Physically-Based, Real-Time Visualization and Constraint Analysis in Multidisciplinary Design Optimization

Yann Deremaux*, Karen Willcox†and Robert Haimes‡
Aerospace Computational Design Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

A framework and methodology are presented for effective, intuitive visualization of design optimization data. The visualization is effected on a CAD-based, physical representation of the system being designed. The Computational Analysis PRogramming Interface (CAPRI) is used to link a general optimization framework to a CAD model. An example is presented for multidisciplinary design optimization of an aircraft. The new methodology is used to visualize the path of the optimizer through the design space. This enables the designer to rapidly gain physical understanding of the design tradeoffs made by the optimizer. The visualization framework is also used to investigate constraint behavior. Active constraints are displayed on the CAD model and the participation of design variables in a given constraint is represented in a physically intuitive manner. This visualization approach serves to dramatically increase the amount of learning that can be gained from design optimization tools and also proves useful as a diagnostic tool for identifying formulation errors.

Introduction

Computational tools have become an invaluable part of the engineering design process. In particular, multidisciplinary design optimization (MDO) has become a popular approach for design of complex engineering systems. By simultaneously considering the disciplines of interest, one can "coherently exploit the synergism of mutually interacting phenomena".[1] Furthermore, by casting the design problem as a formal optimization statement, computational algorithms can be used to search the design space in a rational and efficient manner. MDO has enjoyed successful use in many different engineering applications, ranging from the design of aircraft, aircraft engines and spacecraft to automobiles.

*Research Assistant, deremaux@mit.edu, Member AIAA
†Assistant Professor, kwillcox@mit.edu, Member AIAA
‡Principal Research Engineer, haimes@mit.edu, Member AIAA

Recently, the field of MDO has had considerable impact by improving the performance, lowering the lifecycle cost and shortening product design time for complex systems.[1,2] For example, the Blended-Wing-Body aircraft design team uses an extensive MDO framework, which simultaneously considers aerodynamics, structures, weights, balance, stability and controls.[3,4] This framework leads to improved designs as well as much faster design turnaround time. As the problems considered become more complicated, vast amounts of additional data are produced by the optimizer. In a typical MDO run, hundreds or even thousands of different design options might be evaluated.

Although the complexity of design problems handled with MDO is becoming more impressive, the ability to effectively handle data has languished. MDO frameworks commonly lack flexibility; they are often developed for one particular application by a single individual. The interface is often unfriendly and many problem-specific attributes are hard-coded into the tool. In particular, the information generated during an optimization run is rarely communicated in an effective way (if at all) to the designer. An optimization run does result in a solution to the specified problem, but it also provides a wealth of information about the design space.

By looking at just the optimal solution generated, the designer uses MDO as a "push-button" tool, i.e. specify the problem and get the "best" answer. However, this is not the most effective use of such a design tool. Rather than being used to eke out a 5% improvement in the design solution (where model fidelity is often an issue anyway), MDO ought to be used as a way of gaining insight to the design space, quantitatively identifying trades and finding innovative design options. Often in practical applications, it is the solution concept suggested by the optimizer but not the actual details of the design that are most interesting. For example, in Wakayama,[4] through MDO it was determined that increasing the sweep of the wings could alleviate a balance problem in the Blended-Wing-Body aircraft. The details of the planform were subsequently

refined using high-fidelity computational fluid dynamic analysis. In the field of turbomachinery design, recent work in developing a new design interface builds on the concept that it is insight to and understanding of the problem at hand that achieves the biggest design successes.[5]

Winer, Samant et al.[6,7] propose a new paradigm: Visual Design Steering, based on Graph Morphing. This technique allows, after having ranked the constraints and the design variables of a problem, visualization of the position of the constraints in a subset of the design space. The output is a set of three-dimensional plots of design variables, constraint boundaries, and objective contours. The designer can then "steer" the optimization to obtain better results. Physical programming[8] is another approach used to visualize the optimization process in real time. Two-dimensional plots and color codes are used to identify the value of a solution, and monitor in real time the optimization process.

However, these visualization approaches have not focused on a representation of the optimization process that is linked to the physical design. In this paper we present an approach which uses a physical Computer Aided Design (CAD)-based model to display a more intuitive view of the optimizer design exploration. The Computational Analysis PRogramming Interface (CAPRI)[9] is a CAD vendor-neutral Application Programming Interface (API). This interface allows for establishment of a flexible framework that enables communication between a general optimization routine and a generic CAD system model.

The paper is organized as follows. The general optimization framework is first described, followed by a description of CAPRI. We then focus on two design visualization methodologies. The first is to visualize the path taken by the optimizer from the initial to the final solution, while the second is a detailed visualization of constraint behavior and its impact on the design. For each of these, the methodology will be described. Then, a simple General-Aviation aircraft design example will illustrate the two approaches. Finally, conclusions will be drawn.

Optimization Framework

Consider a general constrained problem as follows:

$$\begin{aligned} &\text{Minimize } F(x) \text{ for } x \in \Re^n \\ &\text{subject to:} \\ &\hat{g}_j(x) \leq 0,\ j = 1, ..., m_1 \\ &\hat{c}_k(x) = 0,\ k = 1, ..., m_2 \end{aligned} \tag{1}$$

where x is the vector of n design variables, $\hat{g}$ is the vector of m_1 inequality constraints, $\hat{c}$ is the vector of m_2 equality constraints and $F(x)$ is the objective function.

In this work, gradient-based optimization algorithms are considered, although the visualization methodology could also be used with heuristic techniques. In general, these algorithms are iterative. Beginning with some initial guess, the algorithm successively refines its current estimate of the design variables based on gradient information. In general terms, this can be written as

$$x^{k+1} = x^k + \alpha^k d^k \tag{2}$$

where x^k is the design vector at iteration k. The guess at iteration $k+1$ is computed by moving some scalar distance α^k in the direction d^k.

Given an initial solution x^0, different gradient-based algorithms will take different paths through the design space. In this work, sequential quadratic programming (SQP) is used. At each step, a subproblem with a quadratic objective function and linear constraints is created and solved. For this algorithm, all iterates are feasible (that is, they satisfy the constraints in (1)). Therefore, it may be of interest to the designer to visualize the path taken by the optimization and to thus gain insight to the physical design space.

Once the optimal solution, $x*$, has been reached, the designer may be interested in further interrogation of the optimal design. Sensitivity analysis yields information on the shape of the design space near the optimum and can lend valuable physical insight. Further insight can be gained by studying the set of active constraints and using sensitivity information to discern which aspects of the design are constrained by which requirements. This information can be obtained in two ways as follows.

The first approach to investigate constraint behavior is to compute the Lagrange multipliers at the optimal solution. The Lagrangian function $L(x, \lambda)$ can be written as

$$L(x, \lambda) \equiv F(x) + \sum_{j=1}^{m_1} \lambda_j \hat{g}_j(x) + \sum_{k=m_1+1}^{m_1+m_2} \lambda_k \hat{c}_k(x) \tag{3}$$

where λ_j is the j^{th} Lagrange multiplier. The Kuhn-Tucker conditions show that at the optimal solution $(x*, \lambda*)$, if a constraint $\hat{g}_j$ is inactive then the corresponding Lagrange multiplier, λ_j*, must be zero. That is, either $\hat{g}_j(x*) = 0$ or $\lambda_j* = 0$ for all $j = 1, 2, \ldots, m_1$. Moreover, the value of a non-zero Lagrange multiplier gives a linear indication of the rate of change in the optimal value of the objective function as the corresponding active constraint is relaxed.

The second method of investigating constraint behavior is to compute sensitivities of each constraint with respect to each of the design variables. Using this sensitivity analysis, one can determine the relative participation of each design variable in a specific constraint. The sensitivity of constraint g_j with respect to design variable x_i is given by the partial derivative $\frac{\partial g_j}{\partial x_i}$, where the vector g now contains both the inequality

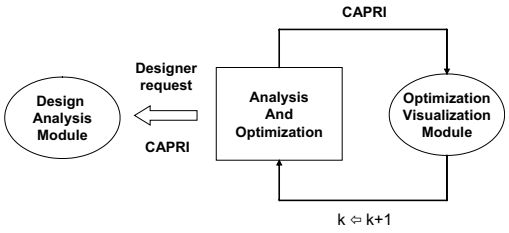

Fig. 1 Optimization/visualization framework

constraints $\hat{g}$ and the equality constraints $\hat{c}$. In order to compare sensitivity values in a meaningful way, it is necessary to compute a relative change:

$$\frac{\Delta g_j / g_{j_0}}{\Delta x_i / x_{i_0}} = \frac{x_{i_0}}{g_{j_0}} \frac{\partial g_j}{\partial x_i} \tag{4}$$

where x_{i_0} and g_{j_0} are normalization quantities for design variable x_i and constraint g_j respectively. The partial derivative in (4) is evaluated at the point of interest (often $x*$), however care must be taken when choosing the normalization quantities x_{i_0} and g_{j_0}. While the value at the optimum solution may be one choice, it is important that both x_{i_0} and g_{j_0} are chosen to be non-zero. While the choice for x_{i_0} is usually clear from physical considerations, the value of g_{j_0} is more difficult to select. For this reason, comparing sensitivity values across different constraints will not yield meaningful insight and in this work, we focus on comparing sensitivity values within each constraint.

Therefore, three mathematical entities have been identified that are of considerable interest to a designer: the sequence of iterates, $\{x^k\}$, the Lagrange multipliers at the optimal solution, λ_j*, and the sensitivities $\frac{\partial g_j}{\partial x_i}$. Two modules have been developed to visualize this information in a physical manner. Figure 1 shows the optimization/visualization framework. While the optimization takes place, the Optimization Visualization Module (OVM) allows for the real-time visualization of the optimization process, i.e. the sequence of iterates during the optimization. The Design Analysis Module (DAM) permits the designer to interrogate a particular solution. DAM displays the active constraints, shows the physical impact of the constraints on the design, and demonstrates how the constraints drive the design variables and the objective function. All the communications between the different modules and the CAD model use CAPRI. CAPRI will be further described in the next section.

These visualization modules satisfy a set of requirements. First, the modules should be flexible. They must couple easily with different geometric representations of the physical system and with different optimization frameworks. They must also be able to handle a broad range of problems. Second, the visualization modules should not substantially increase the computational time of the overall optimization run.

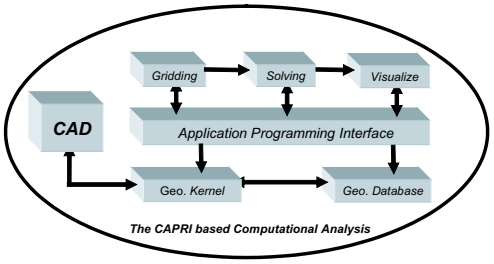

Fig. 2 The CAPRI based Computational Analysis Suite

CAPRI

CAPRI[9] is a CAD vendor-neutral API. This middleware provides appropriate programming access for analysis suites that require direct access to the CAD model and can be used when the desired analysis does not have a direct CAD connection. The CAPRI programmer and/or user need not be a CAD operator.

Figure 2 shows that by allowing access to the geometry from within all the analysis sub-modules (grid generators, solvers and post-processors) such tasks as meshing, solver-based node adaptation and general geometry queries become simpler and consistent. The connection to the geometry is made through an API, which isolates the analysis suite from the geometry kernel, which avoids any loss of solid geometry information in a translation.

The Geometry Viewer of CAPRI will also be used. It is not an integral part of the API, but can be thought of as another module of the software suite. It can be used as the visual front-end for CAPRI.

For the work described in this paper, all CAD access was performed through CAPRI after a CAD part was constructed in a "parametric" sense (called the Master-Model). Pro/Engineer was used as the backend, but any supported CAD system could have been applied to the work presented here. In this case, CAPRI accesses the CAD data in a native manner through Pro/Toolkit (Pro/Engineer's internal API).

CAPRI's API avoids the complete Computational Geometry (CG) perspective while maintaining full functionality. This simplification of the data definition and API provides ease of software generation, without regard for special cases. The important functions found in CAPRI that were used are summarized below:

(1) Support Manifold Solids. By only supporting solid geometry, problems in trimming surfaces do not exist. If handled properly, the geometry need not be fixed or modified.

(2) Direct CAD Access. The data exposed through CAPRI exists in the CAD system. There is no geometry translation, thus avoiding the errors and

other problems associated with CAD model translation.

(3) Dual View of the Geometry. Both the CG and a discrete view of the solid are available through CAPRI. A complete, closed, conformal tessellation[10] of the geometry found in the CAD system is exposed on a surface-by-surface basis. This provides a proper foundation for the type of functions that are required for visualization task at hand.

(4) Master-Model Manipulation. By allowing the specification of both the parameter values (that define the geometry construction) and suppression of nodes of the "feature-tree", different instances of the part (or assembly) can be constructed. This portion of CAPRI allows for both geometric parameter studies as well as full design optimization.

(5) Shape Modification. It has been found that the parametric Master-Model view is inadequate for shape optimization (shapes tend to be too complex to drive in this manner). CAPRI allows for the direct manipulation of specific curves that are the basis for the operations of blending, lofting, extruding and rotating. For example, an application then has the ability to change the shape of a wing by adjusting the curves that define the airfoil shapes at various sections.

Real-Time Visualization

As discussed earlier, one of the issues facing MDO is the disconnection between the designer and the physical representation of the system being designed. An optimizer will take full advantage of gaps in the formulation. For instance, if a constraint is omitted or cast incorrectly, an optimizer will often return a physically unreasonable design or will fail to converge. However, when the optimizer output is communicated to the designer by a list of numbers, it is difficult to quickly observe and diagnose this behavior. Offering a more physical representation to the designer by visualizing optimization data on a CAD model will enable the following:

(1) Have a physical representation of the system being optimized.

(2) Improved understanding of the tradeoffs that are made in the optimization design process.

(3) Assist in the early stages of optimization problem formulation and implementation.

(4) Easily determine the path taken by the optimizer through the design space and interrogate individual design options.

(5) Verify assumptions regarding the physical configuration of the system. In many cases, the designer may not be aware that these assumptions were made.

(6) Simplify the downstream management of the assembly and manufacturing tasks by identifying problems upfront. For example, the optimizer could decrease the thickness of a part to a unreasonable value that would prevent the part from being manufactured. Monitoring the design evolution would help to identify this more quickly.

Approach

The methodology developed to visualize the progress of an optimization routine can be summarized by the following four steps: creation of a CAD model, parameterization of the CAD model, validation of the CAD model, and linking of the CAD model to the optimization framework. Each of these steps will be described in detail below.

CAD model description

The first step in developing the real-time visualization module is to create a robust CAD model. The CAD model should reflect the design approach and encapsulate the trends of the final design. It should capture the aspects of the physical system that are most likely to evolve throughout the optimization and/or should be communicated to the designer. At the least, it should include the key parameters of the design. MDO is often used for conceptual or preliminary design, hence a model that roughly represents the proportions of the object will, in general, lend sufficient insight to the designer. However, a higher fidelity CAD model may be beneficial for other issues, such as assembly and manufacturing concerns. Finally, it should be noted that it is important to recognize the limitations of a coarse CAD model and use caution when utilizing information, such as a mesh, as input to a higher fidelity model.

CAD model parameterization

Parameterization of a CAD model requires insight and input from the designer. The parameterization should be carried out in the context of linking the CAD model to an optimization framework, however, it is not necessary that all design variables map to a different parameter in the CAD model. Instead, simple relations can be defined, which allow the design variables to map to a reduced set of CAD model parameters. Another issue is that the parameters used in the CAD model may not be geometrically linked with any particular design variable. Especially for complicated representations, there may be a number of CAD model parameters which are hidden from the optimization, that is, they are set by relations within the CAD model and cannot be accessed or modified from the

optimization framework. Even though this approach may simplify the CAD model, it is advised to use extra parameters in order to suppress such hidden definitions and create a more robust model.

Different parameterization approaches can be conceived, which lead to the same geometric representation. For example, consider a simple tapered wing. Two parameterization approaches can be considered, as shown in Figure 3. The first approach defines the wing using the following nine parameters: wing area, wing span, taper ratio, sweep angle, dihedral angle, thickness distribution and the X, Y, Z position of a point of the wing. This could also be complemented by the choice of specific airfoils.

The second approach puts the emphasis on the airfoils rather than on the wing itself. One is located at the root, the other at the tip, but both are defined and located by their own set of parameters. In this case the ten parameters are the spatial location (X, Y, Z) of one point at the leading edge of each airfoil, the chord length and the thickness ratio, both defined at the root and at the tip.

The two approaches give a same result, as shown in Figure 3, but largely differ by the fact they involve different sets of design variables, and allow different level of flexibility. For example, the second approach permits the addition of more airfoils between the root and the tip. However, the second approach is less physical than the first one, that may result in a loss of understanding by the designer.

CAD model validation

Once the CAD model has been created, it needs to be tested for different values of the parameters. One should make sure that a particular value of a parameter does not request operations where the CAD system cannot properly generate geometry. The range of variation of the parameters should at least include the projected feasible design space. This is important to ensure that the coupled optimization-visualization framework is robust.

Optimization-visualization link

As Figure 4 shows, the main idea is to have two sequences of separate actions running in parallel (a double-threaded approach) that can communicate with each other and share resources. The link between the two sequences, the optimization module and the design (or analysis) module, is carried out via CAPRI. CAPRI is also the interface between the CAD model and its definition, the model analysis tools, and the visualization module. It allows for information to be shared between the three components, as shown in Figure 5. On the right of the figure, the design module executes the optimization loop and runs the analysis routines. On the left, the visualization module dynamically displays the evolution of the system as the optimization proceeds.

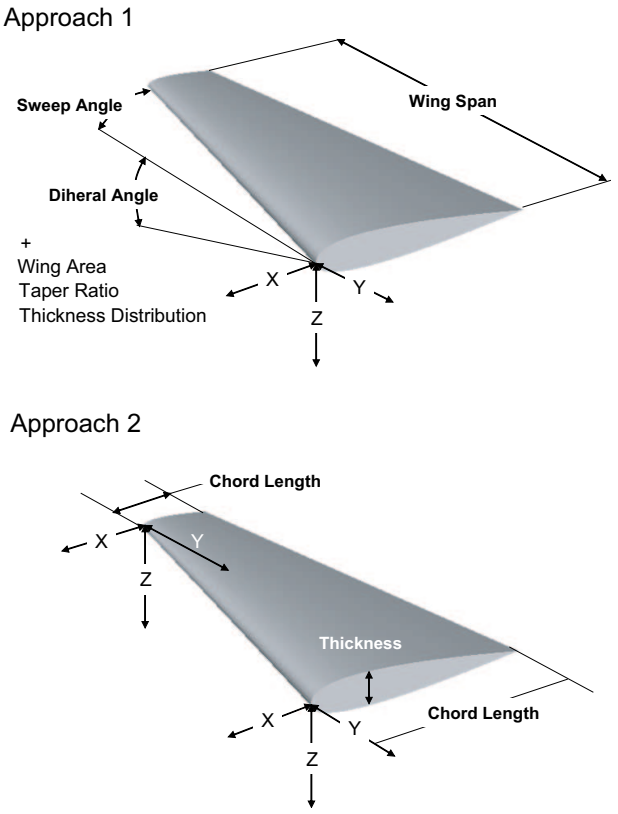

Fig. 3 Two different approaches for the parameterization of a tapered wing

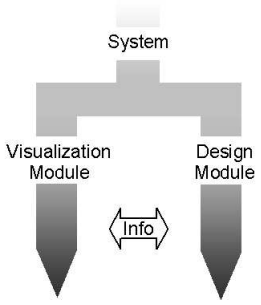

Fig. 4 Double-threaded approach. First thread carries analysis and optimization module. Second thread carries dynamic visualization module

Rate of learning versus complexity

Computation time, even with the dramatic increase in computer power, remains an important issue. Applications of MDO are tending towards consideration of more complex systems, use of higher fidelity tools and use of more complex optimization techniques. Venkatara and Haftka identified three types of complexity:[11]

(1) Model complexity, inherent to the size of the problem, and related to the number of design variables and constraints.

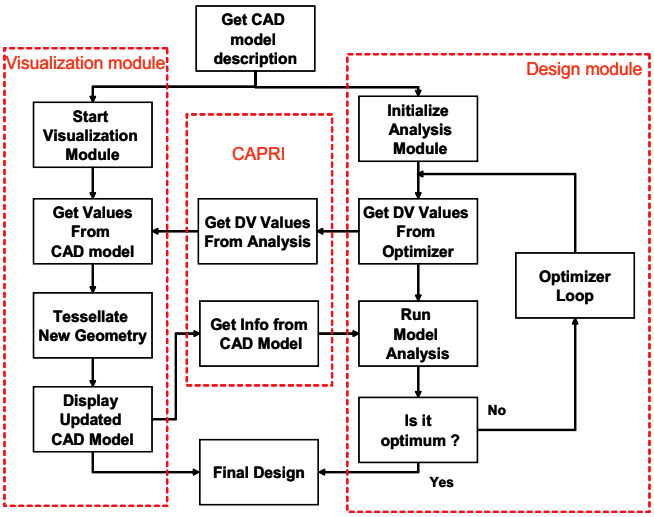

Fig. 5 Approach of the visualization problem

(2) Analysis complexity, related to the level of fidelity of the models used. Fidelity ranges from low-fidelity, empirical models to medium-fidelity models based on a simplified approach, such as beam structural models or panel methods for aerodynamics. High-fidelity models, such as computational fluid dynamic (CFD) and finite element structural analysis, are typically used in MDO for post-analysis and limited optimization.[12]

(3) Optimization complexity, related to the type of optimization: linear or nonlinear, deterministic or probabilistic.

However, the visualization problem does not depend on any of these three types of complexity. Rather, it is completely dependent on the complexity of the CAD model. The tessellation that occurs at each step of the optimization for visualization requires computational resources proportional to the complexity of the shape to be created.

This leads to a trade-off between the need for information and the ratio between the time needed for the combination of analysis and optimization and the time required for the visualization while accounting for the time available.

Rubbert[13] evaluates the quality of a design solution by the amount that can be learnt over a certain period of time.

$$(Rate\ of\ Learning) = \left(\frac{Learning}{Cycle}\right) * \left(\frac{Cycle}{Time}\right) \quad (5)$$

The first term, $\left(\frac{Learning}{Cycle}\right)$, represents the amount of information that is gathered during a design cycle. The second term, $\left(\frac{Cycle}{Time}\right)$, represents the number of cycles that can be effectively carried out in a given amount of time. The product of these two terms can give the designer an idea of the amount of information

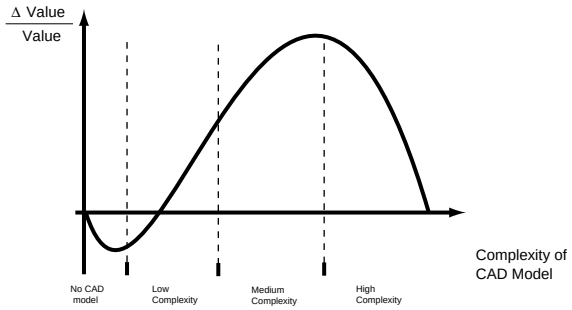

Fig. 6 Effect of the complexity of the CAD model on the value of visualization

that can be obtained over a given amount of time. The higher the rate of learning, the more valuable the information for the designer. However, in order to evaluate the value of bringing visualization in the design process, we need to think in terms of incremental change brought by the visualization. Rubbert[13] uses the following terms: $\left(\Delta\frac{Learning}{Cycle}\right)$, which represents the new information that the visualization will bring to the designer and $\left(\Delta\frac{Cycle}{Time}\right)$, which represents the additional number of cycles that can be executed in a given amount of time. In the case of the visualization, this term is negative since bringing visualization into the design will increase the amount of time necessary per cycle. Ignoring higher order terms, the ratio of the incremental value of visualization to the value of previous information can be written as:

$$\frac{\Delta Value_{Viz}}{Value} = \left(\frac{\Delta\frac{Learning}{Cycle}}{\frac{Learning}{Cycle}}\right) + \left(\frac{\Delta\frac{Cycle}{Time}}{\frac{Cycle}{Time}}\right) \quad (6)$$

Effect of complexity of CAD model on the learning rate

Using this approach, we can analyze the effect of complexity of the CAD model on the rate of learning. Complexity of the CAD model has been decomposed into three levels:
(1) Low complexity: the CAD model encapsulates only the general ideas of the design at the simplest level.
(2) Medium complexity: the CAD model encapsulates the details of the design that are necessary for physical understanding. However, features such as fillets and corners are not considered unless they bring added value to the designer.
(3) High complexity: the CAD model is an exact representation of the physical system, including details such as fillets. This CAD model could be used directly for CFD analysis or in the manufacturing process.

Figure 6 demonstrates the value of having a CAD model that balances accuracy and computation time. Such a model is accurate enough to represent reality, but does not include details superfluous to the designer's needs. Also, note the possibility of introducing

a burden when the CAD model does not encapsulate enough detail to bring sufficiently valuable information to the designer. In that case, the information added by a simple CAD model cannot compensate the loss of time due to the visualization module, and the overall value increment is negative.

Constraint Analysis and Visualization

Obtaining a clear picture of the design space is a difficult issue to address. Displaying optimization progress through the design space is one approach that has been discussed. Further insight may be gained by a more detailed investigation of constraint behavior. In particular, the designer may be interested in discerning which constraints are active, which constraints affect which parts of the system, and how much certain constraints drive the design. Effective visualization of this information is not an easy task. The methodology must be developed in such a way that it is flexible, applicable to general problems and automated. Winer and Bloebaum[6] use Graph Morphing to display the constraints on a 3D representation of a subset of the design space. Here, a novel, more intuitive approach is used, which visually ties the constraints to physical features of the system. Such an approach was performed manually for a Blended-Wing-Body optimization result, and was found to lend valuable insight to the designer.[3] The Design Analysis Module developed here allows physical visualization of the constraints to be performed automatically in real time, as described in the following section.

Constraint sensitivity analysis

The first step of the approach is to determine which constraints are active at a given design solution. This can be easily achieved by finding the constraints with non-zero Lagrange multipliers. The relative participation of the design variables in each constraint is then determined using (4). This sensitivity analysis results in a matrix containing the normalized sensitivities of each constraint with respect to each of the design variables, where a row corresponds to a particular constraint and a column corresponds to a particular design variable. It is important to note that the normalization allows a good comparison of the effect of each design variable within a given constraint (going across a row of the matrix). However, there is no single way to scale each constraint, thus preventing the effective comparison of the relative importance of each constraint (going down a column of the matrix).

Once the importance of each design variable within a specific constraint has been calculated, it is possible to rank the design variables by comparing the sensitivities. This ranking will give the designer insight as to which design variables have an effect on a constraint, and their relative importance. Figure 7 depicts how such information might be displayed using a bar graph. In this case, the normalized sensitivities of a particular constraint with respect to five design variables are plotted. As the figure shows, variables x_2, x_1 and x_5 are particularly important for this constraint. While the plot shown in Figure 7 can provide useful information to the designer, it does not instantaneously lend physical insight. Moreover, if the number of design variables and constraints is large, studying a large number of such graphs with many bars might be time-consuming and ineffective. This approach should therefore be complemented with a more intuitive approach, which displays the information on a physical model of the system.

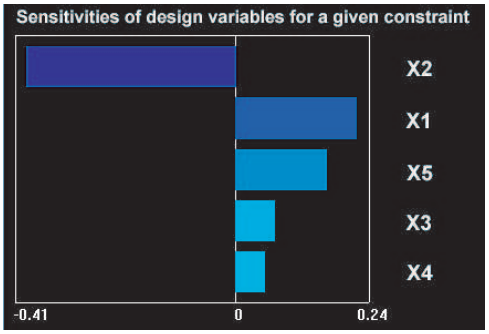

Fig. 7 Effects of five design variables on a given constraint

Constraint Visualization

Types of constraints

Different types of constraint can be identified. The type of constraint will directly affect the way it needs to be displayed.

The first type of constraint that can be encountered is a constraint that is directly linked to a geometric feature of the model. For example, if the designer considers a panel approach for the model, certain constraints can be directly related to a specific panel.

The most common type of constraint is one that is not directly linked to any geometric parameter. In this case, the sensitivity analysis discussed earlier allows the identification of the design variables that drive the constraint towards its bounds. The constraint is therefore linked to the physical model through the design variables.

A multi-level approach

A multi-level approach will be used in order to display the constraints. Each constraint will be linked to a set of design variables. Each design variable will contain three levels of information. The first level of information refers to the feature with which the design variable can be associated. The second level is the type of the design variable. Different types can be distinguished, including length, area, diameter, angle, ratio, and user defined. Finally, the third level represents the real meaning (for the designer) of the design

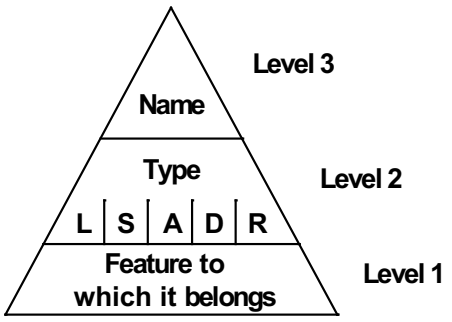

Fig. 8 Multi-level approach for constraint visualization

variable, defined by its name. These three levels of information are depicted in Figure 8.

Each level is represented by a graphical display. The first level of information is depicted by highlighting the specific feature on the CAD model representing the object to be optimized. The second level and third level are accessed by clicking on a given constraint, and then by selecting the design variable for which more information is needed. Doing so, the designer can choose, in order to perfectly identify all the design variables involved, to list the important design variables.

The entire process can be summarized as follows:

(1) Identify the active constraints using a Lagrange multiplier approach.

(2) Identify the design variables that can be associated to a specific active constraint, rank them, and determine the significant design variables that drive the design for a given constraint.

(3) Make the distinction between a design variable that is related to a physical element and a design variable that is not linked to the geometry (for example, the cruise velocity in the case of an aircraft design).

(4) Highlight the geometric features corresponding to the identified design variables.

(5) At the designer's discretion, display the second level of information in a separate window and reveal the type of design variables. Display non-geometric variables in another window.

(6) Display the names of the constraints and/or design variables.

This visual information is also complemented by a list that transmits the status of any constraint, as well as the design variables related to it, at the chosen solution point.

Visualization and constraint analysis user interface

Based on the ideas detailed earlier on real-time visualization and constraint analysis, a user interface has been built around the framework and can be used during the design of a system. This interface is shown in Figure 9. To fulfill the dual functions of real-time visualization and constraint analysis, the user interface is composed of four main panels. In the lower left corner of Figure 9 is the log window. This panel is the primary interface between the user and the optimizer, and displays basic information, such as error messages and optimizer text output. In the upper left corner is the 3D visualization module window, in which the 3D system model is displayed. The constraint analysis module window in the lower right corner allows the designer to navigate through the constraints. This window encapsulates the three levels of information described in Figure 8 and will be discussed in more detail in the example below. Finally, a sensitivity analysis module window in the upper right corner displays constraint sensitivity information. The operation of each of these panels is demonstrated in more detail using the following simple aircraft design example.

Table 1 Summary of the design variables, constraints and objective of the General Aviation aircraft design problem.

Design variables	Wing span
	Wing area
	Fuselage diameter
	Fuselage length
	Cruise velocity
Constraints	Stall velocity
	Rate of climb
	Wing bending stress
	Fuselage diameter
Objective function	Max. Breguet range

Real-time visualization example

A simple General Aviation (GA) aircraft model will be considered here. The optimization framework is based on low-fidelity analysis models as follows. The aerodynamics are modelled empirically using DATCOM data,[14] the structural analysis is based on a simple beam model and the weights model is based on empirical relations.[15–17] Although simple, the model however has the right trends to effectively conduct an optimization. Table 1 summarizes the design variables, constraints and objective function used.

The CFSQP optimization algorithm was used.[18] The initial design point was feasible and was based on Cessna-172 data.[19] Both the initial design point and the resulting optimal solution are shown in Table 2.

Optimization path monitoring

A CAD representation of the aircraft was developed in Pro/Engineer and linked to the optimization framework using CAPRI. At each iteration, the cur-

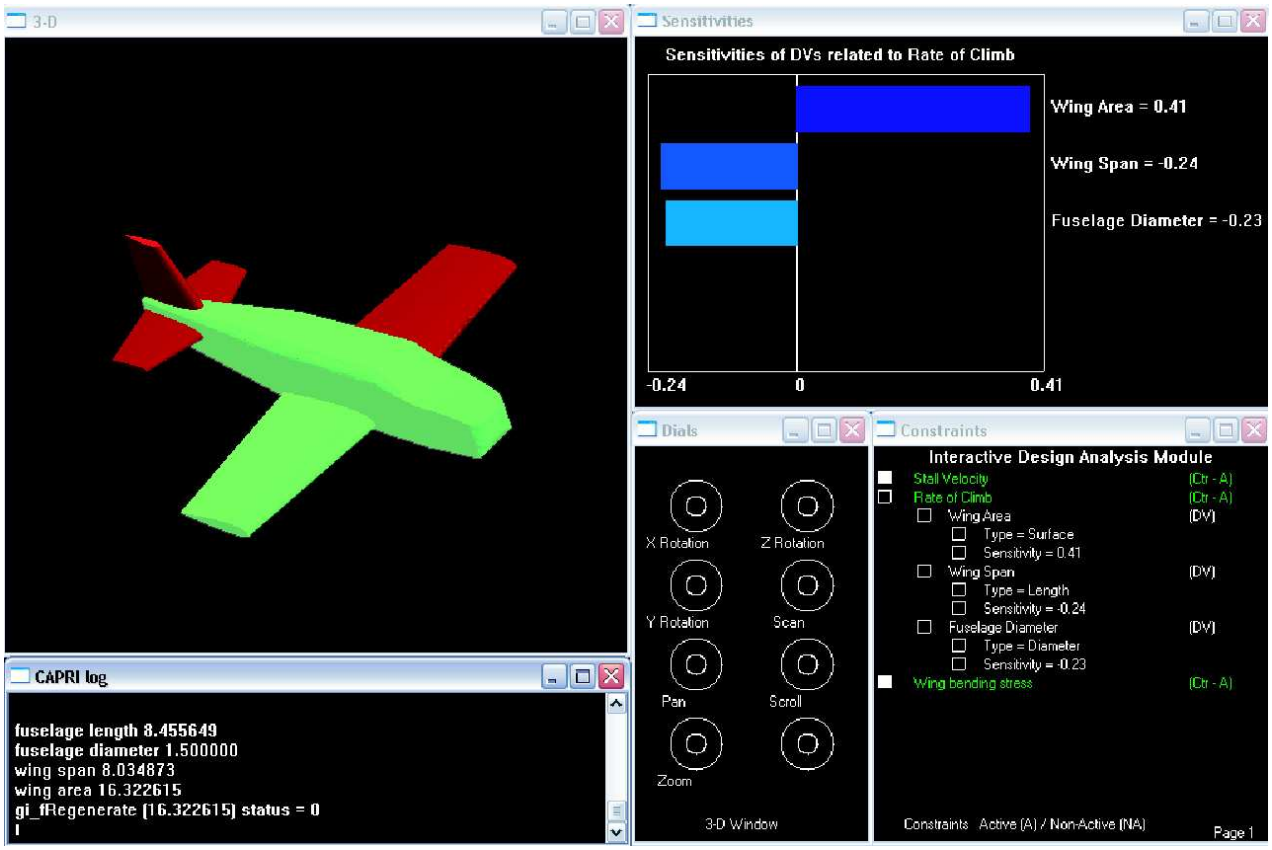

Fig. 9 User interface for the real-time visualization and physical display of constraint analysis during the optimization process

Table 2 Summary of the results obtained for the initial point and the optimal solution for the General Aviation aircraft design problem.

Type	Name	Init	Opt	Units
Dvs	Wing span	7.5	9.4	m
	Wing area	14.3	21.2	m^2
	Fuselage diameter	1.5	1.5	m
	Fuselage length	8.2	6.6	m
	Cruise velocity	40.1	42.9	$m.s^{-1}$
Cntrs	Stall velocity	26.9	23.8	$m.s^{-1}$
	Rate of climb	2.15	2	$m.s^{-1}$
	Wing bending stress	1482	1500	MPa
	Fuselage diameter	1.5	1.5	m
Obj	Max Breguet range	373	775	km

rent design was viewed in real-time using the CAD model. Figure 10 shows snapshots of the design at the initial and optimum solutions. From superimposing the two designs, it can be seen clearly how the optimizer chose to change the design: the fuselage length has decreased, while the wing span and wing area have increased. These physical changes are instantly evident to the designer using the visualization. They can also be confirmed by considering the Breguet range equation:

$$R = \frac{V_{Cruise}.L/D.\eta}{g.SFC} log(\frac{W_{Total}}{W_{Total} - W_{FuelCruise}}) \quad (7)$$

where V_{cruise} is the cruise velocity, L/D the lift to drag ratio, η the propulsive efficiency, SFC the specific fuel consumption, W_{Total} the total gross take-off weight and $W_{FuelCruise}$ the weight of fuel burnt during the cruise.

An increase in wing span improves the L/D of the aircraft, and thus the range. An increase in wing area allows more fuel to be carried. A decrease in fuselage length results in a decrease in empty weight. Similarly, the fuselage diameter does not increase from its lower bound due to both weight and drag considerations. From Table 2 it can also be seen that the cruise velocity was increased; however, this is a design variable that is not visualized on the CAD model. While the optimization progress presents valuable information to the designer, in order to fully understand the tradeoffs, one must also consider the impact of the constraints as discussed next.

Constraint analysis

In order to illustrate the visual constraint analysis in the case of the GA aircraft design, the constraint

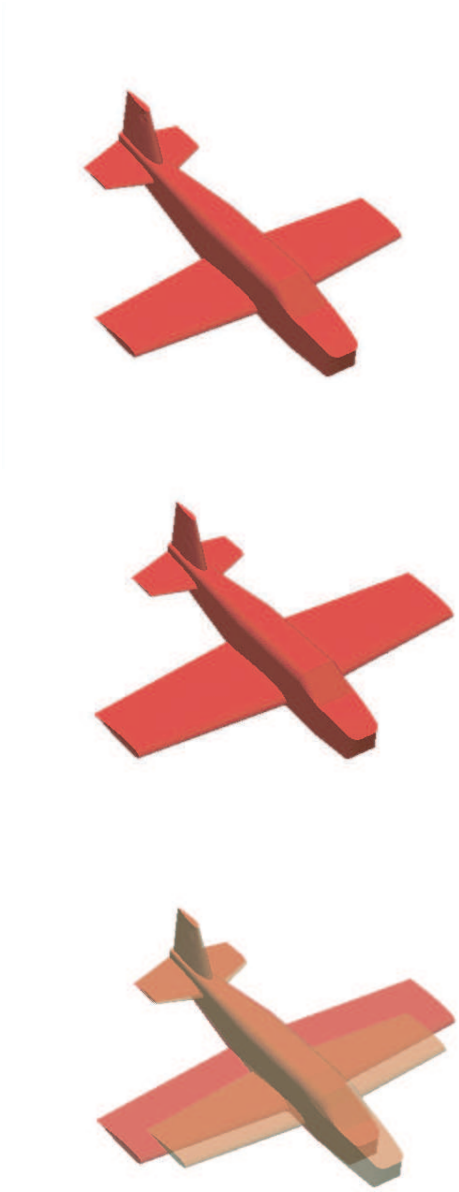

Fig. 10 Snapshots of the design taken for the CAD model. Top: the initial design solution; middle: the final design solution; bottom: the initial and final design solutions are superimposed. The bottom plot clearly shows the design tradeoffs chosen by the optimizer

limiting the rate of climb was analyzed and is shown in Figure 9. This constraint, even though it is physical, is not linked to a specific geometric feature. As a result, sensitivity analysis gives the designer valuable information on the design variables that participate strongly in the constraint.

The user interrogates the constraint using the tree in the constraint analysis module window (lower right of Figure 9). When the rate of climb constraint is selected, the sensitivity information is displayed. As shown by the bar graph in the figure, the sensitivity analysis suggests that the constraint is serving to limit the wing span and fuselage diameter and increase the wing area. At the same time, the physical display of the constraint is activated in the 3D visualization window. The appropriate features identified by the sensitivity analysis are highlighted, as in Figure 9, where the fuselage and wing of the GA aircraft are highlighted. This novel approach to physical constraint visualization, combined with the classic sensitivity bar chart, allows a designer to quickly gain insight to the design space and, if necessary, refine the optimization formulation. The full value of physical constraint visualization would be realized on a more complex example with many constraints and design variables. A similar analysis on other active constraints shows that, as expected, a further increase in wing span is limited by bending stress restrictions.

Optimization health monitoring

While the previous example showed the value that can be gained from physically visualizing optimizer progress, another use of this methodology might be to monitor the health of an optimization run. This allows the designer early detection of errors in the formulation that lead the optimizer toward a physically unreasonable design solution, and may be an invaluable tool in the initial formulation of a complex MDO problem.

Two examples are shown here for the GA aircraft example. Figure 11 represents a design iterate of an optimization for which the constraint on the fuselage diameter was forgotten. As a result, the weight and aerodynamic analyses try to drive the value of the fuselage diameter to its minimum, in this case zero. Figure 12 shows the case of a bad constraint on bending stress. In this case, the design is driven towards an infinite wing span in order to improve the lift-to-drag ratio and thus the range. In each case the optimizer would find, after several iterations, an optimum that is not physically reasonable. For a more complicated system, these wasted iterations could be costly and time-consuming. Visualization therefore allows the designer to detect a flaw in the analysis model, and stop the optimization early.

Fig. 11 The constraint on the fuselage diameter was forgotten. The optimization drives the design towards a fuselage diameter of zero. The CAD model would not regenerate at the next iteration.

Fig. 12 The constraint on the wing bending stress was forgotten. The optimization drives the design towards an infinite wing span.

Conclusion

A framework has been established for visualization of information generated by design optimization algorithms. This framework uses the CAPRI software to create a link between a general optimizer and a generic CAD-based representation of the geometry. This link is created in such a way as to maintain a high level of flexibility. Methodology has been developed to visualize the progress of the optimizer in real time and to effectively investigate the impact of constraints on a given design solution.

A simple problem that involves the design of a GA aircraft was considered here. Monitoring of the optimizer sequence of iterates was shown to lend physical insight to design trade-offs, without a time-consuming study of data files or numerical output. This real-time visualization was also shown to be useful for monitoring the health of an optimization run and for early detection of formulation errors. Visualization of constraint behavior was also demonstrated and shown to provide physical insight to the designer.

References

[1]Giesing, J. P. and Berthelemy, J-F, M., "A Summary of Industry MDO Applications and Needs," AIAA Paper 98-4737, June 1998.

[2]Zang, T. A. and Green, L. L., "Multidisciplinary Design Optimization techniques: Implications and Opportunities for Fluid Dynamics Research," AIAA paper 99-3798, presented at the 30^{th} AIAA Fluid Dynamics Conference, June 1999.

[3]Wakayama, S., "Blended-Wing-Body Problem Optimization Setup," AIAA Paper 2000-4740, presented at the 8^{th} $AIAA/USAF/NASA/ISSMO$ Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 2000.

[4]Wakayama, S. and Kroo, I., "The Challenge and Promise of Blended-Wing-Body Optimization," AIAA Paper 98-4736, presented at the 7^{th} $AIAA/USAF/NASA/ISSMO$ Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 1998.

[5]Jarret, J. P., Dawes, W. N., and Clarkson, P. J., "Accelerating Turbomachinery Design," GT-2002-30618, Proceedings of the ASME Turbo Expo 2002, The Netherlands, June 2002.

[6]Winer, E. and Bloebaum, C., "Visual Design Steering for Optimization Solution Improvement," AIAA-2000-4815, presented at the 8^{th} $AIAA/USAF/NASA/ISSMO$ Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 2000.

[7]Samant, A., Shah, P., and Winer, E., "Visual Design Steering to aid decision-making in optimal design," AIAA-2002-4815, presented at the 9^{th} $AIAA/USAF/NASA/ISSMO$ Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, September 2002.

[8]Messac, A. and Chen, X., "Visualizing the Optimization Process in Real-Time Using Physical Programming," AIAA-98-39707, 1998.

[9]Haimes, R. and Follen, G., "Computational Analysis PRogramming Interface," Proceedings of the 6^{th} International Conference on Numerical grid generation in Computational Field Simulations. University of Greenwhich, September 1998.

[10]Haimes, R. and Aftosmis, M, G., "On Generating High Quality 'Water-tight' triangulations Directly from CAD," Proceedings of the 8^{th} International Conference on Numerical grid generation in Computational Field Simulations. Honolulu, Hawaii, June 2002.

[11]Venkataraman, S. and Haftka, R. T., "Structural Optimization: What Has Moore's Law Done for Us ?" AIAA-2002-1342, Presented at the 43^{rd} AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference in Denver, CO., April 2002.

[12]Bartholemew, P., "The Role of MDO within Aerospace Design and Progress Towards an MDO Capability," AIAA-98-4705, 1998.

[13]Rubbert, P., "On The Pursuit of Value for CFD," in *Frontiers of Computational Fluid Dynamics*, ed. Caughey, D.A and Hafez, M.M., World Scientific Publishing, 1999.

[14]Finck, R. D. and Hoak, D. E., "USAF Stability and Control DATCOM," USAF Contract F33615-76-C-3061, April 1978.

[15]Raymer, D. P., *Aircraft Design: A Conceptual Approach*, chap. 15, AIAA Education Series, 3rd ed., 1999.

[16]Ojha, S. K., *Flight Performance of Aircraft*, chap. 12, AIAA Education Series, 1995.

[17]Roskam, J., *Airplane Aerodynamics and Performance*, DARCoproration, Lawrence, Kansas, 1988.

[18]Lawrence, C., Zhou, J. L., and Tits, A. L., "User's guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, generating Iterates Satisfaying All Inequality Constraints," 1994.

[19]Jackson, P., *Janes All the World Aircraft*, Janes Information Group Ltd, Surrey, U.K., 1997-98.