AIAA 2003–3429

High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model

Juan J. Alonso
Stanford University
Stanford, CA 94305, U.S.A.

Joaquim R. R. A. Martins
University of Toronto
Toronto, Ontario, Canada, M3H 5T6

James J. Reuther
NASA Ames Research Center
Moffett Field, CA 94035, U.S.A.

Robert Haimes and Curran A. Crawford
Massachusetts Institute of Technology
Cambridge, MA 02139, U.S.A.

16th AIAA Computational Fluid Dynamics Conference
June 23–26, 2003/Orlando, FL

AIAA 2003–3429

High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model

Juan J. Alonso*
Stanford University
Stanford, CA 94305, U.S.A.

Joaquim R. R. A. Martins†
University of Toronto
Toronto, Ontario, Canada, M3H 5T6

James J. Reuther‡
NASA Ames Research Center
Moffett Field, CA 94035, U.S.A.

Robert Haimes§ and Curran A. Crawford¶
Massachusetts Institute of Technology
Cambridge, MA 02139, U.S.A.

This paper presents two major additions to our high-fidelity aero-structural design environment. Our framework uses high-fidelity descriptions for both the flow around the aircraft (Euler and Navier-Stokes) and for the structural displacements and stresses (a full finite-element model) and relies on a coupled-adjoint sensitivity analysis procedure to enable the simultaneous design of the shape of the aircraft and its underlying structure to satisfy the measure of performance of interest. The first of these additions is a direct interface to a parametric CAD model that we call AEROSURF and that is based on the CAPRI Application Programming Interface (API). This CAD interface is meant to facilitate designs involving complex geometries where multiple surface intersections change as the design proceeds and are complicated to compute. In addition, the surface geometry information provided by this CAD-based parametric solid model is used as the common geometry description from which both the aerodynamic model and the structural representation are derived. The second portion of this work involves the use of the Finite Element Analysis Program (FEAP) for the structural analyses and optimizations. FEAP is a full-purpose finite element solver for structural models which has been adapted to work within our aero-structural framework. In addition, it is meant to represent the state-of-the-art in finite element modeling and it is used in this work to provide realistic aero-structural optimization costs for structural models of sizes typical in aircraft design applications. The capabilities of these two major additions are presented and discussed. The parametric CAD-based geometry engine, AEROSURF, is used in aerodynamic shape optimization and its performance is compared with our standard, in-house, geometry model. The FEAP structural model is used in optimizations using our previous version of AEROSURF (developed in-house) and is shown to provide realistic results with detailed structural models.

Introduction

DURING the past decade the advancement of numerical methods for the analysis of complex engineering problems such as those found in fluid dynamics and structural mechanics has reached a mature stage: many difficult numerically intensive problems are now readily solved with modern computer facilities. In fact, the aircraft design community is increasingly using computational fluid dynamics (CFD) and computational structural mechanics (CSM) tools to replace traditional approaches based on simplified theories and wind tunnel testing. With the advancement of these numerical analysis methods well underway, the focus for engineers is shifting toward integrating these analysis tools into numerical design procedures.

These design procedures are usually based on com-

*Assistant Professor, AIAA Member
†Assistant Professor, AIAA Member
‡Research Scientist, AIAA Associate Fellow
§Senior Research Scientist, AIAA Member
¶Graduate Student, AIAA Member

putational analysis methods that evaluate the relative merit of a set of feasible designs. The merit of a design is normally based on the value of an *objective function* that is computed using analysis methods such as CFD and CSM programs, while the design parameters are controlled by an optimization algorithm.

Despite revolutionary accomplishments in single-discipline applications, progress towards the development of high-fidelity, multidisciplinary design optimization (MDO) methods has been slow. The level of coupling between disciplines is highly problem dependent and significantly affects the choice of algorithm. Multiple difficulties also arise from the heterogeneity among design problems: an approach that is applicable to one discipline may not be compatible with the others.

An important feature that characterizes the various solution strategies for MDO problems is the allowable level of disciplinary autonomy in the analysis and optimization components. The allowable level of disciplinary autonomy is usually inversely proportional to the bandwidth of the interdisciplinary coupling. Thus, for highly coupled problems it may be necessary to resort to fully integrated MDO, while for more weakly coupled problems, modular strategies may hold an advantage in terms of ease of implementation.

In the particular case of high-fidelity aero-structural optimization, the coupling between disciplines has a very high bandwidth. Furthermore, the values of the objective functions and constraints depend on highly coupled multidisciplinary analyses (MDA). As a result, we believe that a tightly coupled MDO environment is more appropriate for aero-structural optimization.

The difficulty in formulating this kind of MDO problem is that there are significant technical challenges when implementing tightly coupled analysis and design procedures. Not only must MDA be performed at each design iteration but, in the case of gradient-based optimization, the coupled sensitivities must also be computed at each iteration.

Our previous work has presented and validated a tightly-coupled approach to high-fidelity aero-structural MDO that uses CFD and CSM. In addition, a coupled adjoint procedure was developed to produce inexpensive gradients of aero-structural cost functions. The main advantage of this approach resides in the fact that the cost of sensitivity calculations is completely independent of the total number of design variables in the problem and their effect on the system (they may influence either the aerodynamic shape of the aircraft or the shapes and thicknesses of the finite elements in the structure).

During the validation process we have come to realize that there are two significant problems that must be tackled in order to make this design environment truly multi-purpose and industrial-strength. Firstly, high-fidelity design involving multiple disciplines requires that a central, coherent, and accurate description of the geometry of all participating disciplines be carefully computed and stored. This central geoemtry repository changes as both the external shape of the configuration and the properties of the structure vary. Secondly, given that most of our prior experience was based on aerodynamic shape optimization work, we had used simplified structural finite element modeling techniques which we had developed in-house at Stanford. In order to prove the versatility of this new design environment, more accurate and larger finite element models need to be used for the description of the structural behavior.

All multidisciplinary design relies on a parameterization of the system that is being optimized: the shape of the outer mold line (OML) is often described as a function of a number of scalar parameters (design variables) such as the aspect ratio, span, reference area, airfoil shape perturbations, etc., while the shape of the structure typically includes its appropriate parameterization with the addition of elemental thicknesses and areas. As the models become more complex (complete aircraft configurations and detailed finite element models), the task of reconstructing the underlying geometry can become quite tedious and error prone: multiple intersections between aerodynamic surfaces have to be computed, surface meshes must be regenerated, and both CFD meshes and structural models have to be modified to accommodate the newly perturbed shape dictated by the current values of the design variables.

Although in the past we have relied on efficient methods that were developed in-house[11,18] to re-intersect components and reconstruct the shape of the configuration, we have been limited to typical wing-fuselage, wing-pylon, and pylon-nacelle configurations. The true value of a high-fidelity multidisciplinary environment will be brought to bear on *unconventional* configurations with different arrangements of lifting and non-lifting surfaces and with propulsive systems (nacelles, in particular) that are tightly integrated with the rest of the aircraft.

It seems obvious that this type of complex geometry management can be handled routinely by all Computer Aided Design (CAD) packages used in industry. Since the latest generations of all the major CAD packages support parametric design, we have opted to interface our design environment with CAD-based parametric aircraft models of arbitrary complexity. This approach can treat the aero-structural models we are interested in and will allow us to focus on the development of the modules that are more specific to realistic design.

The result of a high-fidelity aero-structural optimization is only as good as the quality of the individual components. For this reason, we have replaced our in-house finite element structural model,[8] which we used for all of our initial development with the full-featured, multi-purpose, Finite Element Analysis Program (FEAP) of Taylor.[20] FEAP includes all the necessary infrastructure to carry out linear and nonlinear analyses on very complicated models. In addition, it has interfaces to various parallel sparse matrix solvers that can reduce the computational burden presented by extremely large models. We have enhanced FEAP with a full Application Programming Interface (API) that allows us to integrate FEAP into our design environment. We have also enhaced FEAP so that it may be used to compute sensitivities of structural functionals via an adjoint method.

The following sections begin with a description of the aero-structural design framework we have previously developed and the kind of design problem that we intend to solve. We then present the philosophy and details of our new CAD-neutral interface together with the details of the parametric aircraft model that we use in our optimizations. After a brief dscription of the additions and modifications that we have made to the FEAP software, we present results of the aerodynamic optimization of a small supersonic jet using the new additions to our aero-structural framework. We finally conclude with some remarks regarding the usability and effectiveness of this new tool and with our view of the potential future use of this framework.

Aero-Structural Design Framework

The main objective of this framework is to calculate the sensitivity of a multidisciplinary cost function with respect to a number of design variables. The function of interest can be either the objective function (typically the drag coefficient at fixed lift or the empty weight of the structure) or any of the constraints specified in the optimization problem (such as element stresses or lumped versions of the element stresses like KS functions). In general, such functions depend not only on the design variables, but also on the physical state of the multidisciplinary problem. Thus we can write the function as

$$I = I(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $\boldsymbol{x}$ represents the vector of design variables and $\boldsymbol{y}$ is the state variable vector.

For a given set of design variables $\boldsymbol{x}$, the solution of the governing equations of the multidisciplinary system yields a state $\boldsymbol{y}$, thus establishing the dependence of the state of the system on the design variables. We denote these governing equations by

$$\boldsymbol{\mathcal{R}}\left(\boldsymbol{x}, \boldsymbol{y}\left(\boldsymbol{x}\right)\right) = 0. \tag{2}$$

The first instance of $\boldsymbol{x}$ in the above equation indicates the fact that the residual of the governing equations may depend *explicitly* on $\boldsymbol{x}$. In the case of a structural solver, for example, changing the size of an element has a direct effect on the stiffness matrix. By solving the governing equations we determine the state, $\boldsymbol{y}$, which depends *implicitly* on the design variables through the solution of the system.

Since the number of equations must equal the number of state variables, $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{y}$ have the same size. For a structural solver, for example, the size of $\boldsymbol{y}$ is equal to the number of unconstrained degrees of freedom, while for a computational fluid dynamics (CFD) solver, this is the number of mesh points multiplied by the number of state variables stored at each point. For a coupled system, $\boldsymbol{\mathcal{R}}$ represents *all* the governing equations of the different disciplines, including their coupling. This can be a rather large set of equations that need to be solved to obtain the equilibrium state of the multidisciplinary system.

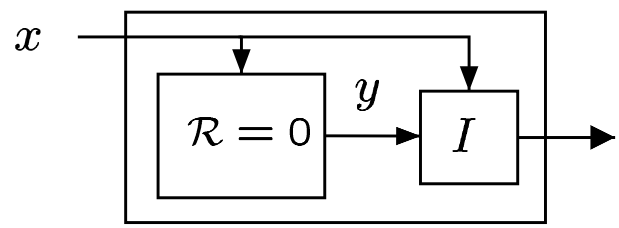

Fig. 1 Schematic representation of the governing equations ($\boldsymbol{\mathcal{R}} = 0$), design variables ($\boldsymbol{x}$), state variables ($\boldsymbol{y}$), and objective function (I), for an arbitrary system.

A graphical representation of the system of governing equations is shown in Figure 1, with the design variables $\boldsymbol{x}$ as the inputs and I as the output. The two arrows leading to I illustrate the fact that the objective function typically depends on the state variables and may also be an explicit function of the design variables.

When solving the optimization problem using a gradient-based optimizer, we require the *total* variation of the objective function with respect to the design variables, $\mathrm{d}I/\mathrm{d}\boldsymbol{x}$. As a first step towards obtaining this total variation, we use the chain rule to write the total variation of I as

$$\delta I = \frac{\partial I}{\partial \boldsymbol{x}}\delta \boldsymbol{x} + \frac{\partial I}{\partial \boldsymbol{y}}\delta \boldsymbol{y}. \tag{3}$$

If we were to use this equation directly, the vector $\delta\boldsymbol{y}$ would have to be calculated by solving the governing equations for each component of $\delta\boldsymbol{x}$. If there are many design variables and the solution of the governing equations is costly (as is the case for large coupled iterative analyses), using equation (3) directly can be impractical.

We now observe that the variations $\delta\boldsymbol{x}$ and $\delta\boldsymbol{y}$ in the total variation of the objective function (3) are not independent of each other since the perturbed system must always satisfy the governing equations (2). A relationship between these two sets of variations can be obtained by realizing that the variation of the residuals (2) must be zero, i.e.

$$\delta\boldsymbol{\mathcal{R}} = \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{y}}\delta\boldsymbol{y} = 0. \tag{4}$$

Since this residual variation (4) is zero we can add it to the objective function variation (3) without modifying the latter, i.e.

$$\delta I = \frac{\partial I}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \frac{\partial I}{\partial \boldsymbol{y}}\delta\boldsymbol{y} + \boldsymbol{\Psi}^T\left(\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{y}}\delta\boldsymbol{y}\right), \tag{5}$$

where $\boldsymbol{\Psi}$ is a vector of arbitrary scalars that we call the *adjoint vector*. This approach is identical to the one used in nonlinear constrained optimization, where equality constraints are added to the objective function, and the arbitrary scalars are known as *Lagrange multipliers*. The problem then becomes an unconstrained optimization problem, which is more easily solved.

We can now group the terms in equation (5) that contribute to the same variation and write

$$\delta I = \left(\frac{\partial I}{\partial \boldsymbol{x}} + \boldsymbol{\Psi}^T\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{x}}\right)\delta\boldsymbol{x} + \left(\frac{\partial I}{\partial \boldsymbol{y}} + \boldsymbol{\Psi}^T\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{y}}\right)\delta\boldsymbol{y}. \tag{6}$$

If we set the term multiplying $\delta\boldsymbol{y}$ to zero, we are left with the total variation of I as a function of the design variables and the adjoint variables, removing the dependence of the total variation on the state variables. Since the adjoint variables are arbitrary, we can accomplish this by solving the *adjoint equations*

$$\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{y}}\boldsymbol{\Psi} = -\frac{\partial I}{\partial \boldsymbol{y}}. \tag{7}$$

These equations depend only on the partial derivatives of both the objective function and the residuals of the governing equations with respect to the state variables. Since these partial derivatives do not depend on the design variables, the adjoint equations (7) only need to be solved once for each I and their solution is valid for all the design variables.

When adjoint variables are found in this manner, we can use them to calculate the total sensitivity of I using the first term of equation (6), i.e.,

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{x}} = \frac{\partial I}{\partial \boldsymbol{x}} + \boldsymbol{\Psi}^T\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{x}}. \tag{8}$$

The cost involved in calculating sensitivities using the adjoint method is practically independent of the number of design variables. After having solved the governing equations, the adjoint equations (7) are solved only once for each I, and the vector products in the total derivative in equation (8) are relatively inexpensive.

It is important to realize the difference between the total and partial derivatives in this context. Partial

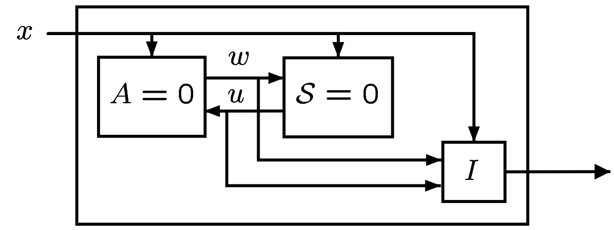

Fig. 2 Schematic representation of the aero-structural governing equations.

derivatives can be evaluated without regard to the governing equations. This means that the state of the system is held constant when partial derivatives are evaluated, except, of course, when the denominator happens to be a state variable, in which case all but that particular state variable can kept constant. Total derivatives, on the other hand, take into account the solution of the governing equations which change the state $\boldsymbol{y}$. Therefore, when using finite differences, the cost of computing partial derivatives is usually a very small fraction of the cost involved in estimating total derivatives.

The partial derivative terms in the adjoint equations are therefore relatively inexpensive to calculate. The cost of solving the adjoint equations is similar to that involved in the solution of the governing equations.

The adjoint method has been widely used in several individual disciplines and examples of its application include structural sensitivity analysis[1] and aerodynamic shape optimization.[9,15,16]

Aero-Structural Sensitivity Analysis

We now use the equations derived in the previous section to write the adjoint sensitivity equations specific to the aero-structural system. In this case we have coupled aerodynamic and structural governing equations, and two sets of state variables: the flow state vector and the vector of structural displacements. Figure 2 shows a diagram representing the coupling in this system. In the following expressions, we split the vectors of residuals, states and adjoints into two vectors corresponding to the aerodynamic and structural systems, i.e.

$$\boldsymbol{\mathcal{R}} = \begin{bmatrix} \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{S}} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{u} \end{bmatrix}, \quad \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{bmatrix}. \tag{9}$$

Using this notation, the adjoint equations (7) for an aero-structural system can be written as

$$\begin{bmatrix} \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{w}} & \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{u}} \\ \frac{\partial \boldsymbol{\mathcal{S}}}{\partial \boldsymbol{w}} & \frac{\partial \boldsymbol{\mathcal{S}}}{\partial \boldsymbol{u}} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{bmatrix} = -\begin{bmatrix} \frac{\partial I}{\partial \boldsymbol{w}} \\ \frac{\partial I}{\partial \boldsymbol{u}} \end{bmatrix}. \tag{10}$$

In addition to the diagonal terms of the matrix that appear when we solve the single-discipline adjoint equations, we also have off-diagonal terms that express the sensitivity of the governing equations of one discipline with respect to the state variables of the other. The residual sensitivity matrix in this equation is identical to that of the global sensitivity equations (GSE) introduced by Sobieski.[19] Considerable detail is hidden in the terms of this matrix and their calculation is not always straightforward. The meaning of all of these terms and the procedures that we have used in our work to compute them have been reported previously.[10–14] The reader is referred to these publications for in-depth explanations.

Note that the diagonal entries in (10) represent the single discipline adjoint solutions for both aerodynamics and structures. Our previous work has dealt with the development of aerodynamic adjoint formulations in detail[15–17] and we have in place the ability to perform aerodynamic shape optimization of complete configurations using our SYN107-MB solver. For linear structures, The adjoint operator is simply the transpose of the stiffness matrix. Since this matrix is symmetric, the linear structures adjoint problem is essentially the same as a structural solution with a different right hand side that is often called a *pseudo-load*.

The off-diagonal terms in (10) introduce the effects of the aero-structural coupling into the calculation of functional sensitivities. These are the terms that are responsible for the differences between truly-coupled aero-structural sensitivities and those obtained via sequential single-discipline optimizations.

The right-hand side terms in the aero-structural adjoint equation (10) depend on the function of interest, I. In our case, we are interested in two different functions: the coefficient of drag, C_D, and the KS function, a lumped version of the stress constraints. When $I = C_D$ we have,

- $\partial C_D/\partial \boldsymbol{w}$: The direct sensitivity of the drag coefficient to the flow variables can be obtained analytically by examining the numerical integration of the surface pressures that produce C_D.

- $\partial C_D/\partial \boldsymbol{u}$: This term represents the change in the drag coefficient due to the displacement of the wing while keeping the pressure distribution constant. The structural displacements affect the drag directly, since they change the wing surface geometry over which the pressure distribution is integrated.

When $I = \mathrm{KS}$,

- $\partial \mathrm{KS}/\partial \boldsymbol{w}$: This term is zero, since the stresses do not depend explicitly on the loads.

- $\partial \mathrm{KS}/\partial \boldsymbol{u}$: The stresses depend directly on the displacements since $\boldsymbol{\sigma} = \boldsymbol{S}\boldsymbol{u}$. This term is therefore equal to $[\partial \mathrm{KS}/\partial \boldsymbol{\sigma}]\, \boldsymbol{S}$.

Since the factorization of the full matrix in the coupled-adjoint equations (10) would be extremely costly, our approach uses an iterative solver, much like the one used for the aero-structural solution, where the adjoint vectors are *lagged* and the two different sets of equations are solved separately. For the calculation of the adjoint vector of one discipline, we use the adjoint vector of the other discipline from the previous iteration, i.e., we solve

$$\left[\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{w}}\right]^T \boldsymbol{\psi} = -\frac{\partial I}{\partial \boldsymbol{w}} - \left[\frac{\partial \boldsymbol{\mathcal{S}}}{\partial \boldsymbol{w}}\right]^T \tilde{\boldsymbol{\phi}}, \tag{11}$$

$$\left[\frac{\partial \boldsymbol{\mathcal{S}}}{\partial \boldsymbol{u}}\right]^T \boldsymbol{\phi} = -\frac{\partial I}{\partial \boldsymbol{u}} - \left[\frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{u}}\right]^T \tilde{\boldsymbol{\psi}}, \tag{12}$$

where $\tilde{\boldsymbol{\psi}}$ and $\tilde{\boldsymbol{\phi}}$ are the lagged aerodynamic and structural adjoint vectors. The final result given by this system, is the same as that given by the original coupled-adjoint equations (10). We call this procedure the *lagged-coupled adjoint* (LCA) method for computing sensitivities of coupled systems. Note that these equations look like the single discipline adjoint equations for the aerodynamic and the structural solvers, with the addition of forcing terms in the right-hand-side that contain the off-diagonal terms of the residual sensitivity matrix. Note also that, even for more than two disciplines, this iterative solution procedure is nothing but the well-known block-Jacobi method. This iterative procedure is guaranteed to converge to an aero-structural adjoint solution as long as the matrix representing the aero-structural operator in equation (10) remains diagonally dominant. Obviously, this may become an issue in problems that are strongly coupled. However, our experience has shown that even in the case of extremely large structural deflections (of the order of 1/3 of the span) the lagging approach continues to converge properly.

Once both adjoint vectors have converged, we can compute the final sensitivities of the objective function by using the following expression

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{x}} = \frac{\partial I}{\partial \boldsymbol{x}} + \boldsymbol{\psi}^T \frac{\partial \boldsymbol{\mathcal{A}}}{\partial \boldsymbol{x}} + \boldsymbol{\phi}^T \frac{\partial \boldsymbol{\mathcal{S}}}{\partial \boldsymbol{x}}, \tag{13}$$

which is the coupled version of the total sensitivity equation (8). For more details regarding the meaning and calculation of these partial derivative terms, the redear is again referred to our previous work.[11,12]

It must be noted that, as is the case of the partial derivatives in equations (10), all these terms in (8) can be computed without incurring a large computational cost, since none of them involve the solution of the governing equations.

In summary, once an aero-structural analsyis has been obtained, the aero-structural adjoint equations can be solved with a cost that is close to that of the multidisciplinary analysis itself. With both the aerodynamic and structural adjoint vectors in hand, the sensitivity of a given cost function with respect to an arbitrary number of design variables can be computed without significant additional cost. One additional coupled-adjoint calculation is required to compute the sensitivities of each additional cost function. Since the cost of the coupled adjoint procedure is proportional to the number of functions for which we are seeking sensitivities (while the cost remains independent of the total number of design variables) the coupled adjoint procedure is more suitable for problems with large numbers of design variables and a handful of cost functions. With appropriate use of constraint lumping approaches (K-S functions in our work, for example) this approach can offer very significant performance advantages with respect to the other available alternatives.

CAD-Based Parametric Modeling

As mentioned in the introduction, the geometric manipulations that are necessary to carry out multidisciplinary design on complete configurations can become quite complicated. As the design evolves, its shape changes, and the geometry must be regenerated repeatedly through a process of geometry creation, intersection, filleting, etc. Although in the past we have created our own geometry creation/intersection routines, their applicability has been limited to a number of *types* of configurations such as wing, wing-body, and wing-body empennage. Support for some classes of pylons and nacelles was also added and the resulting framework was used with success during the High-Speed Research program for complete HSCT configurations.

High-fidelity multidisciplinary techniques, however, can be used to design configurations that are non-standard with much more closely integrated wing/s, fuselages, propulsion system and with possibly multiple lifting surfaces. Although continued development of our in-house geometry framework was considered as a possibility, the advantages of a completely general, CAD-based, geometry engine far outweigh the development cost: arbitrarily complex parametric models can be handled by a design module (based on the CAD package of the designer's choice) which has been specifically developed for that purpose. Moreover, all geometric operations can be carried out more robustly and, in addition, the existence of this central geometry model can be used as the medium for the exchange of information between disciplines and as the provider of information for mesh generation/perturbation for all the participating disciplines.

It is a simple matter of economics (i.e., the enormous amount of work required) that the traditional coupling approach of constructing a direct interface for each software package to each available CAD engine becomes prohibitive, particularly for "smaller" disciplines where the interface development may consume more resources than the discipline solver itself. CAPRI[7] (Computational Analysis Programming Interface) provides a solution to the CAD dependency issue. Coupling to any supported CAD package is both unified and simplified by using the CAPRI definition of geometry (with topology) and its API to access the geometry and topological data. This CAD-vendor neutral API is more than just an interface to CAD data; it is specifically designed for the construction of complete analysis suites. CAPRI's 'Geometry Centric' approach allows access to the CAD part from within all sub-modules (grid generators, solvers and post-processors), facilitating such tasks as node enrichment by solvers and designation of mesh faces as boundaries (for the solver and the visualization system). CAPRI supports only manifold solids at its base level, eliminating problems associated with manually closing surfaces outside of the underlying CAD kernel. Multidisciplinary coupling algorithms can use the actual geometry as the medium to interpolate data from different grids. One clear advantage to this approach is that the geometry never needs to be translated and hence remains simpler and closed. The other major advantage is that writing and maintaining the grid generator (coupled to the CAD system) can be done once through CAPRI; all of the major CAD vendors are then automatically supported.

Geometry Creation and Modification

At the beginning of the CAPRI project there was always the notion that design functionality would be supported. At the time, it was thought that CAPRI would support the direct construction of three-dimensional solid geometry in order to allow for the modification of said geometry. As the geometry readers were being implemented, it became obvious that this would not be possible. Each CAD system deals with the low-level geometry construction in a very different manner. There was certainly not a common vendor-neutral perspective on direct construction. In fact, only those systems based on geometry kernels (and allowing the use of the kernel) could perform construction. Therefore, only if one programmed in *Parasolid*, Acis or OpenCASCADE could this kind of construction be performed.

As it turns out, this limitation was fortunate; another type of construction was available that could be driven by an API. Most modern CAD systems support the master-model concept of representing an object. A master model describes the sequence of topological operations to build the geometry of a solid model. At a basic level, it is an ordered list of extrude, revolve, merge, subtract and intersection operations. CAD systems support more meaningful abstractions, such as blends, fillets, drilled holes and bosses. When the CAD model is regenerated, the operation list is interpreted to sequentially build the geometry of the part. This gives the operator the ability to construct a family of parts (or assemblies) by building a single instance. Many of the operations used in the construction can be controlled by parameters that may be adjustable. By changing these values, a new member of the family can be built by simply following the prescription outlined in the master-model definition. We will refer to this type of modifiable CAD part as a parametric CAD part. This kind of part is fundamental to the generation of the full aircraft aerodynamic and structural meshes in the following sections.

The recipe may be simple, like a serial collection of primitive operations, but can also be complex, where operations are performed on previously or temporarily constructed geometry. The representation of this construction in most CAD systems is in the form of a tree, usually referred to as the *feature tree*. By supporting this method of construction, CAPRI provides both simple and powerful access to the CAD system. This approach is clearly outside the static view traditionally held of geometry. That is, this kind of access and control is not possible from any type of file transfer.

Within CAPRI, this tree is presented to the programmer in the form of branches. Each of these entities has an index to identify where in the tree the reference is made. All indices are relative (that is they can occur anywhere in the tree; the assignment is usually given during initial parsing of the CAD internal structures). There is a special branch always given the index zero: the root of the tree. Therefore, the entire tree may be traversed starting at the root and moving toward the end of each branch. The branches terminate at leaves (branches that do not contain any children). To aid in traversing the tree toward the root the parent branch is always available. Unlike simple binary trees, a branch in CAPRI's feature tree may contain zero or more children.

Currently, the structure of tree itself cannot be edited from within CAPRI (though this may change at some future release). However, some branches may be marked suppressible -these features may be turned off- in a sense removing that branch (and any children of the branch) from the regeneration. This is powerful in that it allows for defeaturing the model, so that it may be made appropriate for the type of analysis at hand. For example: if fasteners are too small for a fluid flow calculation, they may be easily suppressed (if the master-model was constructed with this in mind). After part regeneration the resultant geometry would be simplified and the details associated with the fasteners would not be expressed.

Parameters are those components of the master-model that contain values (and should not be confused with the geometric parameterization). CAPRI exposes all of the adjustable (non-driven) parameters found in the model. This is a separate list from the feature tree, but references back to the associated branch features where the values are used or defined. Parameters may be single- or multi-valued and can be booleans, integers, flooating-points or strings.

This CAD perspective on parametric building of parts and assemblies is fine for driving the part using simple parameters but is problematic for detailed shape design of the kind necessary in high-fidelity aero-structural calculations. For example, simple parameters may be used to define the planform of an aircraft, but are difficult to use to define the airfoil shape of the wing and tail components. The designer would need to expose the curve/surface definition at a very fine and detailed level (i.e. knot points as the parameters) to allow for the exact specification of shapes. CAPRI avoids placing this burden on the CAD designer by exposing certain curves as multi-valued parameters. These curves are obtained from independently sketched features in the model that later are used in solid generation as the basis for rotation, extrusion, blending and/or lofting. The curves can be modified, and, when regenerated, the new part expresses the changed shape(s). This functionality is critical for shape design in general and specifically aerodynamic shape design.

The authors have used this approach on a number of different environments and with several kinds of very different geometries[3,6] with success. Although the object of this paper is to use such an evironment for the parametric design of aircraft, other examples of the use of CAPRI include the geometry representation (for meshing purposes) of highly complicated turbine blades with or without cooling passages[6] and the multidisciplinary design optimization of power-generating wind turbines where the parametric CAD model is central to a number of different analyses (aerodynamics, structures, performance, cost, etc.) which were driven by several kinds of optimizers.[3]

In the current work, the CAPRI interface also appears to fulfill most of the requirements for geometry creation and retrieval in a gradient-based design environment:

- Control: it allows both global and detailed control over the shape of the geometry through the use of user-modifiable parameters and curves.
- Robustness: a large number of geometry modifications can be created through the design process without failure.
- Accuracy: all of the geometries produced represent the intended shape to a high degree of accuracy. This is particularly important when computing gradients, since very small geometry perturbations are required.
- Portability: the multidisciplinary design software that is developed to interact with CAD can do so

with all major vendors without the need to modify the code.

For this purpose, we have developed a CAPRI-based front-end for the CAD package Pro/Engineer that allows for all the geometry manipulations that we have been accustomed to in our multiblock design program SYN107-MB.[15,16]

AEROSURF Parallel/Distributed Geometry Server

During the process of aero-structural optimization, the aircraft geometry is modified according to a parameterization which is controlled by the values of a number of independent design variables. In our aerodynamic shape optimization work, for example, a number of wing and fuselage sections are modified to obtain an optimum-performing shape. In addition, scalar design variables such as the wing sweep angle or aspect ratio can be used to alter the global shape of the aircraft. In the past, we have used a geometry kernel which was developed to support a number of component intersections typical in traditional aircraft shapes.[18]

The advantage of this approach is that the computational cost of each geometry re-generation is extremely low (a fraction of a second). The cost of gradient computation in adjoint methods is presumably independent of the number of design variables. However, the gradient formula typically includes a volume integral term which requires that the volume mesh be perturbed according to the changes in the surface geometry. For large numbers of design variables, this cost can become significant if the geometry re-generation procedure is expensive. With our old geometry kernel, the impact of mesh regeneration was never more than a small portion of the overall cost of the optimization.

In this work, however, we are concerned with the use of an underlying CAD database to guide the process of the multidisciplinary design. At these early stages of the integration process, the cost of CAD geometry re-generation is still quite a bit higher than that of our old method: for the parametric CAD model of the aircraft that we will discuss in the following section, geometry regenerations that do not involve sectional changes (the airfoil and fuselage profiles are kept constant while the global shape of the configuration is altered) consume about 7 sec of CPU time on a single SGI R14000 600 Mhz processor. Due to perceived internal Pro/Engineer limitations with regenerations that involve sectional changes, if airfoil profiles or fuselage sections are changed, the actual cost of re-generation grows significantly to around 150 sec. These figures are obviously a function of both the complexity of the parametric model (how many operations are required to complete a re-generation) and the processor being used. Although the performance quoted above is expected to improve drastically (particularly for sectional re-generations) with different CAD packages, it is the current state of performance that has driven the design of our new geometry kernel, AEROSURF.

AEROSURF is a CAPRI-based, parallel and distributed application that acts as a broker between the CAD package and the simulation software that requests the variation in the geometry (the design code, for example). Figure 3 shows a schematic of the structure of AEROSURF and its relation to an aerodynamic shape optimization program. AEROSURF is built around the concept of a *geometry server* that constantly services multiple requests for parametric geometry. AEROSURF starts a number of instances of the CAD kernel (Pro/Engineer in our case, but the programming, via CAPRI, is independent of the CAD vendor) and maintains a series of queues for each of the CAD kernels that have been started.

The process is straightforward: AEROSURF is started on the server that is able to run the CAD software prior to the start of the design calculation. It itself starts a number N of CAD kernel instances that will do the actual re-generation work. After the design software is started (typically on a different parallel computer), AEROSURF awaits until the need for geometry re-generation arises. At periodic intervals during the design process, processors in the simulation request geometry re-generations. Multiple requests can and should be issed in parallel to minimize the overall cost of the geometry re-generation and to take advantage of the parallel nature of our system. AEROSURF receives these re-generation requests, attaches a unique identifier to them, and either forwards them to a free CAD kernel, or queues them up if no CAD kernels are available. As the CAD kernels complete their re-generation work, they forward their results (typically in the form of a surface grid) to AEROSURF, which, in turn, sends the results back to the requesting simulation processor. AEROSURF uses the Parallel Virtual Machine (PVM) interface to communicate with the simulation software. This enables the distributed running of the geometry and simulation components. All communication between AEROSURF and the simulation software occurs across the network (typically Ethernet). Since the size of each geometry description is (in our particular case) around 315Kb, the cost of transmission (in both directions) is practically negligible.

Since the work in each geometry re-generation is completely independent of the others, the problem is *embarrasingly* parallel and AEROSURF achieves almost perfectly linear scalability, which has been tested up to 32 simultaneous CAD kernels. In this way, the average time for a geometry re-generation can be as low as 7 sec/32 $\approx$ 0.22 sec for scalar parameter manipulation, and 150 sec/32 $\approx$ 4.7 sec for a re-generation with section changes. Note that we typically have

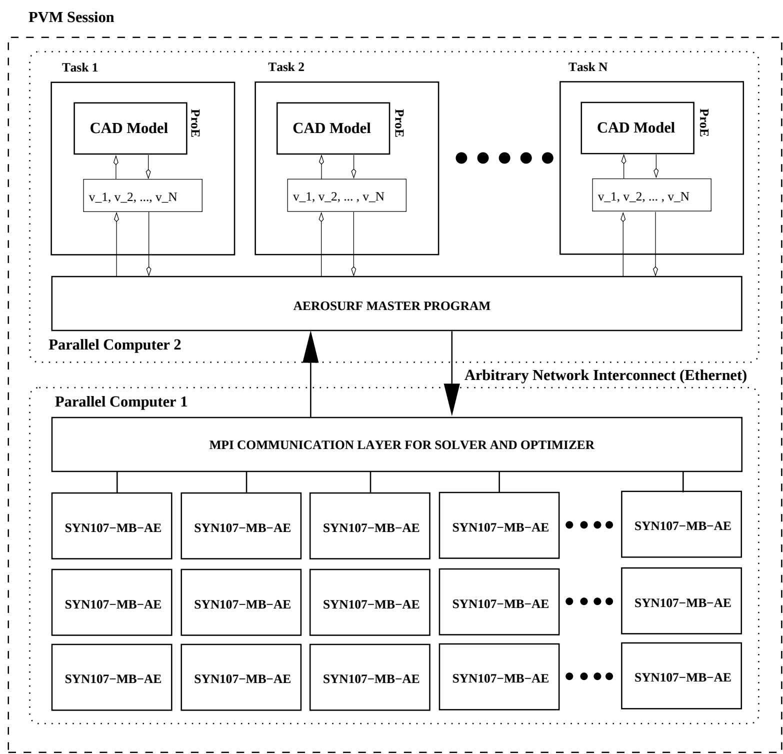

Fig. 3 Schematic representation of the CAPRI-based AEROSURF package.

run aerodynamic shape optimization in the past using O(300-400) design variables that mostly affect the various sections in the wing and fuselage components of the configuration. Given that the CAD re-generation costs are quite a bit higher than those for our old geometry engine, we are at present limited as to the number of design variables that we can handle. Work in the near term will address these performance shortcomings so that the CAD engine can fully replace the more limited (albeit faster), older version.

A typical aerodynamic shape optimization calculation requires a number of different geometry regenerations. After an initial re-generation to produce the baseline configuration (assuming that the initial values of the design variables are non-zero) the geometry needs to be perturbed in each and every one of the design variables (once per design cycle) when the gradient vector is completed. In addition, as we rely on the NPSOL[4] Sequential Quadratic Programming optimizer, after a gradient is computed, the objective function is minimized along this direction. During these line searches, the geometry is perturbed (typically three times to construct a quadratic fit) along the direction of the gradient. For a calculation with N_{dv} design variables, a typical requirement if to have approximately $N_{dv} + 5$ CAD regenerations per design iteration. Typical calculations use around 50 design iterations. The reader should be reminded that, because of the use of adjoint methods, we are able to afford the use of very large numbers of design variables. Consequently, the number of required CAD re-generations can indeed become very large.

Parametric Aircraft CAD Model

The basis of this work is an aicraft parametric model consisting of five components: fuselage, wing, vertical and horizontal tails (in a T-tail configuration) and nacelles. Although the nacelle definition is included in the parametric CAD model, it is ignored in all subsequent simulations. A total of 100 global shape parameters can be changed to alter the configuration. In addition, a total of 36 sections (15 airfoils on the wing, 3 in both the horizontal and vertical tails, and 15 fuselage sections) typically defined by 50-100 points each, can be modified to create exact geometry representations with the level of detail that is often required in aerodynamic shape optimization.

A top view of the parametric CAD model can be seen in Figure 4 where the top and bottom halves correspond to choices of section definitions and parameters that create, using the same CAD parametric part, both the baseline supersonic business jet for this work, and an approximate definition of a Boeing 717-200 jet. This figure is meant to show the versatility of the current parametric model which is able to cover a family of wide-ranging geometries where the components are arranged with the same topology. In the future we intend to create a *library* of parametric CAD models that span the range of our aircraft and spacecraft design interests (advanced supersonic configurations with closely integrated nacelles, blended-wing-body aircraft, standard transonic transport configurations, reusable launch vehicles, and even

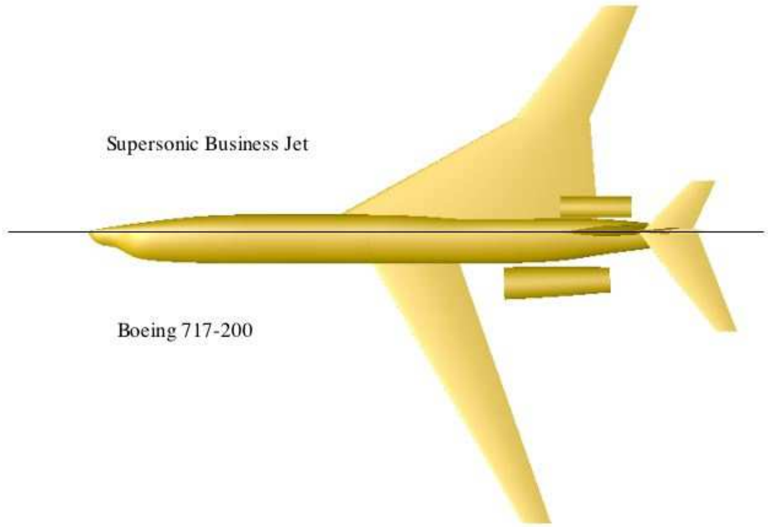

Fig. 4 Aircraft parametric model for two sets of variables defining both a supersonic business jet and the Boeing 717-200.

America's Cup yachts).

Each of the five components of the CAD model has a number of design variables that can alter the shape of that component (in addition to the section changes mentioned earlier). The three wing components have identical parameterizations: a single-crank planform model was adopted where the reference area, aspect ratio, taper ratio, sweep angle, location of the leading and trailing edge crank point, and leading and trailing edge extensions (inboard of the crank point) can be controlled independently. In addition, the twist angle at three spanwise stations (root, crank, and tip locations) can also be controlled. The fuselage shape can be arbitrarily defined at 15 stations, whose shape and location can be changed, thus permitting both configuration area ruling and fuselage camber modifications that can substantially help decrease both the volume- and lift-dependent portions of the wave drag of the aircraft. Finally, the nacelles are simply defined by their length and diameter, their toe-in and pitch orientation, their location, and the airfoil geometry that is revolved to create the actual nacelle.

The design problems in the Results section use this parametric CAD model and a subset of the available parameters to carry out the aerodynamic shape optimization of a Mach 1.5 supersonic business jet.

FEAP - Finite Element Analysis Program

The Finite Element Analysis Program (FEAP),[20] written by Prof. Robert L. Taylor at UC Berkeley, is a general purpose finite element package for the analysis of complex structures. The program includes the capability to construct arbitrarily complex finite element models using a library of one-, two-, and three-dimensional elements for linear and non-linear deformations. In addition, a number of material models (isotropic, orthotropic, plasticity, etc.) are available to model the constitutive properties of the materials that the structure is built of. Once the model is assembled, a number of solution procedures are available for linear, non-linear, and time-accurate problems. In addition, for very large non-linear structural models, interfaces are available for external parallel sparse solvers that can greatly improve the calculation turnaround times. A number of advanced time-accurate integration algorithms are also included with FEAP which can be of interest in the computation of aeroelastic responses and constraints.

The problem solution step is constructed using a command language concept in which the solution algorithm is completely written by the user. Accordingly, with this capability, each application may use a solution strategy which meets its specific needs. There are sufficient commands included in the system for applications in structural or fluid mechanics, heat transfer, and many other areas requiring solution of problems modeled by partial differential equations, including those for both steady-state and transient problems. Users also may add new routines for model description and command language statements to meet specific applications requirements. These additions may be used to assist in the generation of meshes for specific classes of problems or to import meshes generated by other systems.

Following our earlier approach with the in-house structural model, we have developed an interface for FEAP that accesses the major data structures (nodes, elements, materials, displacements, stresses, etc.) and allows other programs (simulation software, optimizers) to carry out the typical steps of a structural analysis. In addition, we have expanded FEAP to include various modules that are necessary for structural optimization. These modules include an adjoint solver and the calculation of the various terms that appear in the coupled aero-structural adjoint equation described earlier.

FEAP has surpassed our expectations and has behaved consistently and robustly in a number of test cases that we have encountered. However, in order to expedite our development, we have used finite differences in some of the FEAP-related terms of the coupled aero-structural adjoint equation. This can lead to both inaccuracies and poor computational performance. For that purpose, we intend, in the near future, to develop additional design modules for FEAP that provide analytic sensitivities of some of the most commonly used elements in aircraft structures. In this way we will by-pass the use of finite differences.

Structural Optimization

As a step toward the final goal of performing fully-coupled aero-structural optimization it was important to perform structural optimization studies for a wing of fixed outer-mold line subject to constant loads.

The structural model of the wing — shown in Figure 5 — is constructed using a wing box with six spars evenly distributed from 15% to 80% of the local chord at the root and tip sections. Ribs are distributed along the span at every tenth of the semispan. A total of 193 finite elements were used in the construction of this model. Appropriate thicknesses of the spar caps, shear webs, and skins were chosen to model the real structure of the wing. The structural analysis is performed by FEAP.

The objective of this optimization case is to minimize the weight of the structure by varying the thicknesses and cross-sectional areas of the finite elements while constraining the stresses in each of these elements to be less than the yield stress of the material.

Because there is a significant number of elements (albeit not close to a realistic structure), it can become computationally very costly to treat the stress constraints separately, especially in the case where the structural optimization is coupled with aerodynamic shape optimization.

The sensitivities of KS functions (see below) with respect to the finite-element sizes are efficiently computed by using an adjoint method.[1,11] Since we are using an adjoint method for computing sensitivities, it is convenient to lump the individual element stresses using Kreisselmeier–Steinhauser (KS) functions. Suppose that we have the following constraint for each structural finite element,

$$g_i = 1 - \frac{\sigma_i}{\sigma_{yield}} \geq 0, \qquad (14)$$

where σ_i is the element von Mises stress and σ_{yield} is the yield stress of the material. The corresponding KS function is defined as

$$\mathrm{KS}\left(g_i(x)\right) = -\frac{1}{\rho} \ln \left[\sum_i \mathrm{e}^{-\rho g_i(x)} \right]. \qquad (15)$$

This function represents a lower bound envelope of all the constraint inequalities and ρ is a positive parameter that expresses how close this bound is to the actual minimum of the constraints. This constraint lumping method is conservative and may not achieve the exact same optimum that a problem treating the constraints separately would. However, the use of KS functions has been demonstrated and it constitutes a viable alternative, being effective in optimization problems with thousands of constraints.[2]

The structure of the wing is parameterized with a total 193 design variables representing the thickness of the shells that model the spars, ribs and skins, and the cross-sectional area of the frames that model the caps for the spars. Although the structural model is small, the design problem is rather large in comparison to typical design space sizes. This compromise represents the ideal spot for early development work since additional model complexity and size would only increase the execution time, but would not increase the complexity of the design problem.

The structural optimization is performed by SNOPT, a nonlinear optimization package.[5] The optimization result shown in Figure 6 took 357 major iterations to find the optimum solution. Note that the structure is not as fully stressed as we would expect for a fully optimized structure. This is due to the conservative character of the KS function.

Results of Aerodynamic Shape Optimization

The objective in this section is to both demonstrate and validate the outcome of our CAD-based aerodynamic shape optimizations. For that purpose, we have designed an efficient baseline configuration with a cruise weight of 100,000 lbs, flying at a cruise altitude of 55,000 ft at $M_\infty = 1.5$. The cruise $C_L = 0.1$ is forced to remain constant throughout our optimizations. The configuration wing planform is designed with a cranked delta wing shape with the inboard leading edge swept behind the Mach cone, while the outboard leading edge remains supersonic. The fuselage was sized to accommodate 10 passengers and area ruling was applied in an approximate manner. An Euler analysis of this configuration results in an inviscid cruise drag coefficient of $C_D = 0.00858$.

Aerodynamic Shape Optimization Using In-House Geometry Engine

Our first design test case modifies the detailed shape of the wing and fuselage in order to minimize the inviscid drag of the configuration at a constant $C_L = 0.1$. Although the wing planform remains fixed, the twist

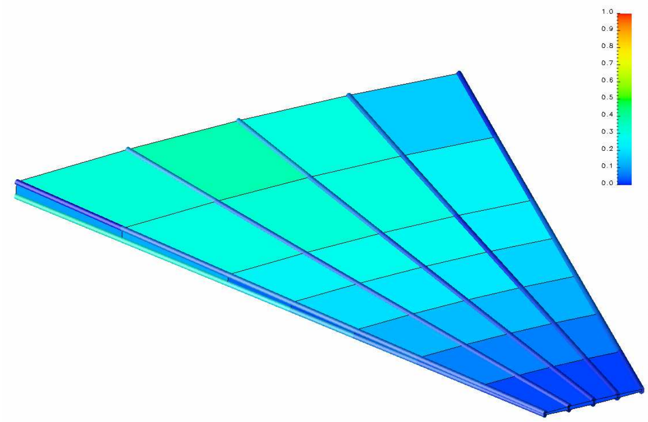

Fig. 5 Baseline structure.

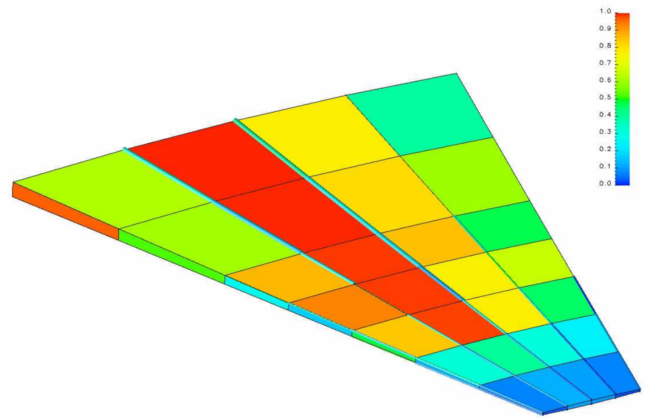

Fig. 6 Optimized structure.

and shape of 7 defining stations evenly spread along the span can be altered. At each of these defining stations, in addition to the twist variable, 10 Hicks-Henne bump functions are added on the top and bottom surfaces. Additional leading and trailing edge camber functions are also used. In order to prevent improvements in performance that simply result from a decrease in the wing volume, a total of 30 thickness constraints are added at six of the defining stations so that the thickness may not decrease at the 2, 25, 50, 75, and 98% chord locations.

The fuselage has circular cross-sections and its volume is constrained to remain constant. A total of 11 fuselage camber design variables are added to the optimization problem. Including the wing shape variables, a total of 136 design variables are considered in this model with 30 linear constraints for the wing thicknesses.

Using the NPSOL optimizer, and after 9 design iterations, the drag of the configuration decreases by 9% to $C_D = 0.00781$. This improvement in drag coefficient has been achieved without decreases in either the wing or fuselage volume and it is about evenly divided between improvements due to fuselage shape perturbations and wing shape perturbations. This fact can be confirmed since we ran an identical optimization without the fuselage design variables which achieved close to 50% of the drag improvement reported here.

The optimizer changes the shape of the fuselage quite drastically: the originally axisymmetric body is given both fore and aft camber, presumably to spread the lift produced by the fuselage in the streamwise direction so as to minimize the contribution of the lift-dependent wave drag. The wing geometry has also changed drastically: the originally untwisted wing now has nearly 0.5 deg of washout. In addition, the baseline configuration was created using a 4% thick RAE 2822 airfoil in the inboard wing panel and a 3% thick biconvex airfoil in the outboard panel. The wing shape design variables have drastically reduced the camber distribution on the wing inboard sections (although not eliminated it completely) and they have also modified the shape of the outboard wing panels.

Side and top views of the resulting design with Mach number contours superimposed (varying from $M = 1.4$ to $M = 1.7$, blue to red) can be seen in Fig. 7.

Aerodynamic Shape Optimization Using CAD-Based AEROSURF

The design problem setup in this case is identical to the one before, except for two main differences. Firstly, all surface re-generations required during the design process are handled by our CAD-based AEROSURF geometry engine. Secondly, in the interest of minimizing the CAD re-generation times at this stage of validation process, we decided to maintain the original shapes of all of the sections in the geometry (both fuselage and wing) and modify the twist distribution on the wing and the fuselage camber. Since the parametric CAD model was constructed with control over the twist angle of the wing root, crank, and tip sections, only three twist design variables are used here. A piecewise linear variation is implied between these three defining stations. On the fuselage, 9 camber variables such as the ones used before are used, for a total of 12 design variables in this test case. Since the wing sections are not allowed to vary, it is not necessary to impose thickness constraints.

After 6 design iterations and a total of 124 CAD re-generations, the coefficient of drag of the configuration is reduced to $C_D = 0.00809$, an improvement of 5.7% compared with the earlier value of 9%. Since only the wing twist distribution is altered, we can observe that the wing de-cambering is responsible for the remaining 3.3% improvement. This is a very significant amount in supersonic design and it highlights the need to use detailed shape parameterizations to obtain the true optimum of such aircraft systems.

Figure 8 shows several views of the resulting design. It is clear that the optimizer has chosen to shape the fuselage in a very similar way to the previous case, thus achieving the improvements that derive from lift re-distribution. Detailed examination of the values of the fuselage camber design variables reveals that this is indeed true: the variations in fuselage camber are very close to each other. Since the wing shape is not allowed to change, the optimizer changes the twist distribution much more drastically than in the previous case to achieve changes in lift that would have otherwise resulted from the combination of twist and de-cambering. The total washout for the wing is now almost -1.2 deg.

The results are very close to our expectations and serve as validation of the CAD-based AEROSURF geometry engine. In the near future we expect to extend this validation to use the same number of design variables as in the first optimization and will validate both the gradients and the results obtained. Since the surface shape parameterization of the in-house and CAD-based engines are slightly different, exact agreement is not expected. However, the outcome of the design is likely to be quite close.

Conclusions and Future Work

In this paper we have reviewed the basis of our coupled-adjoint aero-structural design framework and have provided details of the formulation of the optimization problem. The coupled-adjoint design environment allows for the calculation of coupled aero-structural sensitivities of aerodynamic and structural cost functions with computational cost that is independent of the number of design variables. In order to further the applicability of this design environment, we have pursued the improvement of our geometry management using a CAD-based *geometry server* application. This geometry server, AEROSURF, is made possible in a CAD-vendor-neutral way through the use of the CAPRI API. In addition, we have replaced our structural analysis and design capability by the FEAP solver of Taylor (UC Berkeley). FEAP has been shown to produce accurate and realistic results in both analysis and design environments. Finally, aerodynamic shape optimizations have been carried out using the old and new geometry kernels to validate the use of the more sophisticated geometry re-generation techniques.

At the moment we are pursuing the full aero-structural optimization of the complete supersonic business jet configuration which can be seen in Fig 9 below. This work is now in its preliminary stages and will be presented in the near future.

Acknowledgements

The authors wish to acknowledge the generosity of Prof. Robert L. Taylor (Civil Engineering, UC Berkeley) for allowing the use and modification of his FEAP structural solver. In addition, we would like to thank Dr. David Saunders (NASA Ames Research Center) for his help in the preparation of both configurations used in this work. The first author wishes to acknowledge the support of the Air Force Office of Scientific Research under grant AF F49620-01-1-0291.

References

[1]Howard M. Adelman and Raphael T. Haftka. Sensitivity analysis of discrete structural systems. *AIAA Journal*, 24(5):823–832, May 1986.

[2]Mehmet A. Akgn, Raphael T. Haftka, K. Chauncey Wu, and Joanne L. Walsh. Sensitivity of lumped constraints using the adjoint method. AIAA Paper 99-1314, April 1999.

[3]Curran A. Crawford. An integrated CAD methodology applied to wind turbine optimization. Technical report, Cambridge, MA 02139, June 2001.

[4]P. E. Gill, W. Murray, M. A. Saunders, and M. H. 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.

[5]Philip E. Gill, Walter Murray, Michael A. Saunders, and M. A. Wright. *User's Guide for SNOPT 5.3: A Fortran Package for Large-scale Nonlinear Programming*. Systems Optimization Laboratory, Stanford University, California, 94305-4023, December 1998. Technical Report SOL 98-1.

[6]R. Haimes and C. Crawford. Unified geometry access for analysis and design. Technical report, June 2003.

[7]R. Haimes and G. Follen. Computational analysis programming interface. Technical report, June 1998.

[8]M. E. Holden. *Aeroelastic Optimization Using the Collocation Method*. PhD thesis, Stanford University, Stanford, CA 94305, 1999.

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

[10]J. R. R. A. Martins, J. J. Alonso, and J. Reuther. Aero-structural wing design optimization using high-fidelity sensitivity analysis. In *Proceedings — CEAS Conference on Multidisciplinary Aircraft Design Optimization, Cologne, Germany*, pages 211–226, June 2001.

[11]J. R. R. A. Martins, J. J. Alonso, and J. Reuther. High-fidelity aero-structural design optimization of a supersonic business jet. AIAA Paper 2002-1483, April 2002.

[12]J. R. R. A. Martins, J. J. Alonso, and J. Reuther. High-fidelity aero-structural design optimization of a supersonic business jet. AIAA Paper 2002-1483, April 2002.

[13]J. R. R. A. Martins, J. J. Alonso, and J. Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Submitted for Publication, Journal of Optimization in Engineering*, 2003.

[14]J. R. R. A. Martins, J. J. Alonso, and J. Reuther. High-fidelity aero-structural design optimization of a supersonic business jet. *Accepted for Publication, Journal of Aircraft*, 2003.

[15]J. Reuther, J. J. Alonso, A. Jameson, M. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers: Part I. *Journal of Aircraft*, 36(1):51–60, 1999.

[16]J. Reuther, J. J. Alonso, A. Jameson, M. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers: Part II. *Journal of Aircraft*, 36(1):61–74, 1999.

[17]J. Reuther, J. J. Alonso, J. C. Vassberg, A. Jameson, and L. Martinelli. An efficient multiblock method for aerodynamic analysis and design on distributed memory systems. AIAA Paper 97-1893, June 1997.

[18]J. Reuther, M.J. Rimlinger, and D. Saunders. Geometry driven mesh deformation. Technical Report Vol. 1, Part 2, September 1999.

[19]J. Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28(1):153–160, January 1990.

[20] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method, 5th Edition, Vol. I.* Butterworth-Heinemann, 2000.

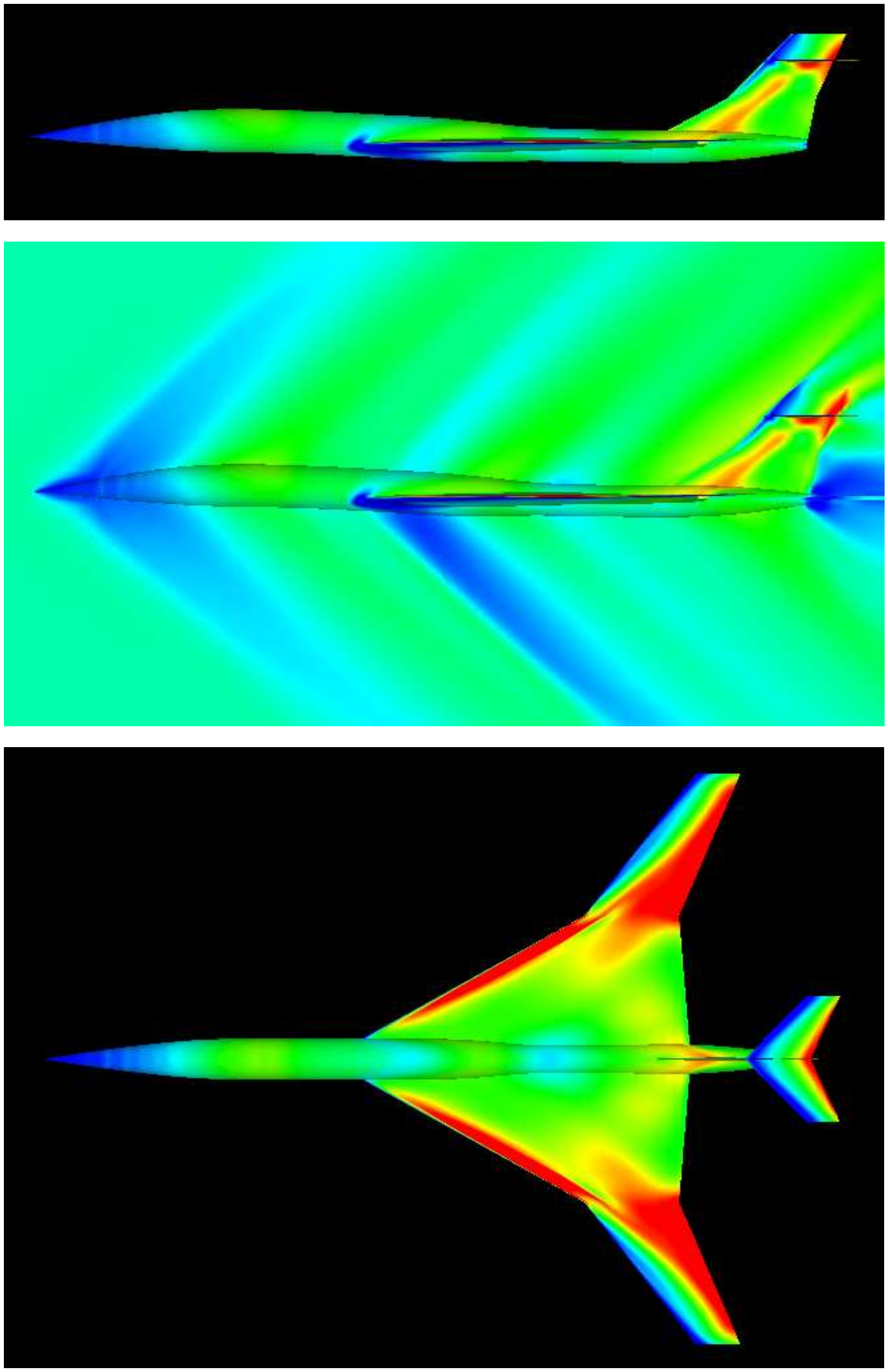

Fig. 7 Optimized aerodynamic configuration. $C_L = 0.1$, $C_D = 0.0078$, $M_\infty = 1.5$. **136 design variables using in-house geometry engine.**

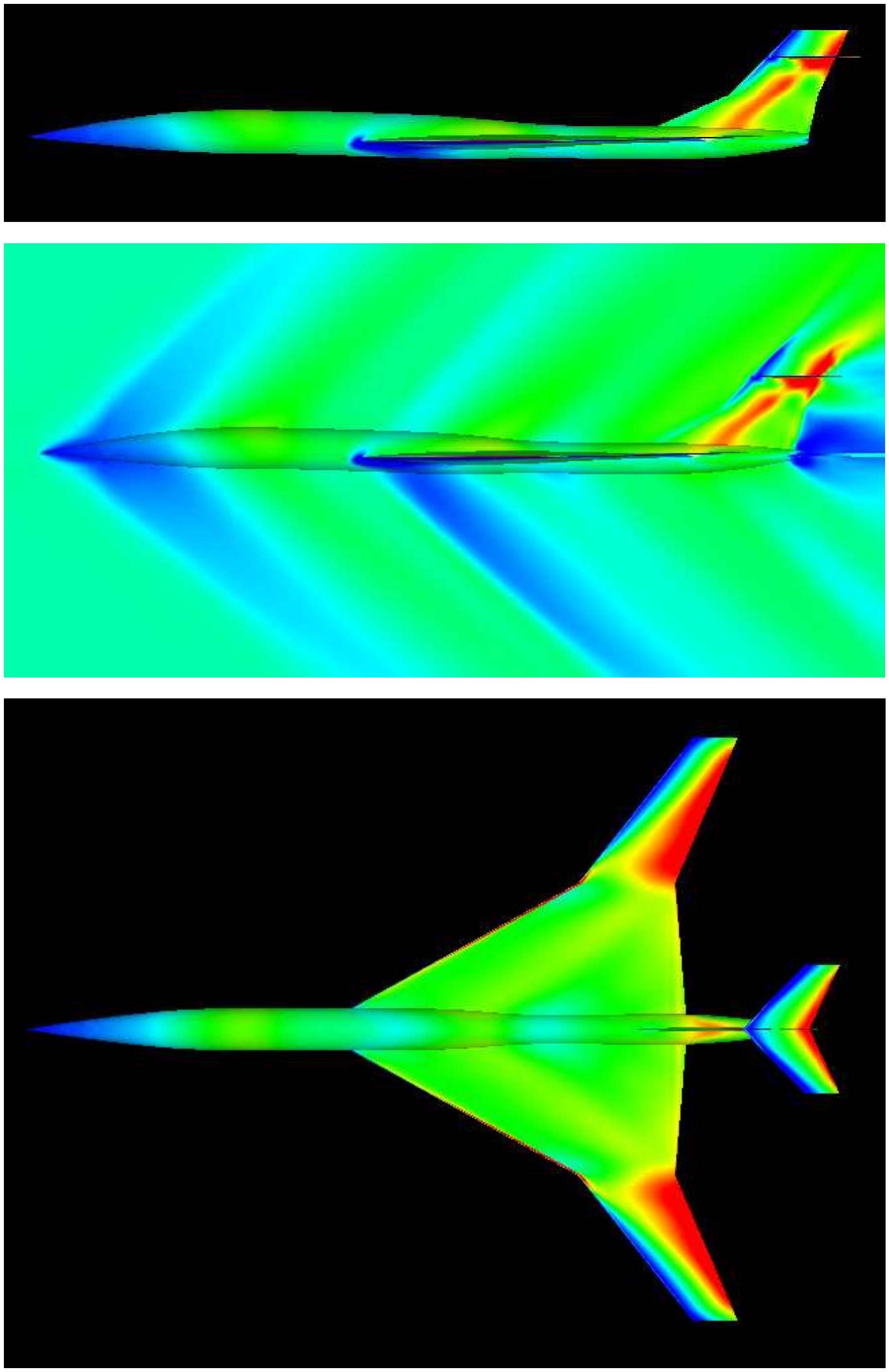

Fig. 8 Optimized aerodynamic configuration. $C_L = 0.1$, $C_D = 0.0080$, $M_\infty = 1.5$. **12 design variables using CAD-based geometry engine.**

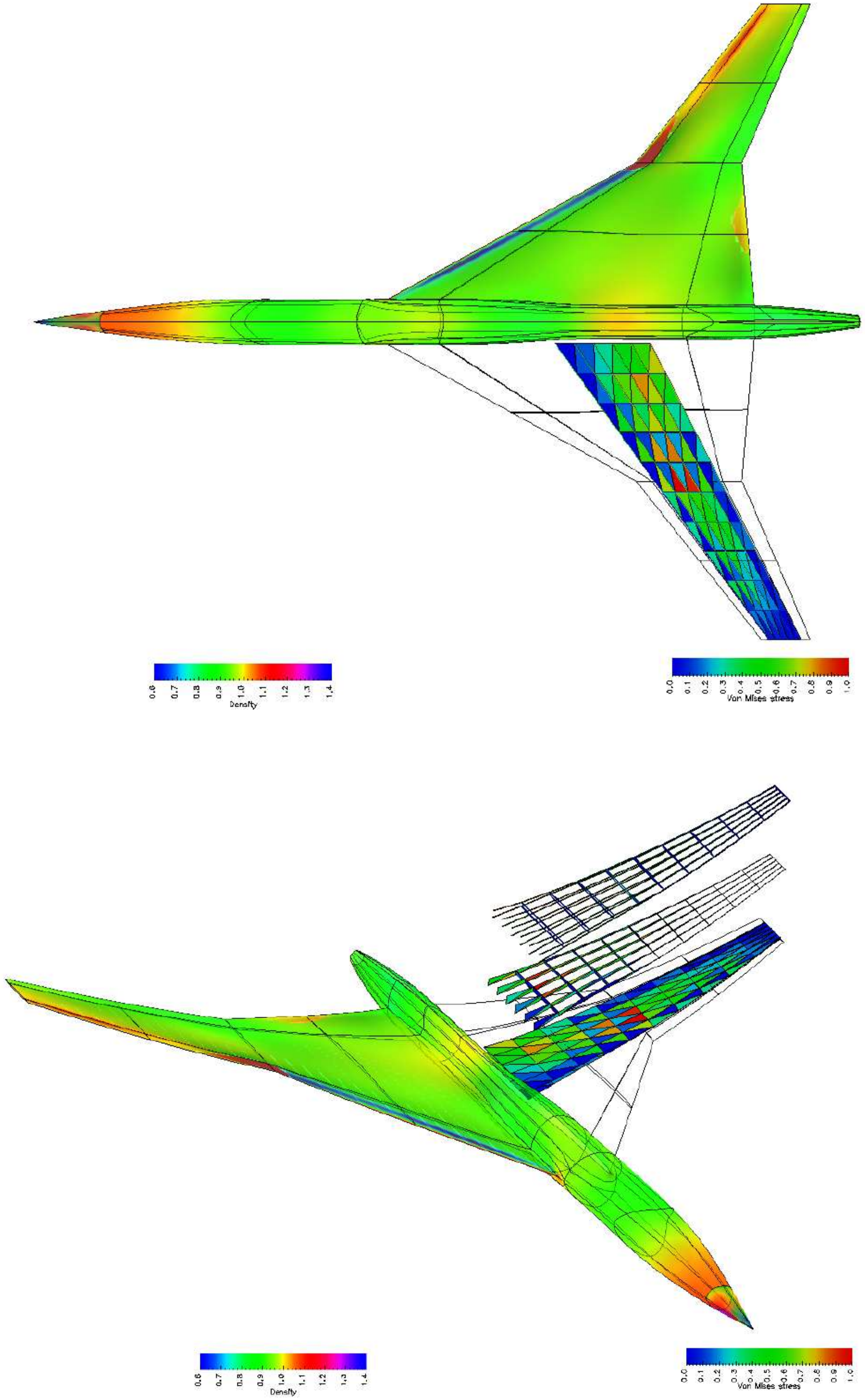

Fig. 9 Top and perspective view of aero-structural optimization setup for supersonic business jet.