AIRCRAFT OPTIMIZATION BY A SYSTEM APPROACH: ACHIEVEMENTS AND TRENDS

ICAS-92-1.3.4

Jaroslaw Sobieszczanski-Sobieski
NASA Langley Research Center
Hampton, Virginia, U.S.A.

Abstract

Recently emerging methodology for optimal design of aircraft treated as a system of interacting physical phenomena and parts is examined. The methodology is found to coalesce into methods for hierarchic, non-hierarchic, and hybrid systems, all dependent on sensitivity analysis. A separate category of methods has also evolved independent of sensitivity analysis, hence suitable for discrete problems. References and numerical applications are cited. Massively parallel computer processing is seen as enabling technology for practical implementation of the methodology.

Introduction

By virtue of the physics involved, an aircraft is a system whose behavior is a resultant of complex interactions among many different physical phenomena and hardware components. Traditionally, designers created vehicles exhibiting the desired behavior by relying on judgment and intuition, combined with experience and statistics, in manipulating design variables, and they resorted to analysis for guidance and verification. Plots limited to 3 or 4 dimensions were the favorite means for visualization of the quantitative information. In the last two decades, availability of digital computers increased the role of analysis as a guide to the design decisions and led to the steadily increasing use of formal optimization methods as tools for determining the values of design variables.

The early attempts of simply connecting a design space search program with a set of analysis programs proved inadequate but they inspired development of a number of alternatives that currently have crystallized in a few major, distinctly different approaches: 1) decomposition of the problem into smaller subproblems coupled in a hierarchic, non-hierarchic, or hybrid manner; 2) generating a population of trial configurations and subjecting it to a selection process according to the Darwinian rules of the "survival of the fittest"; and 3) representing a number of trial configurations strategically placed in the design space by a hypersurface to be numerically searched for optimum. All these approaches tend to strain the present computer technology to the limit. However, the recent trend in that technology toward massively parallel processing is coming just in time to provide means for their cost-effective implementation.

The purpose of the paper is to review the essential features of the above approaches to the problem of optimal design of the aircraft optimization, leaving details to a sample of references cited without attempting a comprehensive literature survey. The review emphasizes the methods pursued at the author's organization at the NASA Langley Research Center, including the disciplinary and system sensitivity analyses that are the foundations of the first of the above three approaches. The review is illustrated by numerical examples selected from the application experience accumulated to date. It is an update on the previous such two reviews, refs. 1 and 2, presented to ICAS in 1984 and 1988.

Data Flow Determines Decomposition Scheme

One may partition the numerical task of supporting the design process in a number of different ways. For example, the partitions (subtasks) may correspond to engineering disciplines, physical components of the vehicle, or organizational units existing in the company. One way that was found to be quite useful lets the availability of mathematical models embodied in the computer codes establish a decomposition scheme.

To arrive at such a scheme one begins (refs. 3, 4) with taking an inventory of the major computer codes applicable for the vehicle design at hand. The inventory is then represented in a graph-theoretic form as shown in Fig. 1 as a system of interconnected modules that will also be called subsystems. The system of modules as in Fig. 1 is a mathematical model of a vehicle being designed. Each box in the diagram represents a computer code and the lines with the arrowheads depict the data flow among the codes. In this representation the codes are treated as black boxes so that the internal details are invisible and the focus is on the input/output data. Also, the graph does not in any way illustrate the execution sequence; i.e., it is not a flowchart, it is only concerned with the data flow.

Once the data flow is established, one moves on to determine the data completeness and the execution sequence with an aid of a table known as the N-square Matrix (ref. 5) portrayed in Fig. 2 for the system from Fig. 1. Each of the n black boxes from Fig. 1 is placed on the diagonal in a n-by-n table referred to as the N-square Matrix. For display purposes, each black box module is defined so that it is capable of transmitting output horizontally to the right and to the left and of accepting input vertically from above or from below, as indicated in Fig. 3. In the table, the data flow from module i to module j is represented by a dot at the intersection of i-th row and j-th column; while the absence of a dot means that no data are being transmitted. A dot indicates only that the data transmission occurs but does not define precisely what data items are being transmitted. To define that, a separate record has to be established in a way to be described later and stored. The output data sets corresponding to each dot in a row may not be mutually exclusive—it is possible that the same data items are being sent from module i to modules j, k, l etc. However, the input data sets represented by a dot in a column must be mutually exclusive, i.e., an input datum for module i must be coming from only one, and no more than one, source module. The N-square Matrix so defined is easy to record in a digital format module by module. The record of module k consists of its address i on the diagonal at the intersection of i-th row with i-th column; the i, j addresses of the dots in i-th row and the i, l addresses in i-th column. For each dot so addressed, there exists a list of the specific data items it represents.

To establish such an N-square Matrix, one begins with the modules placed on the diagonal in a random, or the best guess, order. Next, one scans the entire length of i-th column above and below the module i position. At l-th row one compares the i-th module input list with the l-th module output list to find out what data items from module l fit as input to module i. The data items so found are recorded in the i, l sets. After the entire length of column i is searched, the i-th module input data for which no source has been found are identified as the input external to the system. Alternatively, a new module or modules may be added to the system to supply these data. If more than one source was found for any data item, a choice must be made to make the input uniquely defined and recorded.

The above systematic procedure defines a data flow among the modules in the system. It is very effective in revealing the missing data. Also, it can be generalized beyond the computer code systems by broader interpretation of a module as a source of information, be it a computer code, an experiment, a data graph in a book, or a person's expert judgment. In this way, then, collective data knowledge of an entire engineering organization may be examined and recorded in a manner that is systematic and easy to store in a computer memory.

After defining the data flow, attention shifts to determining the best sequence of execution for the modules. In the convention that assumes execution order along the diagonal from the upper left corner, each dot in the upper right half of the matrix marks an instance of the data passed foward (feedforward). Conversely, a dot in the lower left half marks the instance of a feedback. Each instance of a feedback implies an iteration that may begin with the best guess at the input into module i from a downstream module j that itself may depend, directly or indirectly, on output from module i. Three such iterations (iterative loops) are indicated in Fig. 2 by the feedbacks from modules 1 to 3, 2 to 3, and 4 to 1.

The number of the feedback instances and of the associated iterations may be reduced by permutating the dots in the predecessor-successor module pairs on the diagonal and simultaneously permutating the dots in the rows and columns associated with these modules. In the system of Fig. 2, permutation of the modules to positions shown in Fig. 4 reduces the number of iterative loops from three to one.

While the conventional wisdom holds that one should attempt to eliminate the iterations or, at least to reduce their number in order to lessen the overall computational effort, the advent of parallel computing technology may suggest the opposite—it makes sense sometimes to reorganize the module execution sequence so as to create an opportunity for concurrent execution even if at the price of introducing iterations that might have been avoided. Such artificially created concurrent iterative computation may, if the convergence is sufficiently fast, be completed in a time shorter than the alternative without the iteration. The sequential and concurrent executions of modules coupled by data flow form characteristic patterns of the dots in the feed forward field as illustrated by two examples in Figs. 5a and b.

If the N-square Matrix is stored digitally as defined above, its permutations may be computerized to search either for iteration-minimizing patterns or for patterns that maximize opportunities for concurrent computation. Computer programs capable of doing this are beginning to be available; e.g., ref. 4. An example of an application to a large system of modules is shown in Figs. 6a and b, which portray the initial and improved sequences. In the improved sequence, the iterations have not only been reduced in number but also clustered. The group of modules tied together in a cluster of iterations will be referred to as a supermodule. One supermodule is highlighted with a heavy borderline in Fig. 6.

The iteration clustering is important because it imparts a hierarchic structure depicted in Fig. 7 to a system of supermodules that contain the clusters of the modules. The diagram in the figure is a graph-theoretic representation of the supermodule system termed hierarchic because the data flow only from a parent module to its children and not in reverse or among the children. That is not so inside the supermodules; hence the structures formed by the modules inside of supermodules are called non-hierarchic. The hierarchic structure of supermodules allows their sequential execution with opportunities for concurrent computations; e.g., supermodules within the group (2, 4, 6) and (3, 5, 7, 9) may be processed simultaneously. The entire system of modules and supermodules such as illustrated in Figs. 6 and 7 is referred to as a hybrid system. In the extreme, the hybrid system becomes exclusively hierarchic if each of its supermodules contains only a single module. In the other extreme, it becomes exclusively non-hierarchic if it consists of one and only one supermodule whose internal modules exhibit data connections such as those illustrated in Figs. 1 and 2.

Thus, the data flow defines the vehicle mathematical model as a hierarchic, non-hierarchic, or hybrid system. Optimization schemes for the hierarchic and non-hierarchic systems will be examined next.

Optimization of a Hierarchic System

A methodology for optimization of systems represented by mathematical models organized into a hierarchy such as the one formed by the supermodules in Fig. 7 became well-established in the last decade in a series of theoretical publications; e.g., refs. 2, 6, 7, 8, 9, and 10. It will, therefore, suffice here to restate briefly its foundation for a procedure that is known as the optimization by hierarchic decomposition.

For introductory purposes, the system is simplified to one of only one parent and one level of several children subsystems below as in Fig. 8 and is described in an entirely abstract way. Translation of that abstraction into specifics of a vehicle applications may be found in references to be cited later. The subsystems in Fig. 8 correspond to the supermodules in Fig. 7 and may further decompose internally.

The governing equations of the parent system may be written in the most compact form as

$$F(Y, X, P) = 0 \tag{1}$$

where F is a function vector, P is a vector of given parameters, X is a vector of the design variables, and Y is a vector of unknown behavior variables. Solution of eq. (1) is tantamount to analysis of the assembled system (the vehicle analysis) and it yields Y for an assumed X. Knowing Y and X, one can establish a vector Z to be an input transmitted from the parent to a child subsystem

$$Z = Z(Y, X) \tag{2}$$

where

$$Y = Y(X) \tag{2A}$$

by virtue of the solution of eq. (1).

In a subsystem, the local design variables are collected in vector x, the unknown behavior variables are elements in vector y, and the governing equations are written analogous to eq. (1) using a vector function f that depends on the elements of Z as parameters

$$f(y, x, Z) = 0 \tag{3}$$

Based on the solution of eq. (3), one may solve for one isolated subsystem a standard optimization problem for the independent variables x, while holding its input Z constant

$$\min_x \phi(y, x) \text{ subject to } g(y, x) \leq 0; \; h(y, x) = 0 \tag{4}$$

where $\phi(y, x)$ is an objective function, and $g(y, x)$ and $h(y, x)$ are the vectors comprising the inequality and equality constraint functions. The results of the above optimization are the constrained minimum of ϕ, denoted ϕ_{min}, the optimal values of x, designated x_{opt}, and the corresponding values of constraints designated g_{opt} and h_{opt}.

The above optimization is carried out for each child subsystem. Following that, the assembled system (the parent) is optimized using X as independent variables. Because X exerts influence on the subsystem optimal results through the functional relationships in eqs. (3), (2A), and (2), it is necessary to supply the system-level optimization procedure with the information about that influence; otherwise, the procedure could generate a change of X that would benefit the parent but harm the children.

The influence of X on the subsystem optimization results may be measured, to the first order of approximation, by the derivatives of these results with respect to X. To establish these derivatives, one begins with derivatives with respect to Z

$$\partial \phi_{\text{min}}/\partial Z; \; \partial X_{\text{opt}}/\partial Z; \; \partial g_{\text{opt}}/\partial Z; \; \partial h_{\text{opt}}/\partial Z \tag{5}$$

Considering the functional relations in eqs. (2) and (2A), one can extend the above by chain-differentiation to establish derivatives of the optimal results with respect to X

$$\begin{aligned}
\frac{d\phi_{\text{min}}}{dX} &= \frac{\partial \phi_{\text{min}}}{\partial Z}\left(\frac{\partial Z}{\partial X} + \frac{\partial Z}{\partial Y}\cdot\frac{\partial Y}{\partial X}\right)\\
\frac{dx_{\text{opt}}}{dX} &= \frac{\partial x_{\text{opt}}}{\partial Z}\left(\frac{\partial Z}{\partial X} + \frac{\partial Z}{\partial Y}\cdot\frac{\partial Y}{\partial X}\right)\\
\frac{dg_{\text{opt}}}{dX} &= \frac{\partial g_{\text{opt}}}{\partial Z}\left(\frac{\partial Z}{\partial X} + \frac{\partial Z}{\partial Y}\cdot\frac{\partial Y}{\partial X}\right)\\
\frac{dh_{\text{opt}}}{dX} &= \frac{\partial h_{\text{opt}}}{\partial Z}\left(\frac{\partial Z}{\partial X} + \frac{\partial Z}{\partial Y}\cdot\frac{\partial Y}{\partial X}\right)
\end{aligned} \tag{5A}$$

The sensitivity analysis algorithms of the type reviewed in ref. 11 may efficiently evaluate the partial derivatives in the above chain. The optimum sensitivity algorithms described in refs. 12 and 13 also apply.

The influence of X on the subsystem optimum may now be expressed in a general form referred to as an influence function

$$\gamma = \gamma(\phi_{\text{min}}, \, x_{\text{opt}}, \, g_{\text{opt}}, \, h_{\text{opt}}) \tag{6}$$

whose derivatives with respect to X may be obtained by chain-differentiation using the derivatives from eq. (5A)

$$\begin{aligned}
\frac{d\gamma}{dX} &= \frac{\partial\gamma}{\partial\phi_{\text{min}}}\cdot\frac{d\phi_{\text{min}}}{dX} + \frac{\partial\gamma}{\partial X_{\text{opt}}}\cdot\frac{dx_{\text{opt}}}{dX} + \frac{\partial\gamma}{\partial g_{\text{opt}}}\cdot\frac{dg_{\text{opt}}}{dX}\\
&\quad + \frac{\partial\gamma}{\partial h}\cdot\frac{dh}{dX}
\end{aligned} \tag{7}$$

The above derivatives substituted into the linear portion of the Taylor series provide an approximation to γ as a function of X to represent the influence of X on a particular subsystem.

$$\gamma = \gamma_{\text{opt}} + \frac{d\gamma}{dX}\,\Delta X \tag{8}$$

The above subsystem optimization and sensitivity analysis may be executed for all the subsystems concurrently because the subsystems do not directly exchange any data with each other.

The assembled system optimization that follows solves a standard optimization problem in the independent design variables X, the objective function Φ, and the constraint functions G and H

$$\begin{aligned}
&\min_X \Phi(\Gamma, Y, X) \text{ subject to } G(\Gamma, Y, X) \leq 0;\\
&H(\Gamma, Y, X) = 0;
\end{aligned} \tag{9}$$

Inclusion of the information about the influence of X on the subsystem optima in the above problem may be accomplished by using the influence functions γ that were defined in eq. (6) for this very purpose. The γ functions for all the n subsystem may be used to form a function vector Γ

$$\Gamma = \{\gamma_1, \gamma_2, \ldots \gamma_n\} \tag{10}$$

that appears as an additional argument in the objective function and the constraint functions in the optimization problem of eq. (9), so that

$$\Phi = \Phi(\Gamma, Y, X); \quad G = (\Gamma, Y, X); \quad H = H(\Gamma, Y, X) \tag{11}$$

Whenever X changes, the corresponding changes to the γ functions in Γ, may be approximated by the extrapolation in eq. (8).

The above describes a foundation shared by the methods forming a methodology for optimization of hierarchic systems. The particular methods differ in the formulation details of the functions ϕ, g, h, and γ in the subsystem optimization. In the system optimization, the differences are in the formulation details of the functions Γ, Φ, G, and H, and also in the way they incorporate Γ as an argument. Some of the references that elaborate on these formulation details were quoted at the beginning of this section. Further evolution of this methodology continues.

Application experience has accumulated a number of cases ranging from structural optimization by substructuring documented in refs. 9 and 14 to optimization of a large transport aircraft for fuel economy described in ref. 15. Application to the control of aeroelastic behavior was reported in ref. 16 in which the active controls and airframe were treated as two subsystems in a control-structure system. An example of a recent application provided in ref. 17 is an optimization of a two-stage launch vehicle, depicted schematically in Fig. 9, to maximize the payload placed into a specified orbit. In this application, the large and complex problem comprising optimizations of the lower (booster) stage and the upper stage vehicles and the launch trajectory was decomposed into subsystem problems: one for the booster and one for the upper stage, and a system level problem that adjusts the orbit parameters to maximize the payload.

In rotorcraft, a long-range, comprehensive development of optimization methodology for the rotor blades described in ref. 18 incorporates the hierarchic decomposition as its theoretical basis.

Optimization of Non-Hierarchic Systems

In a non-hierarchic system, every subsystem may, potentially, influence every other one, e.g., module 1 in Fig. 1. An approach to optimization of such systems that attracted attention during the last few years is based on derivatives of the system behavior (response) with respect to design variables. These derivatives are useful both for judgmental decision making and for formal optimization. The essential feature of the approach is an algorithm for system sensitivity analysis formulated in ref. 19. That algorithm, to be discussed next, decomposes the system sensitivity problem into a set of subsystem sensitivity problems while preserving the subsystem couplings.

System Sensitivity Analysis

A system of fully interconnected modules shown in Fig. 10 is an example convenient for introducing the algorithm. The number of modules limited to three is large enough to develop a solution pattern that generalizes to any number of modules. To have a physical reference in mind, consider the system in Fig. 10 as simulating an actively controlled flexible wing. Then, let module α be a mathematical model for aerodynamics, e.g., a CFD code, and the modules β and γ be mathematical models for a structure (a finite element program) and for a control system.

Prerequisite to the sensitivity algorithm is the system analysis. It amounts to finding a solution to the governing equations of the system written as a function vector whose arguments are the vector of the design variables X and the vectors of unknown behavior variables Y.

$$F(X, Y_\alpha, Y_\beta, Y_\gamma) = 0 \tag{12}$$

Each vector Y is the output from the module identified by the subscript; for instance, Y_β may be a vector of structural displacements. Each module input consists of X and Y from the other modules. It also may contain constant parameters P that are dropped as irrelevant to this discussion. Typically, solving eq. (12) which may comprise nonlinear analysis, for instance in the CFD module α, requires iterations among the modules; e.g., iterating between modules α and β to determine aerodynamic loads and deformations of a flexible wing.

When the system is solved, each module is temporarily isolated for the purposes of sensitivity analysis that yields derivatives of the module output with respect to its inputs of Y and X. These derivatives are then placed as coefficients in the set of simultaneous, linear algebraic equations called the Global Sensitivity Equations (GSE). Specifically, the derivatives with respect to the Y inputs are collected in the Jacobian matrices identified by a pair of subscripts. For example,

$$J_{\alpha\gamma} = [\partial Y_\alpha / \partial Y_\gamma] \tag{13}$$

An element i, j in this Jacobian matrix is the derivative of the pressure coefficient at the i-th location on the wing surface with respect to the deflection angle of the j-th control surface. The Jacobian matrices fill the off-diagonal submatrix positions in a square matrix of coefficients on the left-hand side of GSE

$$\begin{bmatrix} I & -J_{\alpha\beta} & -J_{\alpha\gamma} \\ -J_{\beta\alpha} & I & -J_{\beta\gamma} \\ -J_{\gamma\alpha} & -J_{\gamma\beta} & I \end{bmatrix} \begin{Bmatrix} dY_\alpha/dX_k \\ dY_\beta/dX_k \\ dY_\gamma/dX_k \end{Bmatrix} = \begin{Bmatrix} \partial Y_\alpha/\partial X_k \\ \partial Y_\beta/\partial X_k \\ \partial Y_\gamma/\partial X_k \end{Bmatrix} \tag{14}$$

where I are the identity submatrices. The derivatives with respect to a particular element X_k of X are placed in the right-hand side vector. The number of the right-hand side vectors equals the number of X_k's of interest.

The unknowns in eq. (14) are the derivatives of the system behavior Y with respect to X. These derivatives account for the coupling amoung the modules, even though the derivatives in the Jacobian matrices and in the right-hand side vector are obtained from the sensitivity analyses of the modules treated as if they were isolated. To emphasize this, the derivatives obtained from the solution of eq. (14) are termed the total derivatives, later referred to as the System Design Derivatives (SDD), while the other derivatives are recognized to be partial derivatives.

Typically, the GSE matrix is block-sparse because each off-diagonal Jacobian corresponds to a particular output-to-input transmission of the Y data. The same is true for the right-hand side vector since some modules may not be directly affected by a particular X_k. For instance, an X_k representing a cross-sectional dimension in the wing structure will not influence directly the outputs from the aerodynamic and control analyses, hence only the right-hand side vector partition corresponding to the module β will be non-zero.

Complete details of the GSE derivation and a discussion of the solvability conditions may be found in ref. 19. In ref. 20, the above sensitivity analysis was extended to the derivatives of higher order.

Utility of System Design Derivatives

The SDD's are useful in several ways. They are effective in quantifying, for judgmental purposes, the degree of influence of the design variables on the system behavior. An example from ref. 3 is illustrated in Fig. 11. The behavior variable of interest is the range of a general aviation aircraft. The range is influenced by structural weight fraction of the total weight, and the C_l/C_d ratio that is affected by the wing elastic deformations. Hence, the range will depend to some extent on the wing cover thickness. Formally, this may be represented as the behavior of a system depicted on top of

Fig. 11a. The Breguet range equation from the PERFORMANCE module is also shown in Fig. 11a. A highly idealized finite element model of the wing from the STRUCTURES module is illustrated in Fig. 11b. Change of the thickness t in one of the wing cover panels affects, as the arrows in Fig. 11a show, the weight, elastic deformations, aerodynamics, and ultimately, the terms in the range equation. The influences on the weight and aerodynamics are conflicting, hence it is difficult to assess judgmentally the ultimate effect of t on the range. A precise measure of that effect was obtained in form of the SDD values from the solution of the GSE for the system portrayed in Fig. 11a. Normalized values of the range derivatives with respect to the thickness of the four cover panels on the top surface of the wing are represented by vertical bars in Fig. 11b. This type of information when available on line may foster the designers' insight into the cause-effect relationships that should be considered in their decisions.

The SDD's play a key role in formal optimization because most of the optimization algorithms rely on gradients in searching the design space. A procedure for such optimization is shown in Fig. 12. The system analysis and sensitivity analysis discussed above appear as two consecutive operations in the chart. An obvious opportunity for concurrent processing occurs in the sensitivity operation. The operation of approximate analysis usually involves an extrapolation, such as the use of the linear part of the Taylor series. The iterative loop back to the system analysis in the procedure is necessary because in the general case of a nonlinear system the SDD's are valid only in the neighborhood of the system solution.

It is essential in this procedure to use normalized (logarithmic) partial and total derivatives in the system sensitivity analysis to eliminate the effect of differences in the order of magnitude of the variable values that may exist because of differences in the units of measure. That effect may be detrimental to the numerical search. Normalized derivatives are also easy to interpret because they have a uniform meaning of the percent of change of the dependent variable caused by one percent increment of the independent variable. Another caveat is that the volume of data transmitted from one module to another should be kept as low as possible by a judicious use of reduced basis techniques to avoid excessive dimensionality of the Jacobian matrices in GSE.

It is noted that the procedure of Fig. 12 can also be used for optimization of a hierarchic system because the GSE exists for such systems. In that case, the GSE matrix is populated with the Jacobian matrices on only one side of the diagonal hence the GSE solution cost is greatly reduced.

Non-hierarchic System Optimization Examples

Applications of the above procedure have been growing in number much faster than those for hierarchic systems; apparently the non-hierarchic systems occur relatively more often. The applications may be categorized by the level of the analysis employed in the modules.

An example of the application in which the analyses representing major engineering disciplines contributing to aircraft design were deliberately kept at the conceptual design level described in ref. 21 was reported in ref. 22. The subject of the study was a short-takeoff, medium-range heavy transport and the purpose was to show that a formal optimization based on the SDD data obtained from GSE may be combined with the classical parametric study method to investigate how the major configuration design variables influence the aircraft performance. Demonstrating that such a combined approach may be effective constitutes an important contribution because conventionally the parametric studies and the formal optimization based on nonlinear programming were regarded to be mutually exclusive methods.

One of many results furnished in ref. 22 is reproduced in Fig. 13. It shows the take-off gross weight as a function of the cruise Mach number for a prescribed set of constraints that included the required range, maximum allowed take-off run length, etc. The curves labeled 1 to 4 correspond to the different sets of design variables as follows: 1—aspect ratio, wing area; 2—as in set 1 plus the wing sweep angle; 3—as in set 2 plus the airfoil depth; 4—as in set 3 plus taper and cruise altitude. Each point on the curves represents an aircraft configuration optimized by means of the procedure illustrated in Fig. 12 executed for the corresponding Mach number value treated as a constant parameter, using the variables specified in the above sets as elements of X.

Thus, the study was, in effect, a two-level approach. Parameters, such as the Mach number in Fig. 13, were varied systematically as the higher-level design variables. At a selected setting of these variables, the optimization procedure was carried out operating on the configuration variables treated as the lower level, more detailed design variables.

Another application in the same category was described in ref. 23 in which an unconventional transport aircraft with three lifting-surfaces was optimized by the procedure of Fig. 12 using the shape and positions of the lifting surfaces as design variables. The configuration in its initial state (baseline) and after the fourth iteration of the optimization procedure is depicted in Fig. 14. In addition to significant numerical results, this application has also demonstrated that the inherent parallelism in the system sensitivity analysis can be exploited by having members of the engineering team calculate concurrently the partial derivatives for the GSE.

To close the sample of results in this category, applications to hypersonic, single-stage-to-orbit aircraft and to a hypersonic, long-range interceptor were reported in refs. 3 and 24, respectively. In the former, the procedure of Fig. 12 improved the propulsive efficiency index by nearly 13% using the configuration and structural design variables. This was regarded as a very significant gain because the initial configuration procedure was already refined by extensive parametric studies. A similar improvement was noted in the hypersonic interceptor case in which a reduction of the take-off gross weight of 13% was achieved.

An example of an application in which the modules entailed analysis at the level more typical for a preliminary design phase was reported in refs. 25 and 26. That application objective was the development of a methodology for advanced aircraft optimization; a generic supersonic transport aircraft depicted in Fig. 15 was selected as a test case. The above development included systematic organization of the methodology numerical process by means of the N-square Matrix discussed previously. The graph-theoretic representation of the modules in the mathematical model of the supersonic aircraft is illustrated in Fig. 16, and the sequence of the module executions that minimizes the number of iterative loops is portrayed in Fig. 17 in the N-square Matrix format. The module execution sequence in that figure was obtained by means of the software described in ref. 4. Optimization results available in refs. 25 and 26 were limited to those obtained from a system simplified to three modules shown in Fig. 18. A sample of these results is portrayed in Figs. 19 and 20. The former shows the contour plots of the Tsai-Hill criterion constraint which was one of the constraints active in the composite cover of the wing. In the initial design that constraint was well satisfied indicating that the wing structure had some unnecessary material. This state corresponds to the initial point on the optimization histogram illustrated in latter figure. As indicated by the descending weight plot in Fig. 20, optimization removed that unnecessary material and in the process rendered the Tsai-Hill constrain critical in some areas of the wing cover.

The plot continuation in Fig. 20 shows how the configuration study was progressing, including judgmental, discrete changes such as raising the wing cover minimum gage; reducing the sandwich core thickness in the wing cover panels; and switching from a composite material to titanium. The methodology was apparently effective in bringing the system in only very few iterations to a new optimal plateau after each such judgmental design intervention.

The application examples quoted in the preceding two sections have been carried out for systems either completely hierarchic or completely non-hierarchic. So far, no experience was reported with optimization of a truly hybrid system. However, considering success of the above two methods, one may anticipate that the next step will be development of a procedure in which a hybrid system of supermodules, such as the example in Fig. 7, will be optimized by the hierarchic decomposition method employing the non-hierarchic system optimization in each supermodule.

Correlating Simplified and Refined Analyzes

Because of their modular nature, both the hierarchic and non-hierarchic optimization methods described above may accommodate disciplinary analyses of various levels of refinement without changing their procedural organization. Consequently, one may anticipate development of a capability for a coordinated use of analyses at different levels of sophistication. A step in this direction is a technique described in refs. 27 and 28.

To summarize that technique, consider a physical phenomenon to be represented by two mathematical models: a relatively crude but inexpensive to analyze model A and a relatively refined and correspondingly more expensive to analyze model B. At the beginning of optimization, one analyzes both models and obtains results R_A and R_B. The correlation factor β is now introduced, defined as

$$\beta = R_B/R_A \tag{15}$$

Because both R_A and R_B are functions of X design variables, $R_A = R_A(X)$ and $R_B = R_B(X)$, the derivatives of β exist

$$d\beta/dX = (dR_B/dX\ R_A - R_B\ dR_A/dX)/R_A^2 \tag{16}$$

where dR_A/dX and dR_B/dX are obtained from the respective sensitivity analyses. If R_A and R_B are vectors then β is a vector, and $d\beta/dX, dR_A/dX$, and dR_B/dX are the Jacobian matrices.

To save computational costs of repetitive use of model B in the ensuing steps of the optimization procedure, one may now use model A instead and apply a correction formula to approximate R_B

$$(R_B)\text{approx} = R_A\ (\beta_o + d\beta/dX\ \Delta X) \tag{17}$$

which reflects the influence of X on β to the first order of accuracy. In nonlinear problems,the values of β_o and $d\beta/dX$ have to be periodically updated.

Effectiveness of the above technique was demonstrated in ref. 28 in which the object was wing structure, model A was the wing plate representation, and model B was the wing refined finite element model. An example of one of the $(R_B)_{\text{approx}}$ results corrected as in eq. (17) was the first natural frequency whose error was kept to only about 1% for the cross-sectional design variable changes of the order of more than 100%. One may anticipate that type of approximate analysis to be especially useful in applications that require nonlinear aerodynamics analysis. Computational costs of that analysis grows exponentially with its sophistication level relative to the linear analysis as illustrated in Fig. 21 (ref. 25). Using linear analysis as model A corrected by β as above could provide a compromise needed in optimization between accuracy and computational cost. Encouraging progress in that direction was already reported in ref. 29.

Concurrent Subspace Optimization (CSSO)

The optimization method for non-hierarchic systems described in the aforegoing uses decomposition limited to the system sensitivity analysis only. Once the SDD's are obtained, the system optimization is treated as a single problem. This is in contrast to the hierarchic system optimization in which the system optimization itself is divided into subsystem optimizations. It was recognized in ref. 30 that it would be advantageous to extend decomposition in non-hierarchic systems beyond sensitivity analysis so as to optimize the subsystems separately, similar to the way it is done in the hierarchic systems.

An algorithm to do this was introduced in ref. 30 and, subsequently, developed and tested in ref. 31. The algorithm is based on two key ideas: all the subsystems that have an influence on a constraint should share responsibility for that constraint satisfaction, and all the subsystems should share the same objective function.

An example of a wing treated as a system combining aerodynamics and structures illustrates the above idea. Each of the two disciplines is being represented as a module, and they are coupled through the aerodynamic loads-deformation data exchange. Suppose that in the initial design there is a violated stress constraint caused by bending at the wing root. That constraint might be satisfied by the purely structural means of cross-sectional resizing, or by reducing the wing aspect ratio which is a variable traditionally in the domain of aerodynamics. The algorithm engages both disciplines in this case—aerodynamics and structures—into satisfaction of the stress constraint by dividing its value between the two disciplines in a proportion determined by a factor r; i.e.,

$$g_S \leq g_0\ r;\ g_A \leq g_0(1-r) \tag{18}$$

where $g_0 > 0$ is the value of the violated constraint g, and g_S and g_A are the parts of g to be satisfied separately as constraints in the structural and aerodynamic optimizations, respectively. Both of these optimizations use a common system-level objective function, which in this example might be drawn from the aircraft performance; e.g., the flight range. The two disciplinary optimizations may be executed concurrently. Following that, a system-level coordinating optimization is performed to adjust the r factor to improve the common objective function and to maintain satisfaction of all the constraints. The method readily generalizes to the case of n modules in a non-hierarchic system. The subsystem and system optimizations depend on the sensitivity data obtained from GSE.

The above method became known as the CSSO because it is related to a nonlinear mathematical programming method that formally divides the design space into subspaces. It is still in the early development stage, but some application experience beyond the test cases in ref. 31 have begun to emerge. An example is a solar energy recovery system whose CSSO-based optimization was reported in ref. 32. It is anticipated that the CSSO approach has a potential to become a unified method for hybrid systems including their purely non-hierarchic extreme.

Disciplinary Sensitivity Analysis

All the optimization methods for hierarchic and non-hierarchic systems discussed in the aforegoing rely on the disciplinary sensitivity data. Even though one may obtain such data by finite-differencing techniques, the computational costs and potential accuracy problems of these techniques motivated in the recent two decades development of the disciplinary quasi-analytical sensitivity analyses that are intrinsically superior to finite differencing. For the optimization methods discussed herein, these techniques may be regarded as enabling technology.

So far, the quasi-analytical sensitivity analysis has become mature and generally available only for structures where it is based on differentiation of the governing equations (the load-deflection equations) and solution of the resulting simultaneous, linear algebraic equations that comprise derivatives as unknowns. Reference 11 provides a survey of literature. Recently, beginning of a similar development in CFD has become apparent; e.g., refs. 33–43. Because the higher order CFD codes are usually very expensive to execute, continuation of the above development to the production level is important for making the optimization methodology discussed in this writing widely used in aircraft design.

One may anticipate that with structures and aerodynamics paving the way, development of sensitivity analysis in other engineering disciplines will follow. In the meantime, finite differencing remains available as an inferior but still usable alternative.

Discussion of sensitivity analysis would be incomplete without mentioning the new technology of Automatic Differentiation (AD). This technology has been successfully used in the nuclear industry for a number of years but has only recently come to the attention of aerospace engineers. The AD principles are described in ref. 44. In a nutshell, to use an AD approach for computing derivatives of output Y with respect to the input X for an existing code C, one has to use a special AD code, let it be called ADC, as a tool. Several ADC codes are now available, commercially and in the public domain. The ADC reads C, and for a C line that is an assignment statements of the type $a = f(b)$ it performs symbolic differentiation to obtain da/db. However, that symbolic differentiation is performed only internally to evaluate the numerical value of da/db. The derivative analytical expression is not carried forward; only its numerical value is. If on a subsequent line one finds the variable a on the right-hand side; e.g., $c = f(a)$, then a chain differentiation is invoked to obtain $dc/db = dc/da\ da/db$. The chain derivatives are concatenated numerically from the beginning to the end of the code C to obtain the derivatives of dY/dX. The product of ADC processing C is a new source code, let it be called NC, which is the original code C augmented with the calls to the special subroutines in ADC that do the above differentiation. It is remarkable that NC reproduces all the loops and if-branches of C.

The new code NC may then be used to produce the same output Y that C did and, in addition, it yields dY/dX with computational efficiency better than that of finite differencing and with accuracy equal to that of analytical differentiation. For an engineer the principal advantage of AD seems to stem from its bypassing the software development that otherwise would be required by any of the alternative, disciplinary, quasi-analytical, sensitivity analysis methods previously discussed. For that reason alone, *AD* might be a potential breakthrough. An example of some initial applications in engineering was reported in ref. 45.

Genetic Optimization Algorithms

Up to this point in the paper it was tacitly assumed that the $Y = f(X)$ are continuous functions, the X are continuous variables, and that there is no problem with local minima. In many applications these assumptions are not so, hence it is useful to have methods available that are capable of handling problems with discontinuities and local minima. The so-called genetic algorithms are one family of methods that, in addition to other merits, showed promise to do that.

Genetic algorithms simulate the improvement process that occurs naturally in the biological evolution of a species. Adapted to engineering design, the basic conceptual elements of the algorithm are: 1) random generation of a population of designs that differ by the values of design variables; 2) evaluating a measure of fitness for each design in the population; and 3) mating the designs in pairs to produce offspring. The measure of fitness is a function whose value depends on the degree of satisfaction, or violation, of the constraints and on the values of the objective functions (the approach is intrinsically suitable to handle multiobjective problems). The probability of an individual design participation in the mating process is made to rise with the individual design measure of fitness. The features of the mating parents are passed to the offspring by a probabilistic mechanism that ensures that the offspring inherits the parents' features and that occasional mutations occur which produce new offspring features not present in the parents.

Thus, the offspring population replacing the parent population has the measures of fitness improved on the average and, due to the mutations, superior features occur in some offspring to initiate a new line of evolution as a way out of global minima.

To date no vehicle system applications have been reported. However, encouraging results from optimization of wing structure, described in ref. 46, showed that the approach was very effective in homing on the neighborhood of the global optimum in the design space. This application also showed that for locating the optimum more precisely in that neighborhood, it is better to switch to a gradient-based search. This suggests that the genetic algorithms may be regarded as complementary to that type of search. Regarding the applicability range, it is expected that the computer technology progress will reduce the cost of generating large, statistically significant populations required by the genetic approach to the level where application to entire vehicle systems will become economically feasible.

Design of Experiments Methods

Recently, a renewed interest was noted in optimization methods based on the Design of Experiments (DOE) approach. Under that approach, a number of designs is placed as design points in the design variable space spanning the domain of interest. Each such design behavior may be evaluated by any suitable method, including experiments, statistics from past experience, etc. Behavior variable of interest may be approximated as an explicit function, called the response function, fitted to the design points. Generation of the designs, their evaluation, and the response function fitting constitute an initial investment to be recouped in optimization in which the need for behavior data may then be satisfied at a negligible computational cost by evaluating the explicit response functions.

This approach has a long history dating back to ref. 47. An example of usefulness for aircraft design is an application to transport aircraft engine selection in ref. 48. The method does not require sensitivity analysis of the designs placed in the design space, hence it can accommodate discrete variables. Another advantage of the method is that the design points may be generated concurrently and new ones may be added as the design process progresses. On the other hand, the method has a major drawback of requiring a large number of design points that grows exponentially with the number of the design variables. That growth can be moderated somewhat by various statistically based schemes for strategic placement of a reduced number of the design points, but its exponential character cannot be removed because of the combinatorial nature of the method.

Two reasons may be discerned for renewed interest in DOE. The first one is the current emphasis on taking into account in designing the entire life cycle of the product, including manufacturing, maintenance, and disposal, all dominated by cost. These considerations are difficult to model mathematically in the same sense as conventional engineering disciplines but can be accounted for by statistical and experimental data at the design points. The second reason is the success of the orthogonal arrays, also referred to as the Taguchi arrays, in systematic improvement of the industrial product quality. These arrays readily adapt to DOE as a tool for limiting the number of the design points. From a DOE standpoint, the orthogonal array technique is simply a way to place a set of design points in the design variable space in such a way that the maximum of information may be extracted from it. This is achieved by making the vectors comprising the coordinates of the design points orthogonal to each other; each such vector constitutes a column in the orthogonal array. The vector orthogonality removes duplication of the information contained in each design point. The technique does not eliminate the exponential growth problem mentioned above, and the orthogonal arrays commonly available in a tabular form usually represent each variable at no more than three settings which only accounts for the lowest order of nonlinearity. One also needs a prerequisite knowledge of the variable interactions to choose the best array type for the application at hand. A comprehensive assessment of the orthogonal arrays in the DOE context is given in ref. 49.

Despite the limitations, the DOE approach enhanced by the orthogonal arrays proved its usefulness in a growing number of applications. An excellent recent example is the optimization of a single-stage-to-orbit vehicle in ref. 50.

Massively Parallel Computers

All the methods discussed herein strain the present capacity of the computer. The CPU time required by CFD (illustrated in Fig. 21), amplified by the repetitive use of analysis in design, makes that point very clear. Fortunately, the exponential growth of computer speed and capacity is certain to continue even though the speed of a conventional serial machine appears to be approaching natural physical limits. The new way to continue that growth is through development of massively parallel computers. A systematic development program in that direction is described in ref. 51. The aim is to bring the effective computational speed measured in the floating point operations per second into the trillion range. This will require parallelization of computing both at the equation level and at the module level. In the former, the internal code in a module must be rewritten for maximum use of concurrently operating processors. In the latter, the internally unchanged modules execute simultaneously, each on its own processor.

The methods discussed in this paper are all amenable to parallelization at the module level, preserving the investment in existing software. Beginning at the module level with the existing software will provide at least partial benefits from the parallel computing early, before massive investment in a new software parallelized at both the equation and module level pays off.

Conclusions

Starting from an axiomatic "divide and conquer" premise, the basic schemes for decomposing the large optimization problem of aircraft into smaller problems were examined. It was shown that if the vehicle system mathematical model is considered as an assemblage of modules, each module representing a mathematical model of a physical phenomenon (an engineering discipline) or behavior of a vehicle component, then the data flow among the modules defines three basic system organizations: hierarchic, non-hierarchic, or hybrid.

Key ideas and essential features of the optimization methods that have evolved for each of the above system organizations were discussed with a selection of references cited for more details. Particular attention was given to the sensitivity analyses at the discipline and system levels, which are at the core of each of the above methods. Alternative methods were pointed out for applications in which discontinuities of the functions and variables, local minima, and scarcity of analytical models limit usability of the derivative-based methods.

The picture emerging from this review is that of several diverse methods and techniques coalescing into a new, rapidly crystallizing methodology that enables optimization of aerospace vehicles as systems in which everything affects everything else. Far from attempting to supplant the human designer, the methodology is predicated on decomposing the large system optimization problems into smaller ones to be worked concurrently by groups of specialists in engineering organization supported by parallel processing of data.

Development needed to accelerate application of the above methodology entails sensitivity analyses in the key engineering disciplines, other than structures for which such analysis has already been established. The new technology of automated differentiation has a potential for facilitating this development which must also include techniques for trading accuracy for execution speed in mathematical modeling. Finally, the quantum jump in computing speed promised by the new technology of massively parallel computers is seen as a necessary part in the subject methodology development.

The reviewed methodology has the potential for supporting designers in their work with nearly instantaneous answers to quantitative "what if" questions. The result will be a mind-computer, synergistic environment in which human creativity will thrive.

References

1. Sobieszczanski-Sobieski, J.; Barthelemy, J.F.M.; and Giles, G. L.: Aerospace Engineering Design by Systematic Decomposition and Multilevel Optimization. Int'l Council of Aeronautical Sc., 14th Congress, Toulouse, Sept. 1984, Paper No. ICAS-84-4.7.3.
2. Sobieszczanski-Sobieski, J.: Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results. Int'l Council for Aeronautical Sc., 16th Congress, Jerusalem, Aug.–Sept., 1988; Proceedings of, Vol. 2, pp. 953–964.
3. Sobieszczanski-Sobieski, J.: A System Approach to Aircraft Optimization, Presented at AGARD Structures and Materials Panel 72nd Meeting, Bath, United Kingdom, May 1991, AGARD Report, AGARD-R-784. Paris, Feb. 1982, Chapter 2.
4. Rogers, J. L.: A Knowledge-Based Tool for Multilevel Decomposition of a Complex Design Problem; NASA TP-2903,1989.
5. Steward, Donald V.: Systems Analysis and Management: Structure, Strategy and Design. Petrocelli Books, Inc., 1981.
6. Sobieszczanski-Sobieski, J.: A Linear Decomposition Method for Large Optimization Problems—Blueprint for Development; NASA TM-83248, February 1982.
7. Thareja, R.; and Haftka, R. T.: Numerical Difficulties Associated with Using Equality Constraints to Achieve Multi-level Decomposition in Structural Optimization, AIAA Paper No. 86-0854, AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics, and Materials Conference, San Antonio, Texas, May 1986.
8. Haftka, Raphael T.: An Improved Computational Approach for Multilevel Optimum Design, Journal of Structural Mechanics, Vol. 12, No. 2, pp. 245–261, 1984.
9. Sobieski, J.: Optimization by Decomposition in Structural and Multidisciplinary Applications. Lecture Notes: Volume 1-NATO/DFG ASI Optimization of Large Structural Systems, Berchtesgaden, Sept. 1991.
10. Sobieszczanski-Sobieski, J.: Two Alternative Ways for Solving the Coordination Problem in Multilevel Optimization. NASA TM-104036, August 1991.
11. Adelman, H. A.; and Haftka, R. T.: Sensitivity Analysis of Discrete Structural Systems. AIAA J., Vol. 24, No. 5, May 1986, pp. 823–832.
12. Sobieszczanski-Sobieski, J.; Barthelemy, J. F.; and Riley, K. M.: Sensitivity of Optimum Solutions to Problem Parameters. AIAA J., Vol. 21, Sept. 1982, pp. 1291–1299.
13. Barthelemy, J., F.; and Sobieszczanski-Sobieski, J.: Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programming. AIAA J., Vol. 22, No. 6, June 1983.
14. Sobieszczanski-Sobieski, J.; James, B. B.; and Dovi, A. R.: Structural Optimization by Multilevel Decomposition. AIAA J., Vol. 23, No. 11, November 1985.
15. Wrenn, G. A.; and Dovi, A. R.: Multilevel Decomposition Approach to the Preliminary Sizing of a Transport Aircraft Wing. AIAA Journal of Aircraft, Vol. 25, No. 7, July 1988, pp. 632–638.
16. Weisshaar, T. A.; Newsom, J. R.; Gilbert, M. G.; and Zeiler, T. A.: Integrated Structure/Control Design-Present Methodology and Future Opportunities. ICAS-86-4.8.1, Sept. 1986.
17. Beltracchi, Todd J.: A Decomposition Approach to Solving the Allup Trajectory Optimization Problem. AIAA-90-0469. AIAA 28th Aerospace Sciences Meeting, Reno, NV, Jan. 1990.
18. Adelman, H. M.; and Mantay, W. R.: Integrated Multidisciplinary Design Optimization of Rotorcraft. AIAA Journal of Aircraft, Vol. 28, No. 1, January 1991.
19. Sobieszczanski-Sobieski, J.: On the Sensitivity of Complex, Internally Coupled Systems; AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, VA, April 1988. AIAA Paper No. CP-88-2378, and AIAA J., Vol. 28, No. 1, Jan. 1990, also published as NASA TM-100537, January 1988.
20. Sobieszczanski-Sobieski, J.: Higher Order Sensitivity Analysis of Complex, Coupled Systems. AIAA Journal, Vol. 28, No. 4, Apr. 1990.
21. Malone, B.; and Mason, W. H.: Multidisciplinary Optimization in Aircraft Design Using Analytic Technology Models. AIAA-91-3187, AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, September 1991, Baltimore, MD.
22. Mason, W. H.: Analytic Models for Technology Integration in Aircraft Design. AIAA Paper No. 90-3262,

AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, September 1990, Dayton, OH.

23. Barnum, J.; Bathras, C.; Beene, K.; Bush, M.; Kaupin, G.; Lowe, S.; Sobieski, I.; Tingen, K.; and Wells, D.: Advanced Transport Design Using Multidisciplinary Design Optimization. AIAA 91-3082, Aircraft Design Systems and Operations Meeting, September 1991, Baltimore, MD.

24. Consoli, R. D.; and Sobieszczanski-Sobieski, J.: Application of Advanced Multidisciplinary Analysis and Optimization Methods to Vehicle Design Synthesis. 17th Congress of the International Council of the Aeronautical Sciences (ICAS), Stockholm, September 1990, Proceedings of.

25. Coen, P. G.; Sobieszczanski-Sobieski, J; and Dollyhigh, S. M.: Preliminary Results from the High-Speed Airframe Integration Research Project. AIAA Paper No. 92-1004; 1992 Aerospace Design Conference, February 1992, Irvine, CA.

26. Barthelemy, J.-F.; Wrenn, G. ; Dovi, A.; and Coen, P.: Integrating Aerodynamics and Structures in the Minimum Weight Design of a Supersonic Transport Wing. AIAA 92-2372, Presented at AIAA/ASME/ASCE/AHS/ASC 33rd Structures, Structural Dynamics, and Materials Conference, Dallas, TX, April 1992, Proceedings of.

27. Haftka, R. T.: Combining Global and Local Approximations, AIAA Journal, Vol. 29, No. 9, pp. 1523–1525, Sept. 1991.

28. Chang, K. J.; Haftka, R. T.; Giles, G. L.; and Kao, P.-J.: Sensitivity Based Scaling for Correlating Structural Response from Different Analytical Models. AIAA Paper No. 91-0925. Presented at AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, 1991; Proceedings of.

29. Hutchinson, M.; Unger, E.; Mason, W.; Grossman, B.; and Haftka, R.: Variable Complexity Aerodynamic Optimization of an HSCT Wing Using Structural Wing Equations. Paper AIAA 92-0212 presented at Aerospace Sciences Meeting & Exhibit, January 6–9, 1992, Reno, NV.

30. Sobieszczanski-Sobieski, J.: Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems, Recent Advances in Multidisciplinary Analysis and Optimization. Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization; Hampton,Virginia, September 28–30, 1988; Proceedings published as NASA CP-3031.

31. Bloebaum, Christina L; Formal and Heuristic System Decomposition Methods in Multidisciplinary Synthesis. NASA Contractor Report 4413, NASA Langley Research Center, December 1991.

32. Eason, E. D.; Wright, J. E.; and Nelson, E. E.: Software for Multidisciplinary System Optimization. SBIR Phase I Final Report, NASA Langley Research Center, August 1991.

33. Yates, E. C., Jr.; and Desmarais, R.: Boundary Integral Method for Calculating Aerodynamic Sensitivities with Illustration for Lifting Surface Theory. Proceedings of the International Symposium of Boundary Element Methods (IBEM 89), published by Springer-Verlag, Oct. 2–4, 1989, East Hartford, Conn.

34. Elbanna, H. M.; and Carlson, L. A.: Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes. Journal of Aircraft, Vol. 27, No. 6, June 1990, pp. 507–518, also AIAA Paper 89-0532.

35. Yates, E. C., Jr.: Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady Nonplanar Lifting-Surface Theory. NASA TM-100502, NASA Langley Research Center, September 1987.

36. Sobieszczanski-Sobieski, J.: The Case for Aerodynamic Sensitivity Analysis. Proceedings of the Symposium on the Sensitivity Analysis in Engineering. NASA CP-2457, NASA Langley Research Center, 1987.

37. Bristow, D. R.; and Hawk, J. D.: Subsonic Panel Method for the Efficient Analysis of Multiple Geometry Perturbations. NASA CP-3528, 1982.

38. Jameson, A.: Aerodynamic Design Via Control Theory. NASA CR-181749, also ICASE Report No. 88-64, November 1988.

39. Shubin, G. R.; and Frank, P., D.: A Comparison of the Implicit Gradient Approach and the Variational Approach to Aerodynamic Design Optimization. Applied Mathematics and Statistics Technical Report, AMS-TR-163, Boeing Computer Services, Seattle, Washington, April 1991.

40. Dulikravich, G. S.; Aerodynamic Shape Design and Optimization. AIAA Paper 91-0476, AIAA 29th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 1991.

41. Baysal, O., and Eleshaky, M. E.; Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics. AIAA Paper 91-0471, AIAA 29th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 1991.

42. Taylor, A. C. III,; Hou, G. W.; and Korivi, V. M.: A Methodology for Determining Aerodynamic Sensitivity Derivatives with Respect to Variation of Geometric Shape. Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, April 8–10, Baltimore, MD, AIAA Paper 91-1101, also to appear, AIAA Journal, as A Methodology for Calculating Aerodynamic Sensitivity Derivatives.

43. Newman, P. A.; Hou, G. J.-W; Jones, H. E.; Taylor, A. C. III; and Korivi, V. M.: Observations on Computational Methodologies for Use in Large Scale Gradient Based Multidisciplinary Design Incorporating Advanced CFD Codes. NASA TM-104206, NASA Langley Research Center, February, 1992.

44. Griewank, A.: The Chain Rule Revisited in Scientific Computing. SIAM News, Part 1, May 1991, pp. 20-21, part 2, July 1990, pp. 9–24.

45. Barthelemy, J.-F. M.; and Hall, L. E.: Automatic Differentiation as a Tool for Engineering Design. 4th AIAA/Air Force/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, Ohio, September, 1992. Proceedings of.

46. Hajela, P.; and Lin, C.-Y.: Genetic Search Strategies in Multicriterion Optimal Design. AIAA 91-1040, AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, 1991; Accepted for publication in Journal of Structural Optimization, 1992.

47. Fisher, R. A. (1966): The Design of Experiments. 8th Edition, Hafner Publishing Company, New York.

48. Healy, M. J.; Kowalik, J. S.; and Ramsay, J. W.: Airplane Engine Selection by Optimization on Surface Fit Approximations. Journal of Aircraft, Vol. 12, No. 7, July 1975.

49. Montgomery, D. C.: Design and Analysis of Experiments. Published by John Wiley & Sons, 1991.

50. Stanley, D.; Unal, R.; and Joyner, R.: Application of Taguchi Methods to Dual Mixture Ratio Propulsion System Optimization for SSTO Vehicles. AIAA Paper 92-0213, AIAA 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1992.

51. Holst, T. L.; Salas, M. D.; and Claus, R. W.: The NASA Computational Aerosciences Program-Toward Teraflops Computing. AIAA 92-0558, AIAA 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1992.

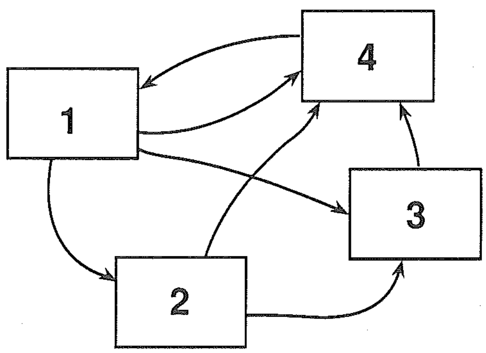

Figure 1. System of four modules (subsystems) in graph-theoretic format.

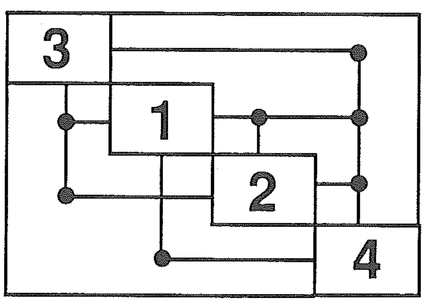

Figure 2. System of four modules in a N-square Matrix format.

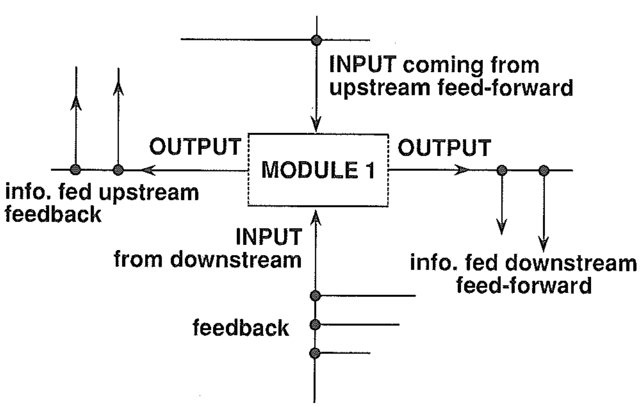

Figure 3. Definition of a module for N-square Matrix format.

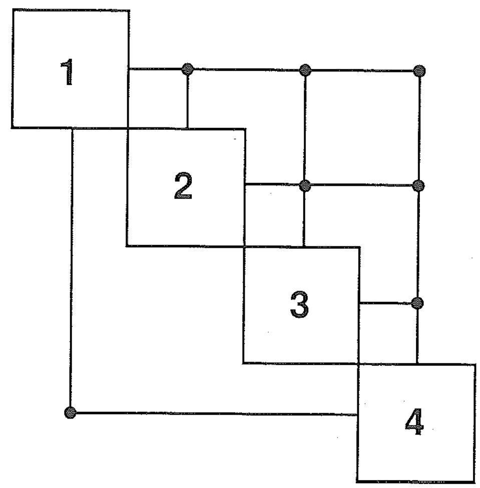

Figure 4. N-square Matrix for improved execution sequence.

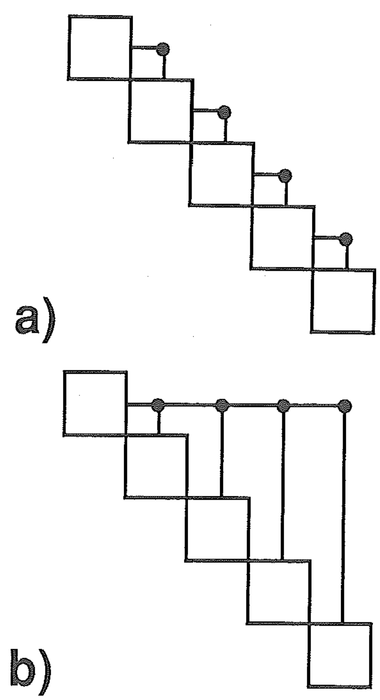

Figure 5. N-square Matrix patterns for executions: a) sequential, b) parallel.

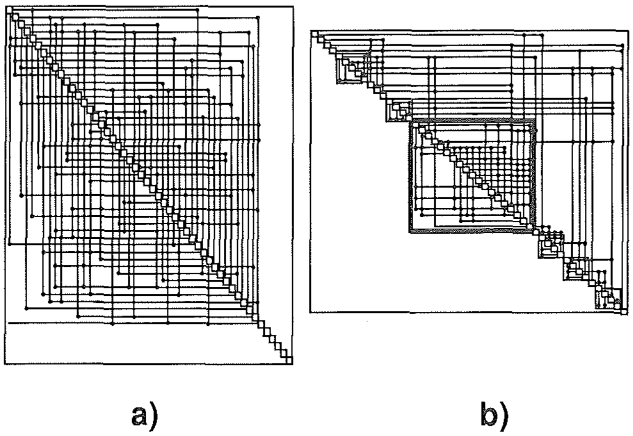

Figure 6. Large system execution: a) initially random, b) improved organization; in an N-square Matrix format.

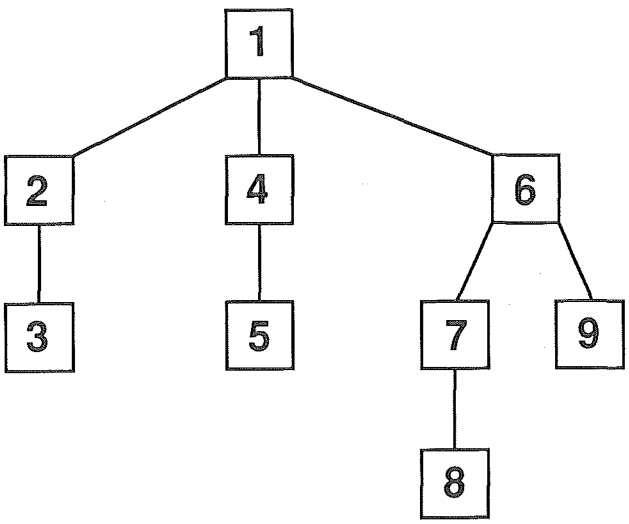

Figure 7. A hierarchic system.

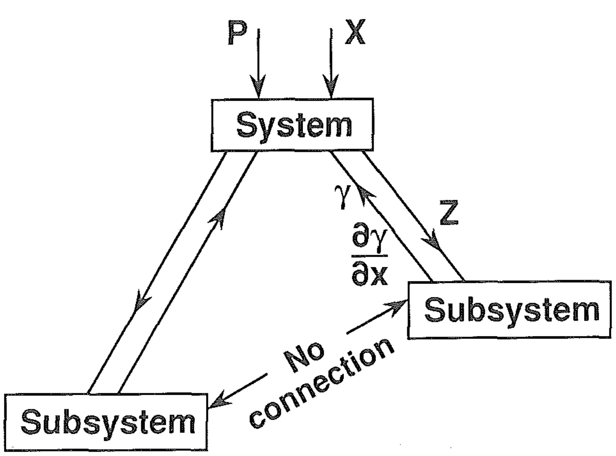

Figure 8. Data flow in a hierarchic system optimization by decomposition.

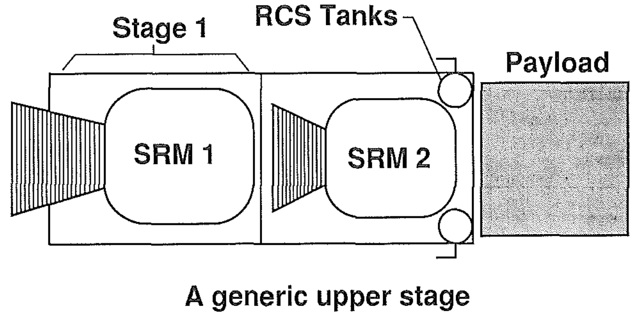

Figure 9. Two-stage launch vehicle with a generic upper stage with payload.

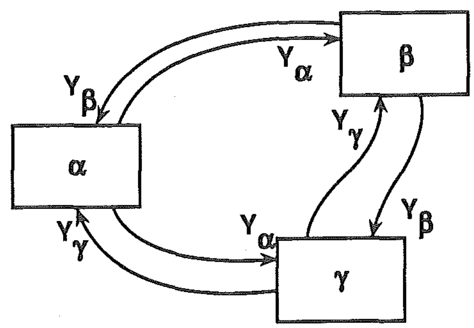

Figure 10. A non-hierarchic system example.

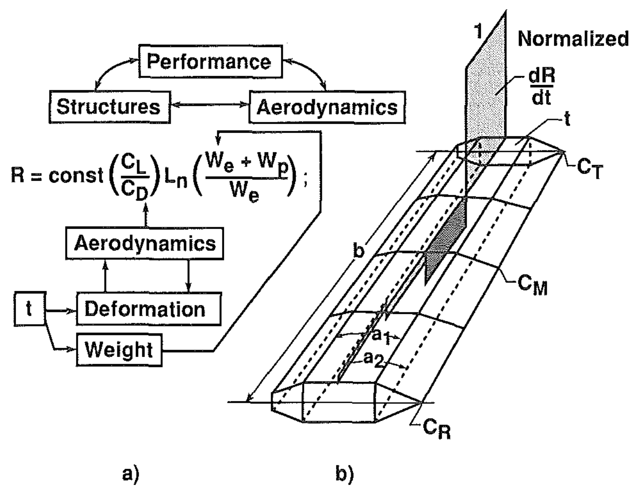

Figure 11. a) System of mathematical models, the Breguet's formula, and the channels of influence of the wing cover thickness on the aircraft range; b) Vertical bars illustrate magnitudes of derivatives of range with respect to thickness (normalized by the largest positive derivative value).

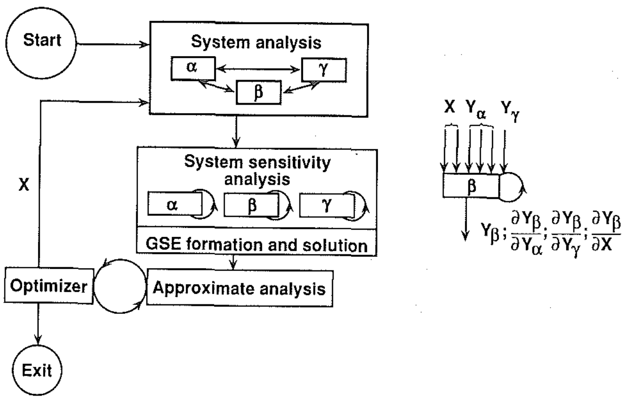

Figure 12. Flowchart of non-hierarchic system optimization procedure.

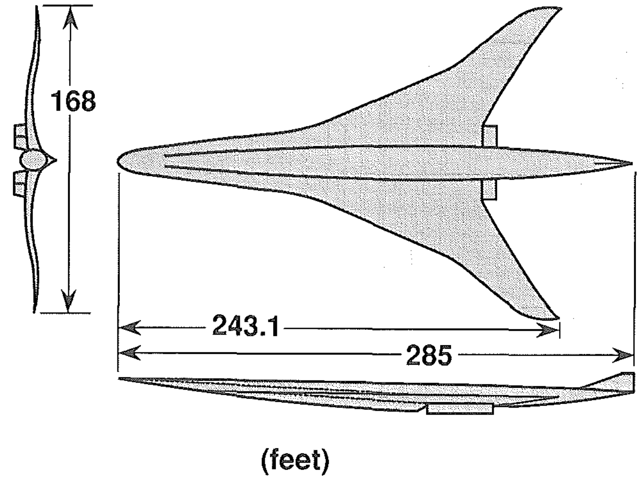

Figure 15. A generic supersonic transport configuration.

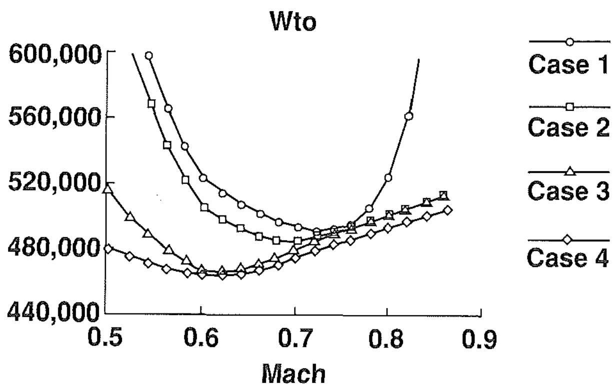

Figure 13. Minimum Take-off Gross Weight (Wto) as function of Mach number for cases defined in the text.

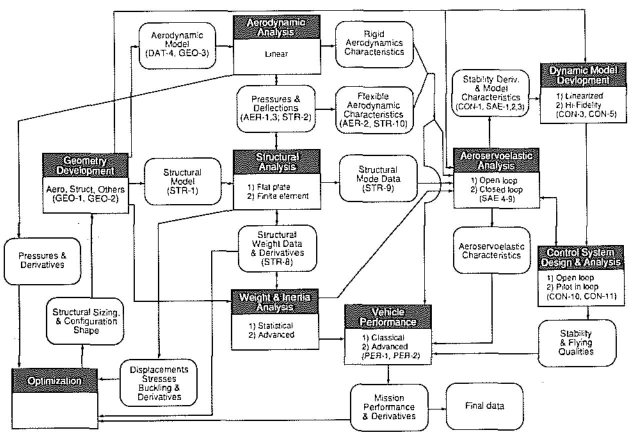

Figure 16. Modules for supersonic transport analysis in a graph-theoretic format.

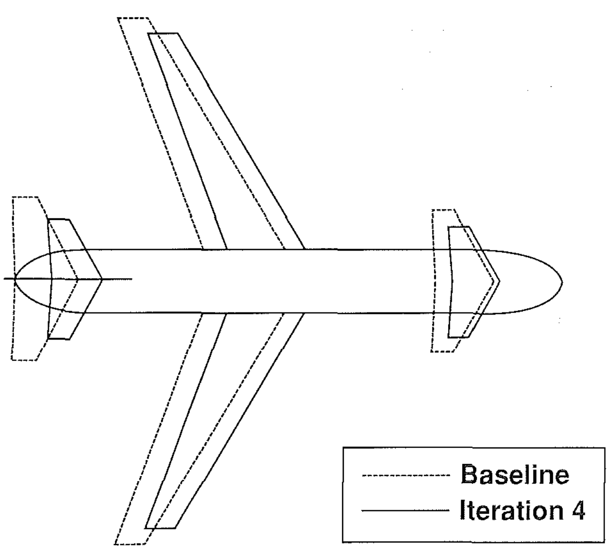

Figure 14. Unconventional transport aircraft configuration with three lifting surfaces: the baseline and after the 4th iteration optimization.

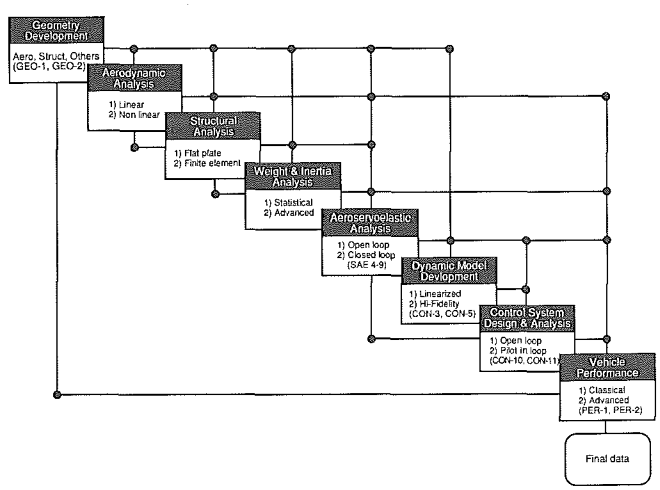

Figure 17. Modules for supersonic transport analysis sequenced for a minimum of iterative loops in an N-square Matrix format.

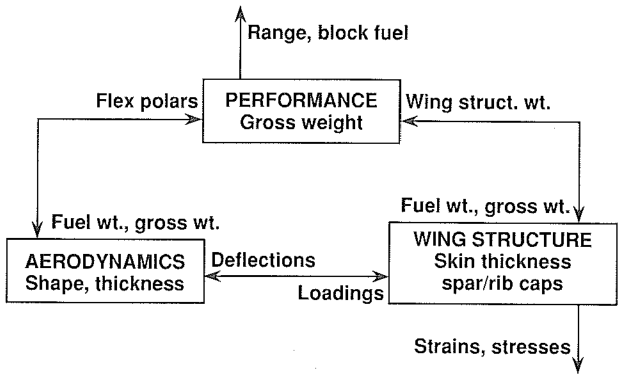

Figure 18. Supersonic transport analysis simplified to three disciplines.

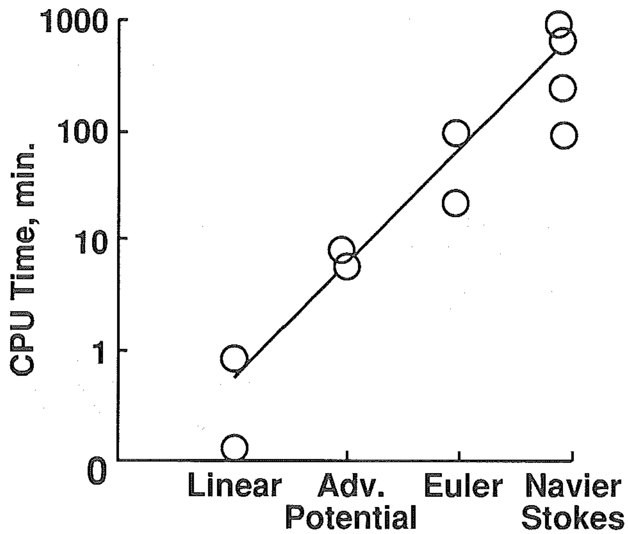

Figure 21. Typical CPU time requirements for CFD at various levels of fidelity.

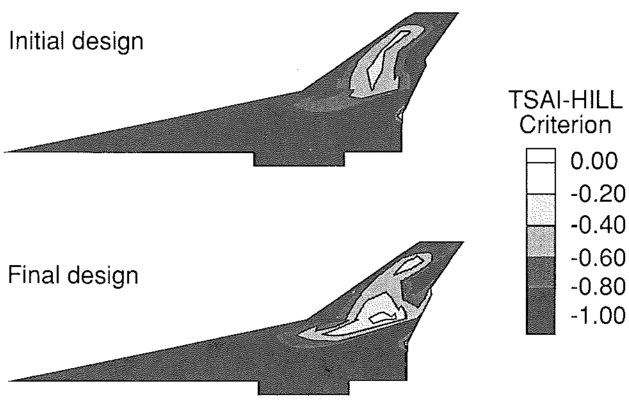

Figure 19. Supersonic transport wing: contour plots of the Tsai-Hill criterion values for the wing composite material covers.

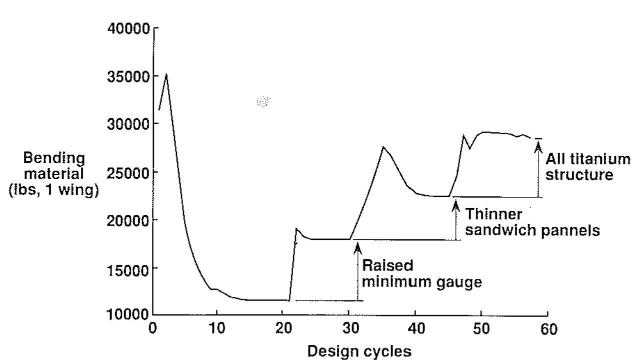

Figure 20. History of the wing bending material weight in the optimization process.