the left in the complex plane. While this promotes stability, it may also require a limit to be imposed on the magnitude of the local time step Δt^* relative to that of the implicit time step Δt. This limitation may be relieved by a point-implicit modification of the multi-stage scheme[48]. In the case of problems with moving boundaries the equations must be modified to allow for movement and deformation of the mesh.

3.5 DYNAMIC REMESHING AND MESH MOVEMENT

In an Eulerian reference frame both the aerodynamic shape design problem and the unsteady aeroelastic problem require a method by which the computational meshes may be efficiently and robustly regenerated. Either problem may demand many hundreds of independent meshes on which the solution is to be calculated.

Traditional structured mesh generation approaches, such as those that solve elliptic or hyperbolic sets of partial differential equations, would be impractical in this setting. These iterative approaches are computationally expensive, and their repeated use for dynamic remeshing would be prohibitively expensive. A second problem presented by the requirement of treating complex geometries is that truly automated methods of generating arbitrary multi-block meshes do not presently exist. In this paper we pursue the commonly used alternative of analytic mesh perturbations. In this approach, a high quality surface and volume mesh is first generated about the initial geometry by any available procedure prior to the start of the time dependent analysis or the optimal design. This initial mesh becomes the basis for all subsequent meshes which are developed by analytic perturbations. In the case where only one surface, such as the wing, is perturbed during the computation, the method reduces to 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

$$x_i^{new} = x_i^{old} + \mathcal{S}^{old}\left(x_{s_i}^{new} - x_{s_i}^{old}\right) \ , \tag{10}$$

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 and normalized so that $\mathcal{S} = 1$ at the inner surface and 0 at the far field.

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 (10) had to be modified in a way that resembles transfinite interpolation (TFI) [49]. Unlike TFI, where there is no prior knowledge of the interior mesh, the perturbation algorithm developed here makes use of the relative interior point distributions in the initial mesh. In our general implementation of the perturbation method, a three-stage procedure is used. For each block in the multi-block mesh the first stage shifts the internal mesh points to produce an interim block that is determined entirely by the new locations of the 8 corner points defining the block. The second stage corrects the perturbations resulting from the first stage by determining the distance that each of the 12 edges resulting from the first stage needs to be moved to attain the desired edge locations. Finally, with both corner and edge point motion accounted for, the third stage corrects the internal points for the relative motion of the six faces.

Since our current flow solver and design algorithm assume a point-to-point match between blocks, each block may be independently perturbed by the algorithm, 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 that are directly affected by the moving surfaces (active faces) are explicitly perturbed.

2. All edges that touch an active face, either in the same block or in an adjacent block, are implicitly perturbed by (10).

3. All inactive faces that either include an implicitly perturbed edge or abut to an active face are implicitly perturbed by a quasi-3D form of the general algorithm.

4. The three-stage scheme outlined above is then 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 throughout the entire unsteady analysis or design process. Close to the surfaces, many blocks will either contain an active face or touch a block which contains an active face, either by an edge or by a corner. As the design variations affect the active faces, the above scheme ensures that the entire mesh will remain attached along block boundaries. Added complexity is needed to accomplish step (2) since the connectivity of the various edges and corners must be specified somehow. Currently, pointers to and from a set of master edges and master corners are determined as a preprocessing step. During the calculation, the motion of any edges and corners are transferred to these master edges and corners from which all connected edges and corners can be updated.

4 DOMAIN DECOMPOSITION AND PARALLEL IMPLEMENTATION

The multi-block method is parallelized using a domain decomposition model, a SPMD (Single Program Multiple Data) strategy, and the MPI (Message Passing Interface) Library for message passing [50]. The choice of MPI was determined by the requirement that the resulting code be portable to different parallel computing platforms as well as to homogeneous and heterogeneous networks of workstations.

Communication between subdomains is performed through halo cells surrounding each subdomain boundary. Since both the convective and the viscous fluxes are calculated at the cell faces (boundaries of the control volumes), all six neighboring cells are necessary, thus requiring the existence of a single level halo for each processor in the parallel calculation. The calculation of the dissipative fluxes requires values from the twelve neighboring cells (two adjacent to each face). For each cell within a processor, Figure 2 shows which neighboring cells are required for the calculation of convective, viscous, and dissipative fluxes. For each processor, some of these cells will lie directly next to an interprocessor boundary, in which case the values of the flow variables residing in a different processor will be necessary to calculate the convective and dissipative fluxes.

The actual communication routines used are all of the asynchronous (or non-blocking) type. In the current implementation of the program, each processor must send and receive messages to and from at most 6 neighboring

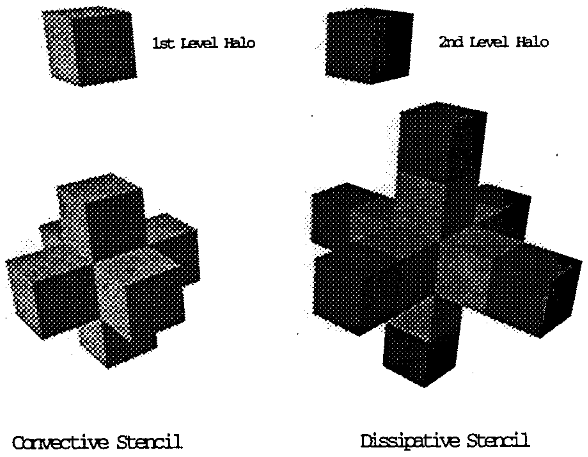


Figure 2: Convective and Dissipative Discretization Stencils.

processors (left and right neighbors in each of the three coordinate directions). The communication is scheduled such that at every instant in time, pairs of processors are sending/receiving to/from one another in order to minimize contention in the communication schedule.

The partitioning of the mesh is performed by allocating complete blocks to each processor. The underlying assumption is the fact that there will always be more blocks than processors available. This approach has the advantage that the number of multigrid levels that can be used in the parallel implementation of the code is always the same as in the serial version. Moreover, the number of processors in the calculation can now be any integer number, since no restrictions are imposed by the partitioning in any of the coordinate directions within each block.

The only drawback of this approach is the loss of the exact load balancing that can be achieved by partitioning single-block meshes along the three coordinate directions. The various blocks in the calculation can have different sizes, and consequently, it is very likely that each processor will be assigned a different total number of cells in the calculation. This, in turn, will imply that some of the processors will be waiting until the processor with the largest number of cells has completed its work and parallel performance will suffer. The approach that we have followed to solve the load balancing problem is to assign to each processor, in a pre-processing step, a certain number of blocks such that the total number of cells is as close as possible to the exact share for per-

Processor Number	Percentage of Load
1	12.50000
2	12.50000
3	12.98701
4	12.01298
5	12.98701
6	12.01298
7	12.50000
8	12.50000

Table 1: Calculated Load Distribution on an 8 Processor Calculation

fect load balancing. For example, one of the meshes for the wing/body/nacelle configuration of a small business jet was made up of 72 structured blocks of different sizes. When 8 processors are used, the load balance obtained can be seen in Table 1 to be quite close to exact. One should note that load balancing based on the total number of cells in each processor is only an approximation to the optimal solution of the problem. Other variables such as the number of blocks, the size of each block, and the size of the buffers to be communicated play an important role in proper load balancing.

Parallel Efficiency

For problems with a low task granularity (ratio of the number of bytes received by a processor to the number of floating point operations it performs), large parallel efficiencies can be obtained. Unfortunately, convergence acceleration techniques developed in the 1980s base their success on global communication in the computational domain. Thus, current multigrid and implicit residual smoothing techniques are bound to hinder parallel performance for problems with smaller mesh sizes. For larger meshes used in viscous turbulent flow calculations on complete configurations, the granularity becomes low enough, and the parallel performance is quite high.

Several techniques can be applied to reduce the communication cost of the multigrid technique. Among these, one can consider completely eliminating communication at the coarser levels of the multigrid cycle (thus allowing each processor to operate independently with the multigrid forcing terms at interprocessor boundaries derived from the flow variables in the finest mesh). Alternatively, one can also avoid sending messages at the end of every stage in the Runge-Kutta time stepping. Past experience has shown [51] that these savings in communication cost are usually offset by a degradation in the convergence rate of the overall algorithm. Therefore, in the present implementation it was decided to allow message passing any time the flow variables were altered.

5 COMPUTATIONS OF STEADY AND TIME-DEPENDENT FLOWS

The flexibility and the efficiency of our multi-block solver is demonstrated by the results included in this section. Both steady and unsteady flow problems are presented.

5.1 STEADY EULER AND RANS ANALYSIS

In the first test case the steady inviscid solution capability is demonstrated on a typical business jet configuration depicted in Figure 3. The same configuration will serve as a test for a viscous analysis case and an inviscid design case. The mesh for the inviscid analysis contains 240 blocks and 4.2 million cells including halos. The geometry modeled consists of a wing-body-nacelle-pylon. The empennage was left out to simplify the initial grid generation. The nacelles are modeled as flow-through. The layout of the mesh topology is that of a general C-O. The mesh fidelity is such that a quick switch to Navier-Stokes calculations is possible by changing the spacing normal to the surface. The wing sweep is 20 degrees. Thus, with the thick airfoil sections featured in the design, it remains a challenge to contain wave drag at the moderate Mach numbers of its design point ($M = 0.75$ - 0.82). Figure 3 shows the configuration colored by calculated iso-C_P levels at $M = 0.82$ and $\alpha = 1.0$ degrees. Although they are not presented here, correlations of the wing pressure distributions have been obtained with experimental data. The comparisons with tunnel data are excellent except for a 5% difference in the location of the upper surface shock due to the omission of viscous effects. Using four multigrid levels, the solution presented in Figure 3 was obtained in 150 cycles and required 30 minutes of wall clock time using 32 processors of an IBM SP2 machine. The convergence criterion for this calculation was a reduction in the average residual of 5.2 orders.

The second example of inviscid analysis is carried out for a supersonic transport configuration. This configuration will serve also as an inviscid design case. Here a possible supersonic transport configuration was sized to accommodate 300 passengers with a gross take-off weight of 750,000 lbs. The supersonic cruise point is $M = 2.2$ with a C_L of 0.105. As can be seen in Figure 4, the planform has a break in the leading edge sweep. The inboard leading edge sweep is 68.5 degrees while the outboard is 49.5 degrees. Since the Mach angle at $M = 2.2$ is 63 degrees it is clear that some leading edge bluntness may be used inboard without a significant wave drag penalty. Airfoils with blunt leading edges were selected that range from 4% thick at the root to 2.5% thick at the leading edge break point. The symmetric initial airfoils were chosen with the purpose of accommodating spars at roughly 10% and 80% chord over the span up to the leading edge break. Outboard of the leading edge break where the wing sweep is ahead of the Mach cone, a sharp leading edge was used to avoid undue wave drag. The four-engine configuration features axisymmetric nacelles tucked close to the wing lower surface. This layout favors reduced wave drag by minimizing the exposed diverter area. However, it may be problematic because of the channel flows occurring in the juncture region of the diverter, wing, and nacelle at the wing trailing edge. The leading edge heights of the diverters are determined by the local boundary layer displacement thickness such that entrainment of boundary layer flow into the engines is avoided. Since the distances from the wing leading edge to the diverter leading edge are different for the two nacelles, this causes a corresponding diverter height difference. The computational mesh on which the analysis is run has 180 blocks and 1.5 million mesh cells. Again the nacelles are modeled as flow-through and a general C-O mesh topology is followed. Figure 4 shows the configuration colored by calculated iso-C_P levels at $M = 2.2$ and $C_L = 0.105$. Using four multigrid levels the solution was obtained in 100 cycles and required 16 minutes of wall clock time with 16 processors of an IBM SP2 machine. The convergence level obtained for this calculation was a reduction in the average residual of 3.8 orders.

The third analysis example corresponds to a steady Navier-Stokes solution for the transonic business jet configuration used in the first test case. This time the complete configuration is modeled, including the wing, body, nacelle, pylon, vertical tail, and horizontal tail. The mesh contains 240 blocks with 5.8 million cells including halos. It has the same general C-O topology with flow-through nacelles. For this calculation only the wing is treated as a no-slip boundary condition with the remaining solid surfaces modeled as inviscid type. The wall normal spacing of the first cell was such that at the flight conditions a $y^+ = 1$ would be attained at the half span trailing edge assuming a flat plate turbulent boundary layer. At the flight conditions ($M = 0.80$ and an altitude of 40,000 ft) the Reynolds number is 1.45 million/ft. A Baldwin-Lomax turbulence model is used in the demonstration and should be adequate for this attached flow condition. Figure 5 shows the iso-C_P solution at $M = 0.82$, Re $= 1.45$ million/ft and $C_L = 0.36$. As will be shown later in the design studies, this condition is above the design point for the configuration both in terms of Mach number and C_L. Figure 6 shows comparisons of the wing C_P

distribution between this Navier-Stokes solution and those obtained by the Euler calculation presented in the first test case at the same flight conditions (M and α). Note that the shock position has moved forward for the Navier-Stokes calculations; and though it is not presented here, this agrees well with experimental data. These results were obtained in 300 four-level multigrid cycles using 32 processors on an IBM SP2. The reduction in the average residual was 4.3 orders and the elapsed wall time was 3.25 hours.

5.2 TIME-RESOLVED HELICOPTER ROTOR

Rigid Rotor–Navier-Stokes Hover

A Navier-Stokes calculation was performed on the Caradonna rotor [52] at a collective pitch of 8 degrees and a tip Mach number of 0.877. Shock-free cases including viscous effects produced results that were very similar to the inviscid and experimental results and are not reproduced here. The grid used in this case was an H-H grid with 256 × 64 × 64 cells, with 128 cells on the surface of the airfoil in the chordwise direction and 48 cells in the spanwise direction. A Baldwin-Lomax turbulence model was used for a tip chord Reynolds number of $3,930,000$. Approximately 24 cells lie in the boundary layer of the rotor. This level of resolution has been shown to be satisfactory for these types of calculations when using a CUSP scheme [53,54,3]. Figure 7 shows experimental and numerical pressure coefficient distributions at different outboard radial locations of the blade. The most likely causes for this disagreement with experimental measurements are the inadequacy of the Baldwin-Lomax turbulence model for flow cases which include shock-boundary layer interaction such as the present case, and the differences between transition locations in the computation and experiment. Transition in this calculation was fixed at the leading edge of the blade, which may not correspond to the experimental location of transition (which was not specified in the experimental report). To reach an adequate level of convergence (five orders of magnitude reduction in the RMS residual of density), this calculation required 6.5 hours on 16 processors of an IBM SP-2. The computation was perfectly load balanced with 64 blocks of 32 × 32 × 16 cells.

Rigid Rotor–Euler Forward Flight

A series of time dependent calculations for the Caradonna rotor were also carried out to establish the feasibility of forward flight simulations. In this case, the problem is no longer symmetric and the full two bladed rotor (24 blocks) must be simulated. The freestream conditions are set appropriately, while the rotor and attached grid are rotated at the correct angular velocity. The collective pitch of the blade was set to 8 degrees. The tip Mach number for this flight condition was 0.628 while the advance ratio was 0.30.

Three calculations using 36, 72 and 144 time steps per revolution (corresponding to 10, 5 and 2.5 degrees per step) were carried out with a second order accurate discretization for the time derivative, and a refinement study was performed to verify the time accuracy. Between 20 and 25 multigrid cycles were used at each time step in order to converge the pseudo-time iteration to an acceptable level. The results are presented in Figure 8 which shows the lift coefficient of the rotor as a function of the azimuthal angle. As would be expected, the series of lift coefficient histories converges as the number of time steps per revolution is increased. Approximately 4-6 revolutions were needed to attain a periodic solution for the lift coefficient. For the 144 time step per revolution case, approximately 4 hours on 12 processors of an IBM SP-2 were used for each full revolution. Additional calculations not presented here indicate that, at lower advance ratios, more revolutions of the blade are needed in order to achieve a periodic solution. When the advance ratio is lowered, the wake is not convected as far away from the blade and therefore has a larger effect on the blade loading.

Aeroelastic Rotor–Euler Forward Flight

A preliminary aeroelastic calculation was attempted using the five bladed rotor at a tip Mach number of 0.628 and an advance ratio of 0.30. The same mesh used in the hover cases was repeated at 72 degree intervals resulting in a total mesh size of $5 \times 96 \times 32 \times 56 = 860,160$ cells with $5 \times 18 = 90$ blocks. Aeroelastic deflections were computed for all blades, but only modal deflections for one of these blades are reported. A simple structural deflection model was coupled to the flow equations to account for the aeroelastic properties of the blade.

A total number of 36 time steps per revolution was used allowing for the motion of the blades at 10 degree intervals. Within each time step, 50 multigrid cycles were used to fully converge the coupled fluid/aeroelastic system. Information between equation systems was exchanged after every 5 multigrid cycles of the flow solver.

Figure 9 shows the time evolution of three of the bending modes during the last computed rotor revolution. For the first mode of vibration, a negative modal coordinate represents an upward tip displacement. As one can see, after 6 revolutions the modal coordinates have nearly reached a periodic state. In particular, it is interesting to note that the maximum modal deflections are achieved on the retreating side, which is not unreasonable given the assumptions made in the modeling of the structural properties of the blades.

The problem was solved using 30 processors of an IBM SP-2 system (6 processors per blade), achieving almost perfect load balance (4% variation between processors). Nine hours were required to compute a total of 6 revolutions.

Further verification using more realistic structural models and experimental data is needed. Nevertheless, it is important to point out that this calculation indicates that forward flight rotor calculations including aeroelastic effects are indeed feasible on current high performance parallel computing platforms.

5.3 THE ADJOINT APPROACH TO OPTIMAL DESIGN

5.4 GENERAL FORMULATION

While a detailed derivation of the adjoint formulation for optimal design using either the Euler or the Navier–Stokes equations goes well beyond the scope of this paper, it is helpful to summarize the general description of the adjoint approach which has been thoroughly documented in references[9,10,55].

The progress of the design procedure is measured in terms of a cost function I which could be, for example, the drag coefficient or the lift to drag ratio. For flow about an aircraft configuration, the aerodynamic properties which define the cost function are functions of the flow-field variables (w) and the physical location of the boundary $\mathcal{F}$. Thus the cost function may be written as

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

while its first variation is given by

$$\delta I = \left[\frac{\partial I^T}{\partial w}\right] \delta w + \left[\frac{\partial I^T}{\partial \mathcal{F}}\right] \delta \mathcal{F}. \tag{11}$$

Using control theory, the governing equations of the flow-field are introduced as a constraint in such a way that the final expression for the gradient does not require multiple flow solutions. This corresponds to eliminating δw from (11).

Suppose that the governing equation R which expresses the dependence of w and $\mathcal{F}$ within the flow-field domain D is written as

$$R\,(w, \mathcal{F}) = 0. \tag{12}$$

Then its corresponding first variation can also be written

$$\delta R = \left[\frac{\partial R}{\partial w}\right] \delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right] \delta \mathcal{F} = 0, \tag{13}$$

since $R = 0$ must be satisfied at any point in the design space. Next, introducing a Lagrange multiplier ψ, we have after combining (11) and (13),

$$\begin{aligned} \delta I &= \frac{\partial I^T}{\partial w}\delta w + \frac{\partial I^T}{\partial \mathcal{F}}\delta \mathcal{F} \\ &- \psi^T \left(\left[\frac{\partial R}{\partial w}\right] \delta w + \left[\frac{\partial R}{\partial \mathcal{F}}\right] \delta \mathcal{F} \right) \\ &= \left\{ \frac{\partial I^T}{\partial w} - \psi^T \left[\frac{\partial R}{\partial w}\right] \right\} \delta w \\ &+ \left\{ \frac{\partial I^T}{\partial \mathcal{F}} - \psi^T \left[\frac{\partial R}{\partial \mathcal{F}}\right] \right\} \delta \mathcal{F}. \end{aligned} \tag{14}$$

Choosing ψ to satisfy the adjoint equation

$$\left[\frac{\partial R}{\partial w}\right]^T \psi = \frac{\partial I}{\partial w}, \tag{15}$$

the first term is eliminated, and we find that

$$\delta I = \mathcal{G}\delta\mathcal{F}, \tag{16}$$

where

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

The advantage is that (16) is independent of δw, with the result that the gradient of I with respect to an arbitrary number of design variables can be determined without the need for additional flow-field evaluations. In the case that (12) is a partial differential equation, the adjoint equation (15) is also a partial differential equation and determination of the appropriate boundary conditions requires careful mathematical treatment.

The computational cost of a single design cycle is roughly equivalent to the cost of two flow solutions since the the adjoint problem has similar complexity to that of the flow solution problem. When the number of design variables becomes large, the computational efficiency of the control theory approach over the traditional approach, which requires direct evaluation of the gradients by individually varying each design variable and recomputing the flow-field, becomes compelling.

Once equation (13) is established, an improvement can be made with a shape change

$$\delta\mathcal{F} = -\lambda\mathcal{G}$$

where λ is positive, and small enough that the first variation is an accurate estimate of δI. Then

$$\delta I = -\lambda \mathcal{G}^T \mathcal{G} < 0$$

After making such a modification, the gradient can be recalculated and the process repeated to follow a path of steepest descent until a minimum is reached. In order to avoid violating constraints, such as a minimum acceptable wing thickness, the gradient may be projected into an allowable subspace within which the constraints are satisfied. In this way, procedures can be devised which must necessarily converge at least to a local minimum.

The adjoint system is solved on the multi-block domain in a fashion identical to that used for the flow solution. Thus like the flow solver, the adjoint solver uses an explicit multistage Runge-Kutta-like algorithm accelerated by residual smoothing and multigrid. Inter-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. In the test cases to be presented in the next section NPSOL[56], a Sequential Quadratic Programming (SQP) optimization algorithm was used to drive the design process. References[9,10,11,12,57,58,59,55] give complete treatments of the details of how the adjoint equations are derived specifically for the Euler and Navier-Stokes equations as well as details regarding how the final gradient terms are evaluated. The references are also useful for an understanding of the options that are available in linking an adjoint method to various popular optimization algorithms. Finally, reference[60] shows some of the possible discretization schemes that can be used for the adjoint equations.

5.5 EXAMPLES OF DESIGN OPTIMIZATION

Numerical results will be presented for two classes of problems to demonstrate the versatility of our method. Reference[58] gives a treatment of the reliability of the flow solver as well as the ability of the adjoint method to provide accurate gradients very efficiently. The parallel speed-ups attained by the method have been demonstrated in reference[13], and are generally better than 90%.

Transonic Constrained Aircraft Design

As a first demonstration of the multi-block solver in the design mode, the transonic business jet configuration analyzed earlier is considered. In this Euler-based design case the initial multi-block mesh about the business jet wing, body, and nacelle has 72 blocks and 750,000 cells. Underlying geometry entities that are used to drive design changes include the wing with six defining stations and the fuselage. The initial configuration was designed for $M = 0.8$ and $C_L = 0.3$.

In the first design case (Test Case 1), a single-point constrained design is attempted in which the design Mach number is pushed from 0.80 to 0.82. The objective is to minimize configuration pressure drag at a fixed lift coefficient of 0.3 by modifying the wing shape. Eighteen Hicks-Henne design variables are chosen for five of the six defining sections for a total of 90 design variables. (The section at the symmetry plane is not being modified.) Spar thickness constraints are also enforced at each defining station at $x/c = 0.2$ and $x/c = 0.8$. Maximum thickness is forced to be preserved at $x/c = 0.4$ for all six defining sections. Each section is also constrained to have the thickness preserved at $x/c = 0.95$ to ensure an adequate included angle at the trailing edge. A total of 30 linear geometric constraints are imposed on the configuration. Figure 10 shows overlays of the C_p distributions at four stations along the wing for the initial configuration and final design after 5 NPSOL iterations. It is seen that the final result has reached a near-shock-free condition over much of the outboard wing panel. The drop in configuration pressure drag for this case was 22.5%. Noting that most of this drag reduction came from a decrease in wing wave drag implies that further improvements may be possible through the reshaping of other components.

Before proceeding to the next test case, it should be noted that this business jet design example is only representative of the potential for automated design, and is not intended to provide a design for actual construction. In fact, only 5 NPSOL steps were taken when considerably more steps could have improved the design further. More importantly, for the case of transonic design, the inclusion of viscous effects may prove to have an important impact on the optimized shape. In our future transonic studies, the viscous flow solver will be used.

Supersonic Constrained Aircraft Design

In the case of supersonic design, it is conjectured that as long as turbulent flow is assumed over the entire configuration, the inviscid Euler equations suffice for aerodynamic design. The pressure drag does not seem to be greatly affected by the inclusion of viscous effects, and a flat plate skin friction estimate of viscous drag is often a good approximation.

Here the configuration which was considered for the Euler analysis case presented in section 5.1 will be revisited. The mesh contains 180 blocks and 1.5 million mesh points, while the underlying geometry entities used to drive the design changes define the wing with 16 sectional cuts and the body with

200 sectional cuts. In this case, since we hope to optimize the shape of the wing, care must be taken to ensure that the nacelles remain properly attached with the diverter heights maintained. To accomplish this without introducing additional geometric complexity, the portions of the nacelles and diverters that are actually below the wing planform outline take their associated surface mesh point motion from their projected locations on the lower parametric wing surfaces.

The objective of the design is to reduce the drag at a single design point ($M = 2.2$, $C_L = 0.105$) by modifying the wing shape. Just as in the transonic cases, 18 design variables of the Hicks-Henne type are chosen for a given wing defining section. However, instead of applying them to all 16 sections, they are applied to 8 of the sections and then lofted linearly to the neighboring sections. Spar constraints are imposed for all wing defining sections at $x/c = 0.05$ and $x/c = 0.8$. An additional minimum thickness constraint is specified along the span at $x/c = 0.5$. A final thickness constraint is enforced at $x/c = 0.95$ to ensure a reasonable trailing edge included angle. An iso-C_p representation of the initial and final designs is depicted in Figure 11 for both the upper and lower surfaces.

It should be noted that the strong oblique shock evident near the leading edge of the upper surface on the initial configuration has been largely eliminated after 5 NPSOL design iterations. It is also seen that the upper surface pressure distribution in the vicinity of the nacelles has formed an unexpected pattern. These upper surface pressure patterns are conjectured to be the result of sculpting of the lower surface near the nacelles, which affects the upper surface shape through the thickness constraints. For the lower surface, the leading edge has developed a suction region while the shocks and expansions around the nacelles have been somewhat reduced. Figure 12 shows the pressure coefficients and (scaled) airfoil sections for four sectional cuts along the wing. These plots further demonstrate the removal of the oblique shock on the upper surface, and the addition of a suction region on the leading edge of the lower surface. The airfoil sections have been scaled by a factor of 2 so that shape changes may be seen more easily. Most notably, the section at 38.7% span has had the lower surface drastically modified such that a large region of the aft airfoil has a forward-facing portion near where the pressure spike from the nacelle shock impinges on the surface. The final overall pressure drag was reduced by 8%, from $C_D = 0.0088$ to $C_D = 0.0081$.

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

The development of a rapidly convergent multi-block flow solver, and its efficient implementation on parallel computers, makes the routine steady-state analysis of complete aircraft entirely feasible. It also enables automatic aerodynamic design optimization, and time-resolved calculations on complex configurations. The multi-block approach described in this paper has already been extended to compute steady and time-dependent incompressible flow. Future developments will focus on the implementation of advanced turbulence models, as well as the implementation of design optimization techniques based on the Reynolds Averaged Navier Stokes equations.

ACKNOWLEDGMENT

This work has benefited from the generous support of AFOSR under Grant No. AFOSR-91-0391, DOD/URI/ONR/ARPA under Grant No. N00014-92-J-1796 and the NASA-IBM Cooperative Research Agreement. The simulations of the flow past helicopter rotors were carried out by Scott Sheffer. Most of the grid generation work for both the steady-state analysis and design optimization problems was performed by Mark Rimlinger. Their contributions to this work are gratefully acknowledged by the authors.

1. A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper 91-1596*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.
2. J. J. Alonso and A. Jameson. Fully-implicit time-marching aeroelastic solutions. *AIAA paper 94-0056*, AIAA 32nd Aerospace Sciences Meeting, Reno, Nevada, January 1994.
3. J. J. Alonso, L. Martinelli, and A. Jameson. Multigrid unsteady Navier-Stokes calculations with aeroelastic applications. *AIAA paper 95-0048*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.
4. A. Belov, L. Martinelli, and A. Jameson. A new implicit algorithm with multigrid for unsteady incompressible flow calculations. *AIAA paper 95-0049*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.
5. S. Sheffer, J. Alonso, A. Jameson, and L. Martinelli. Time-accurate simulation of helicopter rotor flows including aeroelastic effects. *AIAA paper 97-0399*, January 1997.
6. N.P. Weatherill and C.A. Forsey. Grid generation and flow calculations for aircraft geometries. *J. Aircraft*, 22:855–860, 1985.

7. K. Sawada and S. Takanashi. A numerical investigation on wing/nacelle interferences of USB configuration. In *Proceedings AIAA 25th Aerospace Sciences Meeting*, Reno, NV, 1987. AIAA paper 87-0455.
8. J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.
9. A. Jameson. Aerodynamic design via control theory. *J. Sci. Comp.*, 3:233–260, 1988.
10. A. Jameson. Optimum aerodynamic design via boundary control. Technical report, AGARD FDP/Von Karman Institute Special Course on Optimum Design Methods in Aerodynamics, Brussels, April 1994.
11. J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA paper 95-0123*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.
12. J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA paper 96-0094*, AIAA 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1996.
13. J. Reuther, J.J. Alonso, M.J. Rimlinger, and A. Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on parallel computers. *AIAA paper 96-4045*, 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 1996.
14. B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flow. AIAA Paper 78-257, 1978.
15. L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. *AIAA paper 88-0414*, 1988.
16. L. Martinelli, A. Jameson, and E. Malfa. Numerical simulation of three-dimensional vortex flows over delta wing configurations. In M. Napolitano and F. Solbetta, editors, *Proc. 13th International Confrence on Numerical Methods in Fluid Dynamics*, pages 534–538, Rome, Italy, July 1992. Springer Verlag, 1993.
17. A. Jameson and L. Martinelli. Mesh refinement and modelling errors in flow simulation. *AIAA paper 96-2050*, AIAA 27th Fluid Dynamics Conference, New Orleans, June 1996.
18. W.P. Jones and B.E. Launder. The calculation of low-Reynolds-number phenomena with a two-equation model of turbulence. *Int. J. of Heat Tran.*, 16:1119–1130, 1973.
19. D.C. Wilcox. A half a century historical review of the k-ω model. AIAA Paper 91-0615, AIAA 29th Aerospace Sciences Meeting, Reno, NV, Jan-

uary 1991.

20. C.G. Speziale, E.C. Anderson, and R. Abid. A critical evaluation of two-equation models for near wall turbulence. AIAA Paper 90-1481, June 1990. ICASE Report 90-46.
21. R. Abid, C.G. Speziale, and S. Thangam. Application of a new k-τ model to near wall turbulent flows. AIAA Paper 91-0614, AIAA 29th Aerospace Sciences Meeting, Reno, NV, January 1991.
22. F. Menter. Zonal two-equation k-ω turbulence models for aerodynamic flows. AIAA Paper 93-2906, AIAA 24th Fluid Dynamics Meeting, Orlando, July 1993.
23. T.J. Coakley. Numerical simulation of viscous transonic airfoil flows. AIAA Paper 87-0416, AIAA 25th Aerospace Sciences Meeting, Reno, January 1987.
24. A. Jameson. Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. of Comp. Fluid Dyn.*, 4:171–218, 1995.
25. A. Jameson. Analysis and design of numerical schemes for gas dynamics 2, artificial diffusion and discrete shock structure. *Int. J. of Comp. Fluid Dyn.*, 5:1–38, 1995.
26. S. Tatsumi, L. Martinelli, and A. Jameson. A new high resolution scheme for compressible viscous flows with shocks. *AIAA paper* 95-0466, AIAA 33nd Aerospace Sciences Meeting, Reno, Nevada, January 1995.
27. R. Chipman and A. Jameson. Fully conservative numerical solutions for unsteady irrotational transonic flow about airfoils. AIAA Paper 79-1555, AIAA 12th Fluid and Plasma Dynamics Conference, Williamsburg, VA, July 1979.
28. A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. AIAA Paper 81-1259, 1981.
29. A. Jameson. Multigrid algorithms for compressible flow calculations. In *Second European Conference on Multigrid Methods*, Cologne, October 1985. Princeton University Report MAE 1743.
30. A. Jameson. Transonic flow calculations for aircraft. In F. Brezzi, editor, *Lecture Notes in Mathematics, Numerical Methods in Fluid Dynamics*, pages 156–242. Springer Verlag, 1985.
31. A. Rizzi and L.E. Eriksson. Computation of flow around wings based on the Euler equations. *J. Fluid Mech.*, 148:45–71, 1984.
32. A. Jameson, T.J. Baker, and N.P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. *AIAA paper 86-0103*, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 1986.

33. L. Martinelli. Calculations of viscous flows with a multigrid method. *Princeton University Ph.D. Thesis*, May 1987.

34. R.P. Fedorenko. The speed of convergence of one iterative process. *USSR Comp. Math. and Math. Phys.*, 4:227–235, 1964.

35. A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333–390, 1977.

36. W. Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20:291–306, 1978.

37. R.H. Ni. A multiple grid scheme for solving the Euler equations. *AIAA Journal*, 20:1565–1571, 1982.

38. A. Jameson. Solution of the Euler equations by a multigrid method. *Appl. Math. and Comp.*, 13:327–356, 1983.

39. M.G. Hall. Cell vertex multigrid schemes for solution of the Euler equations. In *Proc. IMA Conference on Numerical Methods for Fluid Dynamics*, Reading, April 1985.

40. A. Jameson. A vertex based multigrid algorithm for three-dimensional compressible flow calculations. In T.E. Tezduar and T.J.R. Hughes, editors, *Numerical Methods for Compressible Flow - Finite Difference, Element And Volume Techniques*, 1986. ASME Publication AMD 78.

41. D.A. Caughey. A diagonal implicit multigrid algorithm for the Euler equations. AIAA Paper 87-453, 25th Aerospace Sciences Meeting, Reno, January 1987.

42. W.K. Anderson, J.L. Thomas, and D.L. Whitfield. Multigrid acceleration of the flux split Euler equations. AIAA Paper 86-0274, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

43. P.W. Hemker and S.P. Spekreijse. Multigrid solution of the steady Euler equations. In *Proc. Oberwolfach Meeting on Multigrid Methods*, December 1984.

44. A. Jameson and D.J. Mavriplis. Multigrid solution of the Euler equations on unstructured and adaptive grids. In S. McCormick, editor, *Multigrid Methods, Theory, Applications and Supercomputing. Lecture Notes in Pure and Applied Mathematics*, volume 110, pages 413–430, April 1987.

45. M.H. Lallemand and A. Devrieux. A multigrid finite-element method for solving the two-dimensional Euler equations. In S.F. McCormick, editor, *Proceedings of the Third Copper Mountain Conference on Multigrid Methods, Lecture Notes in Pure and Applied Mathematics*, pages 337–363, Copper Mountain, April 1987.

46. G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3:27–43, 1963.

47. C.W. Gear. The numerical integration of stiff ordinary differential equations. Report 221, University of Illinois Department of Computer Science, 1967.
48. N. D. Melson, M. D. Sanetrik, and H. L. Atkins. Time-accurate navier-stokes calculations with multigrid acceleration. In *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, 1993.
49. 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.
50. 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.
51. J. J. Alonso, T. J. Mitty, L. Martinelli, and A. Jameson. A two-dimensional multigrid-driven Navier-Stokes solver for multiprocessor architectures. In *Proceedings of the Parallel CFD '94 Conference*, Kyoto, Japan, May 1994.
52. F. X. Caradonna and C. Tung. Experimental and analytical studies of a model helicopter rotor in hover. NASA TM 81232, AVRADCOM Research and Technology Laboratories, 1981.
53. A. Jameson and L. Martinelli. Mesh refinement and modeling errors in flow simulation. *AIAA paper 96-2050*, AIAA 27th Fluid Dynamics Conference, New Orleans, LA, July 1996.
54. S. Tatsumi, L. Martinelli, and A. Jameson. A new high resolution scheme for compressible viscous flow with shocks. *AIAA paper 95-0466*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.
55. A. Jameson, N. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *AIAA paper 97-0101*, January 1997.
56. P.E. Gill, W. Murray, M.A. Saunders, and M.A. Wright. User's guide for NPSOL (version 4.0). A FORTRAN package nonlinear programming. *Technical Report SOL86-2*, Stanford University, Department of Operations Research, 1986.
57. 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.
58. J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA paper 96-0094*, 34th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1996.

59. J. Reuther, A. Jameson, J. Alonso, M.J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *AIAA paper 97-0103*, January 1997.

60. J. J. Reuther. Aerodynamic shape optimization using control theory. *Ph. D. Dissertation*, University of California, Davis, Davis, CA, June 1996.

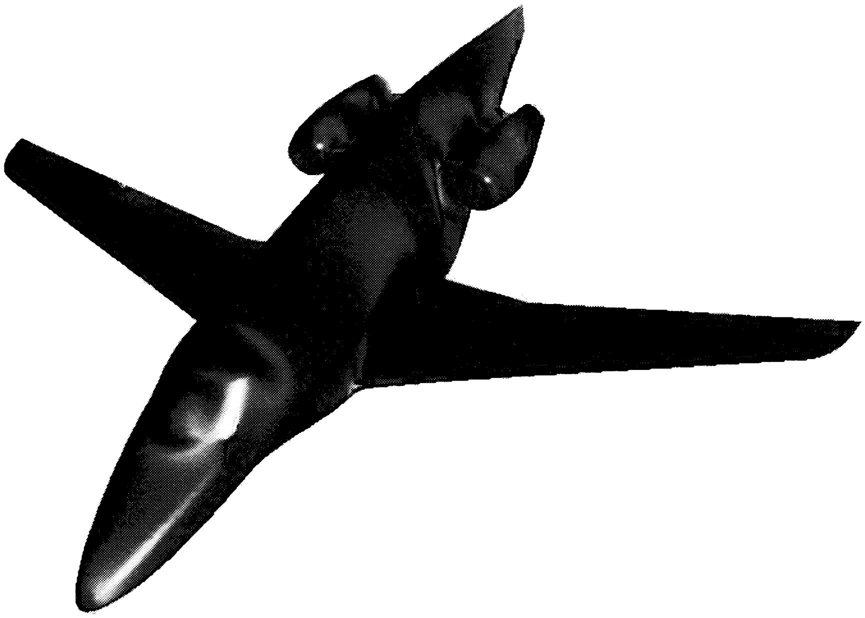


Figure 3: Business Jet Configuration. Iso-C_p Euler solution with 240 blocks and 4.2 million mesh points. $M_\infty = 0.82$, $\alpha = 1.0°$.

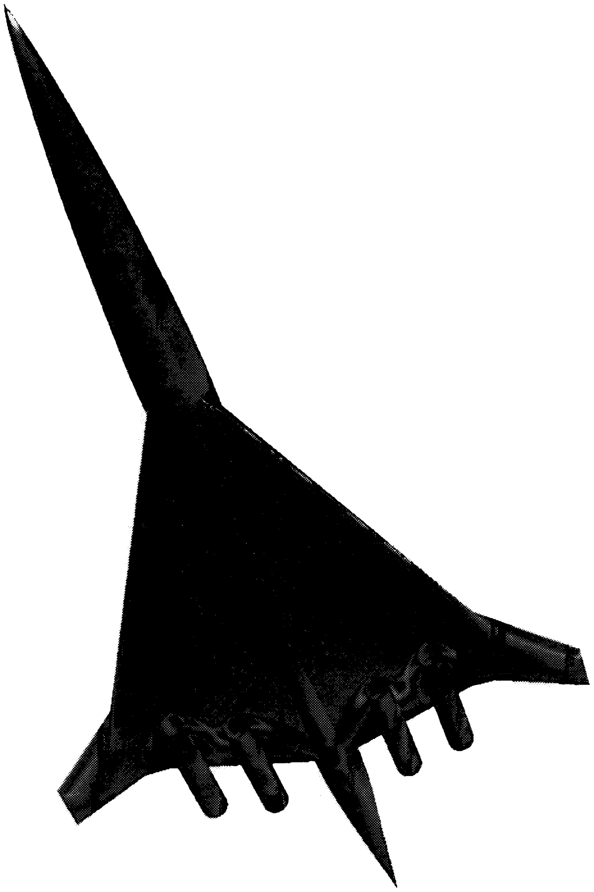


Figure 4: Supersonic Transport Configuration. Iso-C_p Euler solution with 180 blocks and 1.5 million mesh points. $M = 2.20$, $C_L = 0.105$.

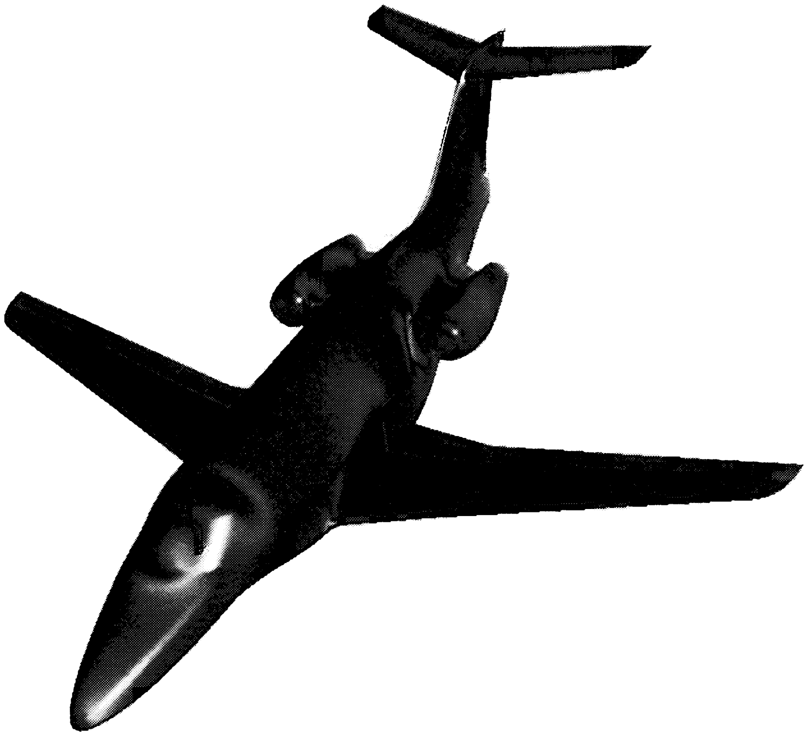
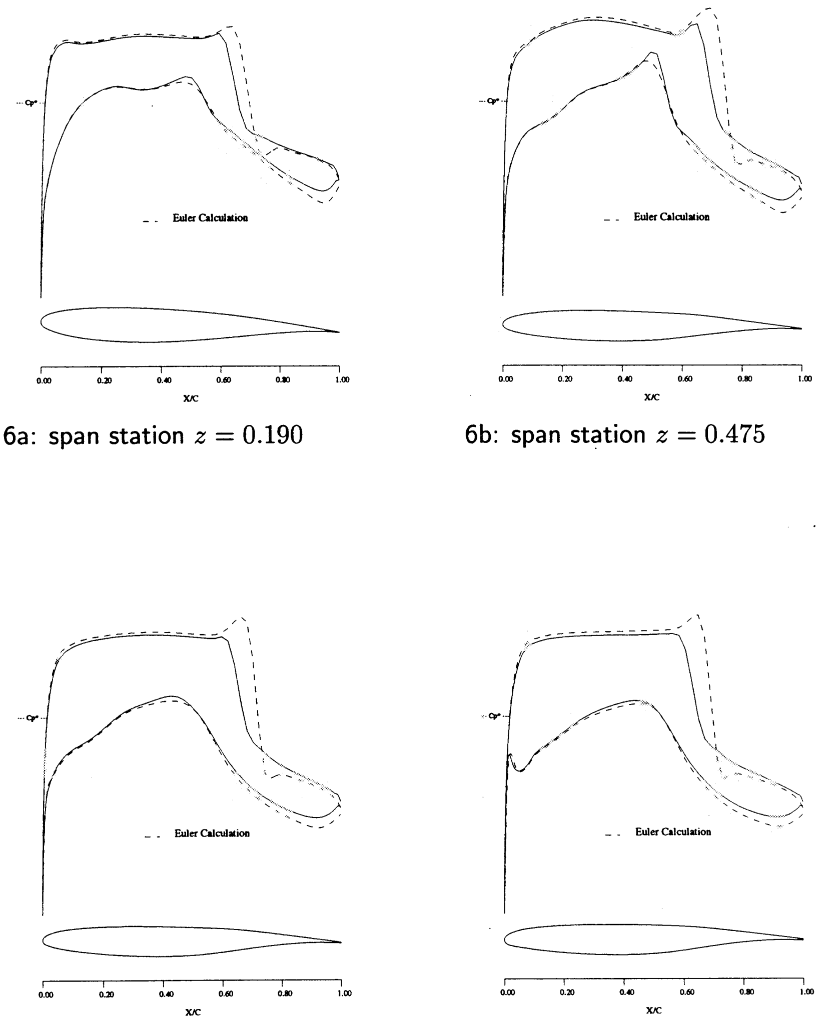


Figure 5: Business Jet Configuration. Iso-C_p Navier-Stokes solution with 240 blocks and 5.8 million mesh points. $M = 0.82$, $\alpha = 1.0^\circ$



6a: span station $z = 0.190$

6b: span station $z = 0.475$

6c: span station $z = 0.665$

6d: span station $z = 0.856$

Figure 6: Business Jet Configuration. Comparison between Euler and Navier-Stokes solutions $M = 0.82$, $C_L = 0.36$ - - -, Euler C_P; —, Navier-Stokes C_P.

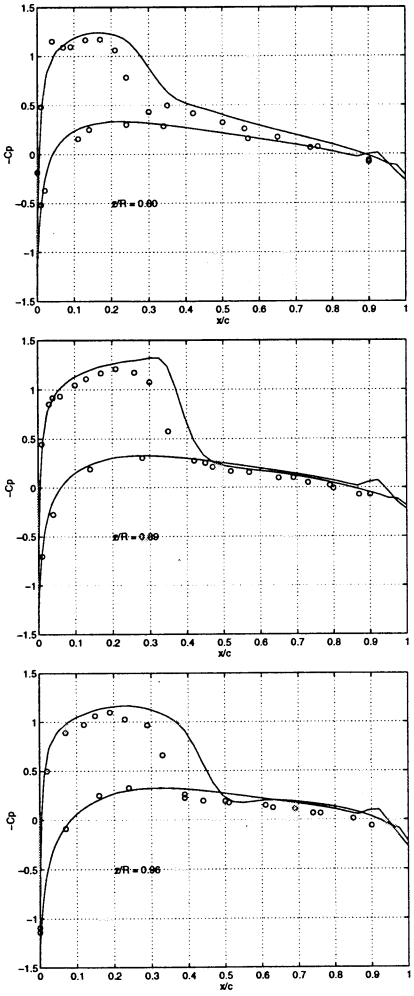


Figure 7: Pressure distribution on a rotor in hover, $\theta_c = 8°$, $M_t = 0.877$.

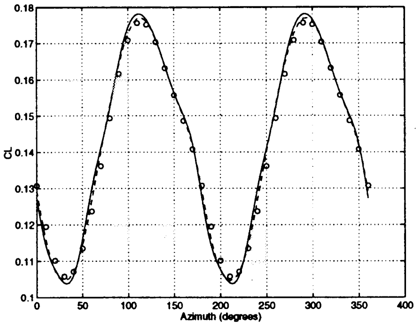


Figure 8: Two bladed rotor lift coefficient versus azimuth for advance ratio of 0.30 , o = 36 steps per revolution, - - = 72 steps per revolution, — = 144 steps per revolution.

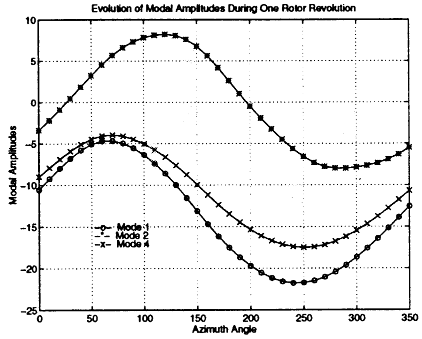
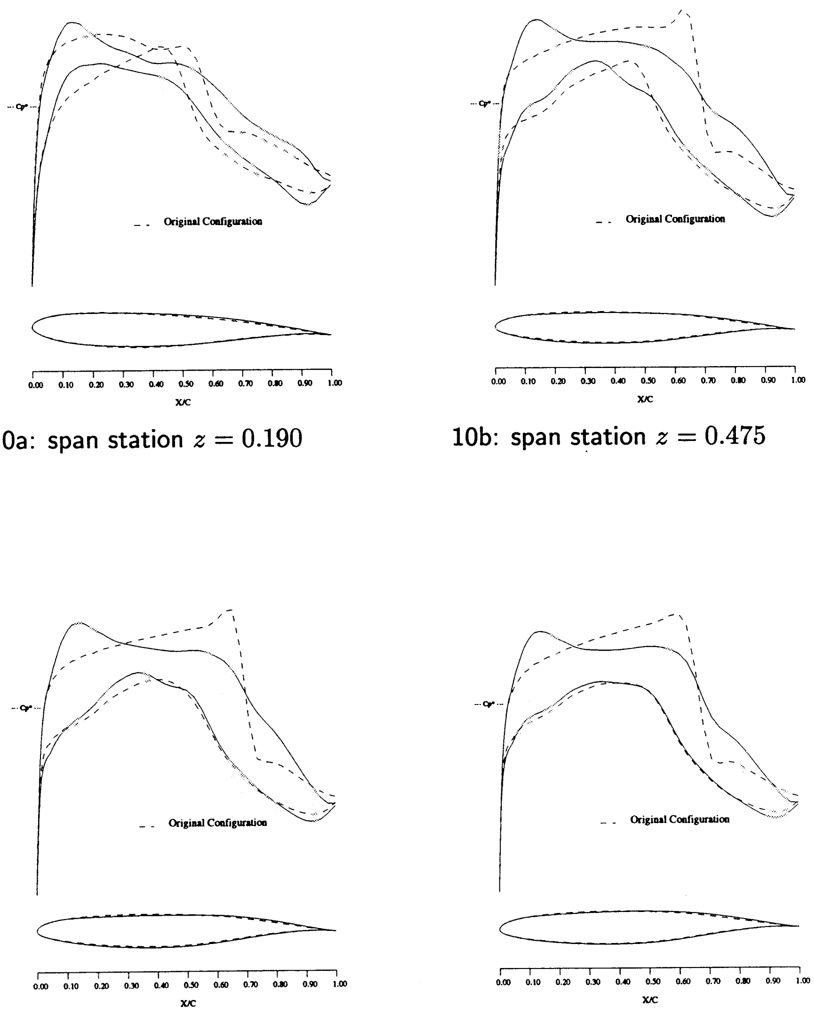


Figure 9: Time history of three bending modes in forward flight for a five bladed rotor.



10a: span station $z = 0.190$

10b: span station $z = 0.475$

10c: span station $z = 0.665$

10d: span station $z = 0.856$

Figure 10: Business Jet Configuration. Drag Minimization at Fixed Lift. $M = 0.82$, $C_L = 0.3$ 90 Hicks-Henne variables. Spar Constraints Active. - - -, Initial Pressures; —, Pressures After 5 Design Cycles.

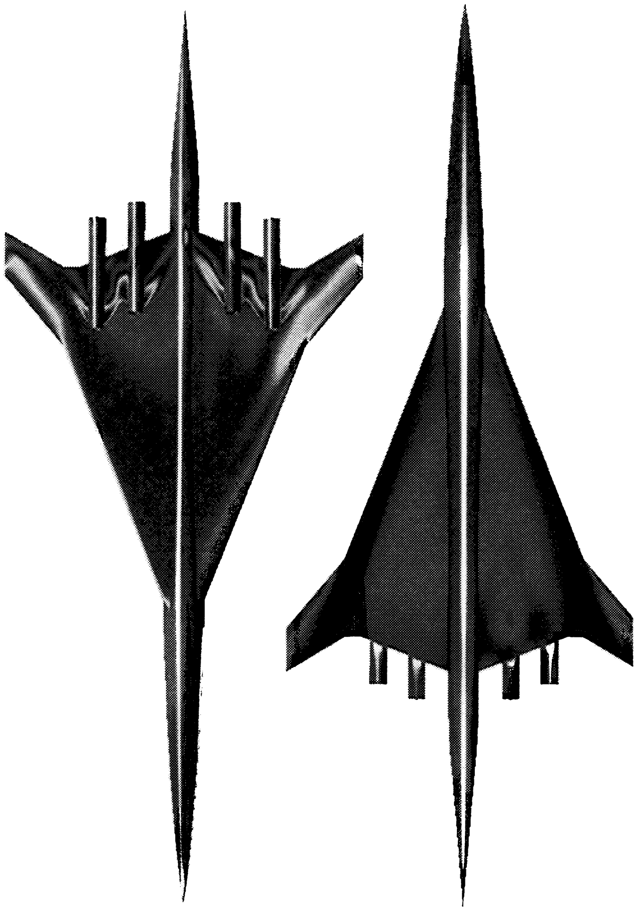
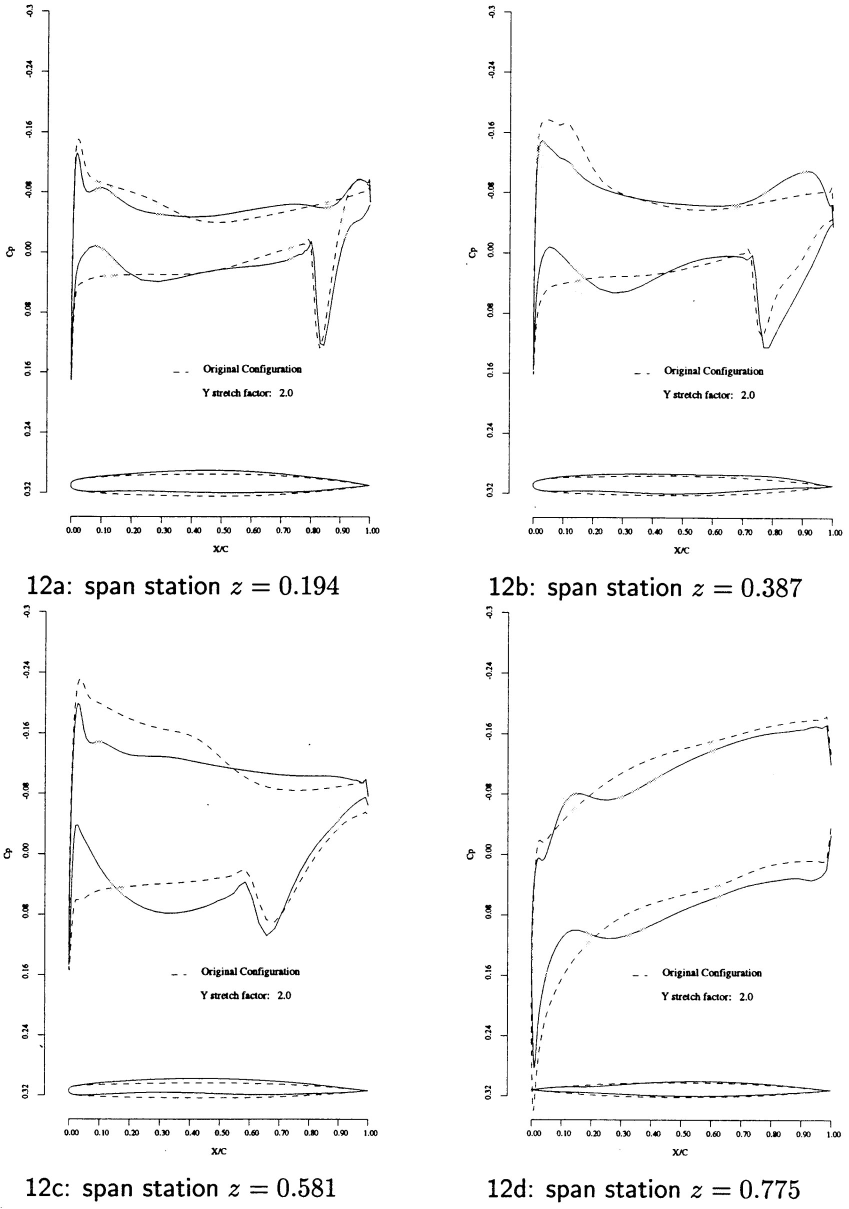


Figure 11: Supersonic Transport Configuration. Drag Minimization at Fixed Lift. Baseline solutions on the left half, optimized solutions on the right half. $M = 2.20$, $C_L = 0.105$



12a: span station $z = 0.194$

12b: span station $z = 0.387$

12c: span station $z = 0.581$

12d: span station $z = 0.775$

Figure 12: Supersonic Transport Configuration. Drag Minimization at Fixed Lift. $M = 2.20$, $C_L = 0.105$ 144 Hicks-Henne variables. Spar Constraints Active. - - -, Initial Pressures —, Pressures After 5 Design Cycles.