J Optim Theory Appl (2009) 142: 533–556
DOI 10.1007/s10957-009-9520-9

Coupled Aerostructural Design Optimization Using the Kriging Model and Integrated Multiobjective Optimization Algorithm

X.B. Lam · Y.S. Kim · A.D. Hoang · C.W. Park

Published online: 17 March 2009

Abstract The paper develops and implements a highly applicable framework for the computation of coupled aerostructural design optimization. The multidisciplinary aerostructural design optimization is carried out and validated for a tested wing and can be easily extended to complex and practical design problems. To make the framework practical, the study utilizes a high-fidelity fluid/structure interface and robust optimization algorithms for an accurate determination of the design with the best performance. The aerodynamic and structural performance measures, including the lift coefficient, the drag coefficient, Von-Mises stress and the weight of wing, are precisely computed through the static aeroelastic analyses of various candidate wings. Based on these calculated performance, the design system can be approximated by using a Kriging interpolative model. To improve the design evenly for aerodynamic and structure performance, an automatic design method that determines appropriate weighting factors is developed. Multidisciplinary aerostructural design is, therefore, desirable and practical.

Communicated by K.K. Choi.

The authors acknowledge the support of a Korea Research Foundation Grant funded by the Korean Government and the second stage of Brain Korea 21st project.

X.B. Lam · Y.S. Kim · A.D. Hoang · C.W. Park (✉)
Department of Mechanical and Aerospace Engineering, ReCAPT, Gyeongsang National University, Gajwadong 900, Jinju, Gyeongnam, South Korea
e-mail: parkcw@gnu.ac.kr

X.B. Lam
e-mail: binhlamxuan@gnu.ac.kr

Y.S. Kim
e-mail: dudtkddldh@gnu.ac.kr

A.D. Hoang
e-mail: haduong83@gnu.ac.kr

Keywords Fluid/structure interface (FSI) · Global optimization · Multiobjective optimization · Kriging model

1 Introduction

Multidisciplinary design optimization (MDO) [1–14] has received considerable attention in the aircraft industry. MDO encompasses an extensive research area that includes the implementation of high-fidelity analysis tools in both aerodynamic and structural fields, investigations of robust interfacing algorithms for coupling these tools and improvement of the optimization algorithms so as to predict the best performances quickly. Scientists in this area have focused attention on three main categories, embracing the accuracy, robustness and expensiveness of the proposed algorithms for application to realistic design problems effectively. For instance, Sobieski and Haftka [1] found that sound coupling and optimization methods were shown to be extremely important since some techniques, such as sequential discipline optimization, were unable to converge to the true optimum of a coupled system. On the other hand, Wakayama [2] showed that in order to obtain realistic wing planform shapes with aircraft design optimization, it is necessary to include multiple disciplines in conjunction with a complete set of real-world constraints.

To develop the analysis tools, the aerospace researchers have incessantly enhanced the quality as well as the fidelity of the applied codes to predict the system responses. Walsh et al. [3], for example, investigated the progresses of high-speed civil transport (HSCT) design in detail. Originally, the HSCT2.1 design was realized by using low-fidelity analysis tools. A panel code with a low number of grid points was combined with an equivalent laminated plate analysis code to progress with design optimization. Meanwhile, HSCT3.5 was a multidisciplinary application that integrated medium-fidelity analysis tools, including a marching Euler code and a finite element analysis code with a limited number of mesh points. In the HSCT4.0 design, high-fidelity tools, incorporating the CFL3D Navier-Stokes flow solver and the GENESIS structural analysis package, were utilized in the design process. Alternatively, Martins [4] employed SYN107-MB Euler/Navier-Stokes computational fluid dynamics (CFD) module and FESMEH computational structural mechanics (CSM) module for his research of small business jet design. The high-fidelity Euler/Navier-Stokes CFD and CSM packages have correspondingly become the state-of-the-art analysis modules in MDO field. Moreover, Venkataraman and Haftka [5] also explored the factors, such as the degrees of freedom (DOF), the topology of the structure, the types of analysis, etc., that make the increasing complexity of a structural optimization problem. This would be very meaningful for the designers to enhance the fidelity as well as the efficiency of analysis and design. Besides, the flexible aerodynamic grid can be handled by using a grid generation package (Kim et al. [6]), or grid deformation algorithm WARPMB (Martins [4]), or hybrid grid deformation algorithm as presented in this paper, etc. The structural mesh can be managed by using a CSM mesh generator as stated in this paper, etc.

In addition, Kamakoti [15] and Guruswamy [16] conducted a statistical analysis of fluid/structure interaction algorithms. A remarkable amount of interfacing tech-

niques was enumerated correlative to their grades in application. Those were the infinite plate spline (IPS), the thin plate spline (TPS), the multiquadratic biharmonic (MQ), the finite plate spline (FPS), the nonuniform rational B-spline (NURBS) and bilinear interpolation (BI). The first technique is appropriate for linear analytical fluid models and modal approach structure models, while the last technique is highly suitable for the full Navier-Stokes flow solver and the three-dimensional (3D) finite element structural solver. On the other hand, Martins [4] also suggested his extrapolative techniques to transfer the interactive data during the process of aeroelastic analysis. Particularly, Hounjet and Meijer [17] evaluated elastomechanical and aerodynamic transfer methods, comprising of surface spline interpolation (SSI) and volume spline interpolation (VSI), for nonplanar configurations. In general, these SSI and VSI methods are relatively simple, efficient and simultaneously adaptive to the conservation of virtual work. Consequently, they are used widely and become very popular interfacing algorithms in the field of aeroelasticity.

The improvement of optimization algorithms is also an active research area in aerospace design. The researchers in this area initially considered various traditional optimization methods, such as gradient-based optimization [4, 5, 9–11], as effective tools to enhance their designs. The efficiency of gradient-based optimizer can significantly be enhanced by using the adjoint method [4, 5, 9–11]. Nevertheless, gradient-based is only a local optimizer hence can not determine the global optimum. Furthermore, the application of a global optimization algorithm for MDO system is a time-consuming activity and is nearly impossible to carry out in reality. Many scientists have considered imitating the design problem as a virtual problem in order to overcome the above difficulties. Imitating the design problem as a virtual problem implies approximating the problem to be designed by a set of basic equations that can accurately simulate the system responses. Thus far, there have been several efficient approximation methods, such as the response surface method (RSM) [6–8, 18], the artificial neural networks (ANN) [19–21], the multivariate adaptive regression splines (MARS) [22], the nonuniform rational B-spline (NURBS) [23], the extended radial basis function (ERBF) [24, 25], the Kriging method (KM) [26–32], the support vector regression (SVR) [33], the moving least square (MLS) [34], etc., that can successfully be applied for design optimization. According to our experience, KM, ERBF, SVR and MLS are state-of-the-art metamodels due to their high efficiency and accuracy. After being approximated by metamodelings, the design system needs to be improved and optimized by using several famous global optimization algorithms, such as genetic algorithm (GA) [35–40], simulated annealing (SA) [40–44], evolutionary multiobjective optimization algorithms (EMOA) [45–47], etc.

In general, MDO has become an increasingly interesting research area in aerospace science. The development of computational design methods reduces the overall design costs and turn-around time for the development of aerospace technology. The use of high-fidelity tools also brings more confidence to the design. On the scope of this paper, good-fidelity analysis tools were employed to validate and improve the MDO system. The commercial computational fluid dynamics (CFD) code FLUENT [48] and the 3D finite element analysis (FEA) code were coupled to execute the static aeroelasticity and optimization process. High-fidelity interfacing algorithms were also investigated. volume spline interpolation (VSI) [17], defined

relying on the 3D biharmonic equation which adapts to the conservation of virtual work, is used as a load transfer module that maps the aerodynamic pressure onto structural mesh. The deflections obtained from structural analysis can be transmitted onto the CFD grid using reversed VSI. The new CFD grid can be regenerated by using a robust grid deformation algorithm. This deforming algorithm is a combination of the expensive spring analogy [49] and inexpensive transfinite interpolation [50–53]. The CSM mesh can be managed by using a CSM mesh generator. Moreover, the research has utilized the Kriging method [26–32] as an approximation model to imitate the system responses precisely. A multiobjective design procedure was also developed to enhance the performances evenly and moderately. The procedure here is an integrated optimizer that employs the robust genetic algorithm [35–40] for the entire design process.

2 Fluid/Structure Interface

The aerodynamic and structural performances of aerodynamic bodies are tightly coupled. Structural deformation will change the distribution of the aerodynamic force on the body surface. In contrary, this alternative force makes a reverse influence on the structural deformation. Therefore, the efficient fluid/structure interface (FSI) [15, 16, 54, 55] should be developed in an effort to predict the system responses accurately. The FSI can be classified broadly under three major categories: fully coupled, loosely coupled and closely coupled analyses. Full coupling can be done by reformulating and resolving the governing equations that combine the fluid and structural equations of motion in time simultaneously. Meanwhile, the loose and close couplings, unlike the full coupling, solve the fluid and structural equations using two separate modules of solvers. To do this, these suggested approaches must execute an interfacing technique to exchange the information back and forth between the two modules. The loosely coupled approach has only external interactions between the fluid and structure modules; the information is exchanged after partial or complete convergence. With the closely coupled method, the fluid and structure modules can be coupled into a single module with exchanges of information taking place at the interface or the boundary via an interface module [15]. As the fully coupled FSI is a very expensive approach, many scientists prefer employing the loose or close coupling to resolve static and dynamic aeroelastic phenomena. The typical loose fluid/structure coupling is shown in detail in the flow diagram of Fig. 1.

The fluid/structure interface is an iterative process that connects five principal modules together, involving computational fluid dynamics (CFD), computational structural mechanics (CSM), CFD grid generation or deformation, CSM mesh generator and data transfer (implying load and displacement transfer) modules. For each of iteration, it is necessary to map the surface loads from the CFD grid system onto the structural grid to obtain the forces on the CSM mesh system, which are then used to obtain the displacements on the CSM mesh. These displacements need to be interpolated onto the CFD surface grid to obtain a new CFD grid. This repetitive process is repeated until the convergent criterion is satisfied. The stopping condition is merely fulfilled if no considerable changes of the structural mesh are created.

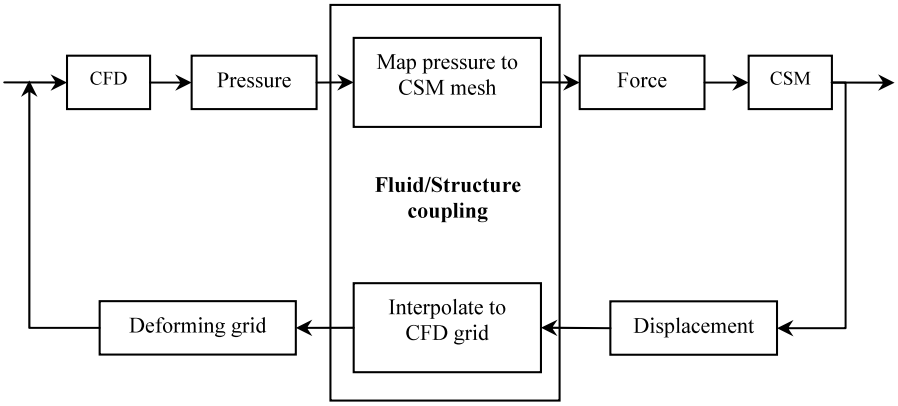


Fig. 1 Fluid/structure interface

2.1 Aerodynamic Analysis

The aerodynamic analysis package used in this paper is the commercial CFD code FLUENT [48]. FLUENT is a high-fidelity and relatively-automatic flow solver, based on the finite volume method [56–60], that integrates many viscous and turbulence modelings while resolving Navier-Stokes equation. It can completely be considered as an effective fluid flow analysis module for executing coupled aerostructural design optimization. In this paper, the Spalart-Allmaras viscous modeling is integrated in the design process in order to precisely predict the aerodynamic performance. The initial multiblock structured CFD grid is generated using the Gridgen [61] package. This CFD grid can be altered using a robust grid deforming algorithm.

2.2 Structural Analysis

The process of structural analysis can be executed by a high-fidelity, fully-automatic and robust structural analysis code ADFEAP, which is developed from the original research code of Professor O.C. Zienkiewicz and Professor R.L. Taylor. The package is a structured finite element [62–66] solver that incorporates several element types, embracing truss, beam, shell, solid, etc, elements. The code can be effectively employed for usual structural analysis as well as investigating and verifying the novel structural algorithms. The CSM mesh is automatically created using the GiD [67] mesh generator.

2.3 Grid Deformation Algorithm

The grid deformation code used in this paper is based on the combination of a typical algebraic (spring analogy [49]) and iterative (transfinite interpolation [50–53]) method. The displacement of the vertices and edges is computed by the expensive spring analogy, while the displacement of the remaining grid points is specified by the inexpensive transfinite interpolation (TFI).

The spring analogy developed in this paper is a segment, not a vertex, model. Each fictitious segment spring has its own stiffness [49]

$$k_{ij} = \frac{\lambda}{[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{\beta}}, \tag{1}$$

where λ and β are used to control the stiffness of the grid cells, while (x_i, y_i, z_i) and (x_j, y_j, z_j) are the coordinates of the two spring points.

The displacement of the vertices in each block is defined from the equations of static equilibrium [49]

$$\sum_{j=1}^{v_i} k_{ij}(\delta_i^n - \delta_j^n) = 0. \tag{2}$$

Here, δ_i and δ_j denote the displacements of node i and j, respectively, while v_i is number of neighbors of node i.

After retrieving the deformation of all vertices, the displacement of the interior grid points can be determined using the arclength-based TFI. This TFI algorithm is composed of three steps: parameterization, computation of the deformation and reproduction of the new grid. In the phase of parameterization, all grid points are parameterized according to the global i, j and k indices of the arclength. All of the arclengths on the surface are computed by adding all of the displacements along the grid line. This is done by keeping the other two indices constant and varying the third. Consider a sampled normalized arclength parameter in the i direction for any fixed j and k surface [50, 53]

$$s_{1,j,k} = 0,$$

$$s_{1,j,k} = s_{i-1,j,k} + d\{(i-1, j, k), (i, j, k)\} \tag{3}$$

for $i = 2, \ldots, i_{max}$, where i_{max} is the maximum grid index in the i direction and $d\{(i-1, j, k), (i, j, k)\}$ is the distance between two grid points $(i-1, j, k)$ and (i, j, k).

The i-direction normalized arclength parameter for this grid line is inversely proportional to the maximum i-index parameter [50, 52, 53],

$$F_{i,j,k} = \frac{s_{i,j,k}}{s_{i\,max,j,k}}. \tag{4}$$

Similarly, the grid parameters for the grid lines in the j and k directions, denoted by $G_{i,j,k}$ and $H_{i,j,k}$, can be determined using the same way.

In the phase of the deforming computation, the TFI formula is used to solve the displacement of the interior points of block edges, surfaces and volumes. The one-dimensional (1D) TFI in the i direction is simply [50]

$$\Delta E_{i,1,1} = (1 - F_{i,1,1})\Delta P_{1,1,1} + F_{i,1,1}\Delta P_{i\,max,1,1}, \tag{5}$$

where ΔE refers to the deformation of an edge, which in this example varies in the i index, whereas $\Delta P_{1,1,1}$ and $\Delta P_{i\,max,1,1}$ are the deformations of the two corner points of the edge. Essentially, the two-dimensional and three-dimensional (2D and 3D) equations can be developed successfully from the one-dimensional case. For instance, a 2D equation for a surface in the $k = 1$ plane is specified as [50]

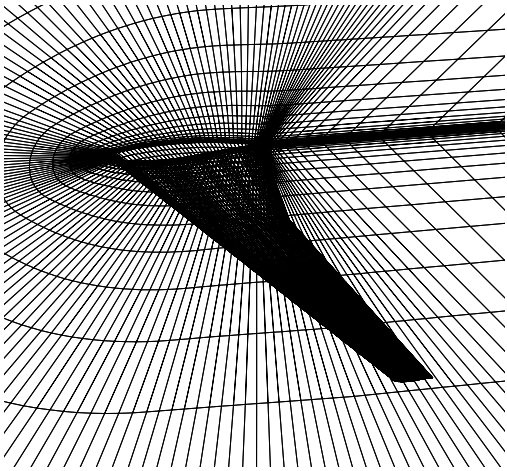


Fig. 2 Original wing-only grid

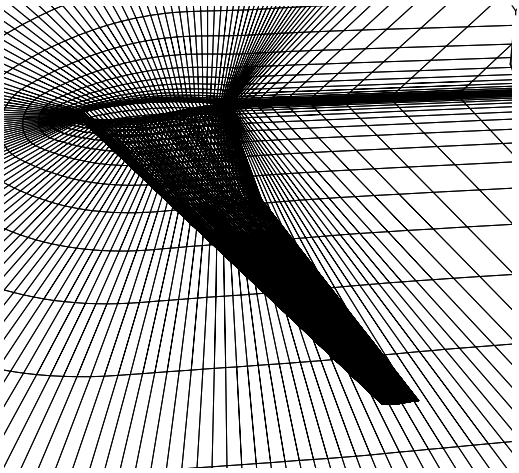


Fig. 3 Deformed wing-only grid

$$\begin{aligned}\Delta S_{i,j,1} = {} & (1 - F_{i,j,1})\Delta E_{1,j,1} + F_{i,j,1}\Delta E_{i\,max,j,1} \\ & + (1 - G_{i,j,1})(\Delta E_{i,1,1} - (1 - F_{i,j,1})\Delta P_{1,1,1} - F_{i,j,1}\Delta P_{N,1,1}) \\ & + G_{i,j,1}(\Delta E_{i,j_{max},1} - (1 - F_{i,j,1})\Delta P_{1,N,1} - F_{i,j,1}\Delta P_{N,N,1}), \qquad (6)\end{aligned}$$

where ΔS, ΔE, ΔP refer to the surface, edge and corner points deformations, respectively.

The deformation of all grid points is finally added to the original positions to obtain the new grid; this is known as the process of grid reproduction. On the whole, this is a fast and robust grid deformation algorithm which can efficiently support multidisciplinary design optimization. A typical example of this deforming algorithm is presented in Fig. 2 and Fig. 3. The original wing in Fig. 2 is extended by an amount equal to 30% of semi-span to form a new wing as shown in Fig. 3.

2.4 Data Transfer

In coupled aerostructural analyses, the information has to be exchanged between elastomechanical and unsteady aerodynamic simulation programs. The information concerns the structural deformation connected to the elastomechanical grid and aerodynamic forces connected to the aerodynamic grid. As aerodynamic and elastomechanical models are based on grids with different structures, interpolation procedures which transfer aerodynamic and elastomechanical data between the elastomechanical and aerodynamic surface grids must be developed. It is of fundamental importance that no energy is lost in this transfer. Consequently, the forces on the structural grid and the deflections on the aerodynamic grid are restricted by [17]

$$\begin{aligned} \{f^s\} &= [G_{as}]^T \{f^a\}, \\ \{u^a\} &= [G_{as}]\{u^s\} \end{aligned} \tag{7}$$

which adapts to the conservation of virtual work. $\{f^s\}$, $\{f^a\}$ and $\{u^s\}$, $\{u^a\}$ are in turn forces and deflections on structural and aerodynamic mesh, while $[G_{as}]$ is the interpolation matrix. This matrix clearly depends on the shapes of both grids and must be calculated by a reliable interpolation algorithm. In keeping with the scope of this paper, a simple, effective and robust technique, termed volume spline interpolation (VSI) [17], is implemented. The VSI is a very simple method which does not require any additional logic and can be applied straightforwardly to any 3D data set, without drifting so far away from the original data even the original data is non-smooth. The volume spline function can be essentially defined by relying on the 3D biharmonic equation [17]

$$h = d_0 + \sum_{m=1}^{N^{S+}} d_m E_m. \tag{8}$$

Here, $E_m = \sqrt{(x^a - x^s)^2 + (y^a - y^s)^2 + (z^a - z^s)^2}$, N^{S+} is the number of structural points together with one additional constraint, (x^a, y^a, z^a) denotes the coordinates of the aerodynamic points, and (x^s, y^s, z^s) denotes the coordinates of the structural points.

The coefficients d_m can be determined from the equations of equilibrium [17],

$$\begin{aligned} &\sum_{m=1}^{N^{S+}} d_m = 0, \\ &d_0 + \sum_{m=1}^{N^{S+}} d_m E_m = h^l, \quad l = 1, \ldots, N^{S+}. \end{aligned} \tag{9}$$

To utilize this algorithm, a prolongation matrix $[G^*]$ has to be constructed [17],

$$[G^*] = [A][C]^{-1}, \tag{10}$$

where

$$[C] = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & E^s_{11} & E^s_{12} & \cdots & E^s_{1N^{S+}} \\ 1 & E^s_{21} & E^s_{22} & \cdots & E^s_{2N^{S+}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & E^s_{N^{S+}1} & E^s_{N^{S+}2} & \cdots & E^s_{N^{S+}N^{S+}} \end{bmatrix}, \tag{11}$$

$$[A] = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & E^a_{11} & E^a_{12} & \cdots & E^a_{1N^{S+}} \\ 1 & E^a_{21} & E^a_{22} & \cdots & E^a_{2N^{S+}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & E^a_{N^{a+}1} & E^a_{N^{a+}2} & \cdots & E^a_{N^{a+}N^{s+}} \end{bmatrix}, \tag{12}$$

with

$$E^s_{lm} = \sqrt{(x_l - x_m)^2 + (y_l - y_m)^2 + (z_l - z_m)^2}, \tag{13}$$

$$E^a_{lm} = \sqrt{(x^a_1 - x_m)^2 + (y^a_l - y_m)^2 + (z^a_l - z_m)^2}. \tag{14}$$

Finally, the interpolation matrix $[G_{as}]$ is obtained from $[G^*]$ by deleting the first column [17],

$$[G^*] = [0 \quad G_{as}]. \tag{15}$$

3 Optimization Algorithms

The research utilizes a robust Kriging approximation method and an effective multi-objective optimization algorithm that is an integration of the two cycles of the genetic algorithm.

3.1 Kriging Model

The Kriging [26–32] model postulates a combination of a global trend function $P(x)$ and a local deviated function $Z(x)$ of the following form [29, 30, 32]:

$$\hat{y}(x) = P(x) + Z(x), \tag{16}$$

where $\hat{y}(x)$ is the unknown function of interest, $P(x)$ is a known polynomial (normally constant, linear or quadratic) function of the p-dimensional-variable x and $Z(x)$ is the realization of a normally distributed stochastic process in which the covariance structure of Z relates to the smoothness of the response. While $P(x)$ globally approximates the design space, $Z(x)$ creates localized deviations so that the Kriging model interpolates the n sampled data points.

The Kriging predictor is explicitly defined as [26–29, 32]

$$\hat{y}(x) = \hat{\beta} f^T(x) + r^T(x) R^{-1}(Y - F\hat{\beta}), \tag{17}$$

where $\hat{\beta}$ is the matrix of m regression parameters, $f(x)$ is a set of m regression functions, $r(x)$ is a vector that presents the correlation between an unknown point x and the n sample points, F is an overall matrix constructed by evaluating $f(x)$ at each of the n known observations, Y is a vector of n response values, and $R(x^i, x^j)$ is the correlation function between any two of the n sampled data points x^i and x^j. There are several popular correlation functions, such as Gaussian, exponential or cubic spline, etc., which can be effectively applied for the Kriging model. For instance, the Gaussian correlative function can be defined as $R(x^i, x^j) = \exp(-\sum_{k=1}^{p} \theta_k (x_k^i - x_k^j)^2)$.

The key point of Kriging is specifying the process variance σ^2 and the correlation parameters θ_k. These quantities are typically obtained by using the maximum likelihood estimator (MLE) technique. The θ_k^*, MLE of θ_k, used to fit the model are found by maximizing the following log-likelihood function [26–29, 32];

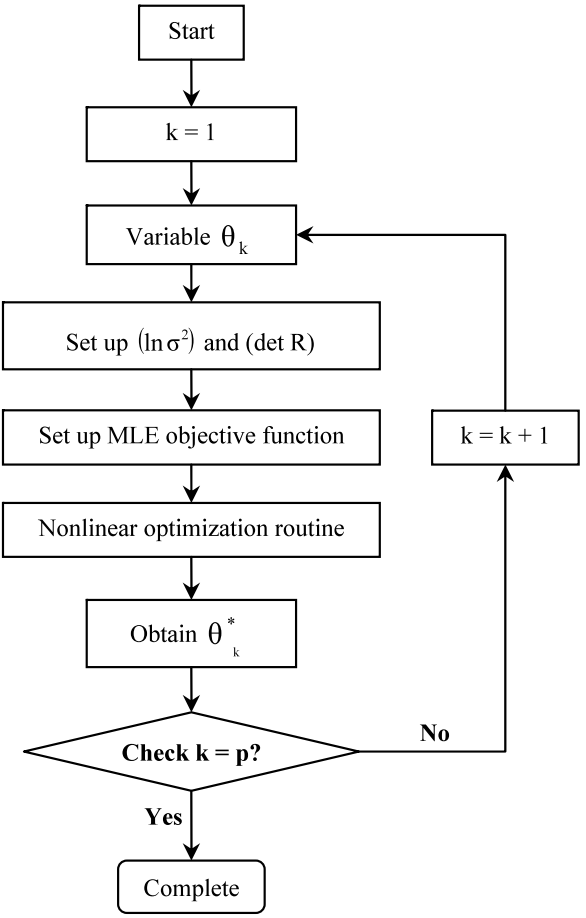
$$\begin{aligned} L(\theta_k) &= -\frac{1}{2}[n \ln(\sigma^2) + \log(\det(R))], \\ \sigma^2 &= \frac{1}{n}(Y - F\hat{\beta})^T R^{-1}(Y - F\hat{\beta}). \end{aligned} \tag{18}$$

The overall performance of the approximate model is evaluated using three accuracy measures. These are the root mean square error (RMSE) which provides a global error measure, the max absolute error (MAE) which is indicative of local deviations, and the average absolute error (AAE) which illustrates the average of local deviations [32],

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{\sum_{i=1}^{n_e} (y(x^i) - \hat{y}(x^i))^2}{n_e}}, \\ \mathrm{MAE} &= \max_i |y(x^i) - \hat{y}(x^i)|, \\ \mathrm{AAE} &= \frac{\sum_{i=1}^{n_e} |y(x^i) - \hat{y}(x^i)|}{n_e}. \end{aligned} \tag{19}$$

Here, n_e is the number of random test points used, $y(x^i)$ denotes the exact function value for the test point x^i and $\hat{y}(x^i)$ is the corresponding predicted value using the approximate model. The smaller these error measures are, the more accurate the approximate model is. In order to realize and verify the set of error measures in the above equation, a large number of additional validation points n_e need to be afforded. However in some cases, taking additional validation points is not possible due to added expense of running additional experiments on the computer simulation; thus, an alternative model assessment which requires no additional points is needed. One such approach is the leave-one-out cross validation [68]. The crossvalidation

Fig. 4 Kriging algorithm



root mean square error (CVRMSE) is defined as

$$\mathrm{CVRMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y(x_i) - \hat{y}'(x_i))^2}{n}}, \tag{20}$$

where $\hat{y}'(x_i)$ is the predicted response using the new approximate model.

In this approach, each sample point used to fit the model is removed one at a time, the new model is rebuilt without that sample point, and the difference between that new model and actual value at the sample point is computed for all of the sample test points. The initial MLE θ_k^* are fixed and applied for all new models. The CVRMSE is employed to investigate the approximate Kriging accuracy of expensive computer simulation.

3.2 Design of Experiments

Up to now, many experimental designs have been discovered and employed in the construction of the approximate metamodel. In practice, many researchers in the area of computer simulation found that the use of space-filling designs, such as Latin hypercubes or orthogonal arrays or uniform designs, etc., are better suited for building approximations of deterministic computer analyses compared to the use of classical experimental designs, such as central composite or Box-Behnken designs, etc. Consequently, the space filling is preferred for computer experimental designs. On the scope of this paper, the authors utilize the Latin hypercube space filling design.

Latin hypercube sampling: The Latin hypercube [26, 31] is a matrix of n_s rows and m columns where n_s is the number of sampled points and m is the number of design variables. Each of the k columns contains the level $1, 2, \ldots, n_s$ randomly permuted and the k columns are matched at random to form the Latin hypercube. Latin hypercube sampling (LHS) offers flexible sizes while ensuring stratified sampling, i.e., each of the input variables is sampled at n_s levels. These designs can have relatively small variance when measuring output variance.

3.3 Genetic Algorithm

Genetic algorithm (GA) [35–40] is a search algorithm based on the mechanics of natural selection and natural genetics, known as Darwinian principle. A traditional GA may be essentially composed of three basic operators:

(i) Reproduction or selection: The reproduction is a process in which individual strings are copied according to their objective function values ("fitness"). Copying strings according to their fitness means that strings with higher value have a higher probability of contributing one or more offspring in the next generation. This operator is very similar to natural selection, survival of the fittest among string creatures. The reproduction may be done in a number of ways, but the easiest one is spinning a typical roulette wheel.

(ii) Crossover: Members of the newly reproduced strings in the mating pool are mated at random and cross over their chromosomes together. For instance, the parents "abcde" and "ABCDE" can create an offspring with a possible chromosome "abcDE". The position between "c" and "D" is determined as crossover point where the chromosome set of the second parent overwrites the chromosome set of the first parent.

(iii) Mutation: The mutation operator helps changing the state of some linking points on the parent's chromosome in order to prevent from loosing potentially useful genetic material (1's or 0's at particular locations).

Generally, a genetic algorithm starts with an initial n-population chosen from a random selection of parameters in the parametric space. Each parameter set presents the individual's chromosome. Each individual is assigned a fitness based on how well each individual's chromosome allows it to perform in its environment. Naturally, only fit individuals are selected for mating, while weak ones die off. Mated parents create their children with chromosome sets are mix of the parent's chromosomes. The

process of mating and children creation is continued so as to create a fitter generation of n children; practically, this is well presented by the increase or decrease of average fitness of the population. The process of reproduction-crossover-mutation is repeated until entire population size is replenished with children. The successive generations are created until very fit individuals are obtained.

3.4 Integrated Multiobjective Optimization Algorithm

In this paper, a general multiobjective optimization algorithm, known as weighted global criterion [39, 47], is utilized. This is a scalar method that combines all objective functions to form a single function U. The most common weighted global criterion for k objectives $f_i(x)$ may be defined as follows:

$$U = \left\{ \sum_{i=1}^{k} [w_i (f_i(x) - f_i^0)]^p \right\}^{1/p}. \tag{21}$$

Here w_i is a vector of weights typically set by the decision maker such that $\sum_{i=1}^{k} w_i = 1$ with $w_i > 0$ and p is an adjusted coefficient which is proportional to the amount of emphasis placed on minimizing the above function with the largest difference between $f_i(x)$ and the ideal point $f_i^0 = \min\{f_i(x)\}$. Practically, the set of ideal points of multiple objectives is unique and explicit for each multiobjective optimization problem. The idea of U was developed from the concept of the Pareto optimal. The Pareto optimal is a compromise solution which is retrieved by minimizing the Euclidian distance $D(x) = \{\sum_{i=1}^{k} [f_i(x) - f_i^0]^2\}^{1/2}$ from the ideal point in the criterion space.

In practice, the major difficulty with multiobjective optimization algorithm is to determine the appropriate weighting factors. The final decision for these factors is normally depends on the experience of the designer; thus, it can not yield even increases in the performance for all disciplines reliably. In order to overcome this difficulty, an automatic design method that determines appropriate weighting factors which improves the design for each discipline evenly is developed. It is shown that the different sets of weighting factors can yield different design results of multiple objectives optimization; these factors, therefore, have to be considered as additional design variables. In the proposed method, the weighting factors are integrated in a new objective function which is defined as follows:

$$\begin{aligned} \min \quad & F_n = \sum_{i=1}^{k} \sum_{j>i}^{k} |\mathrm{loss}_i - \mathrm{loss}_j|, \\ \text{s.t.} \quad & \mathrm{loss}_i = \frac{f_i^0 - f_i(X^{\mathrm{opt}})}{f_i^0}. \end{aligned} \tag{22}$$

The superscript opt shows the optimum point of the multiobjective function U. It is clear that X is considered as a set of design variables of multiobjective function U. w is treated as a set of design variables of the integrated objective function F_n. Practically, the loss_i function indicates the performance loss of each optimized objective

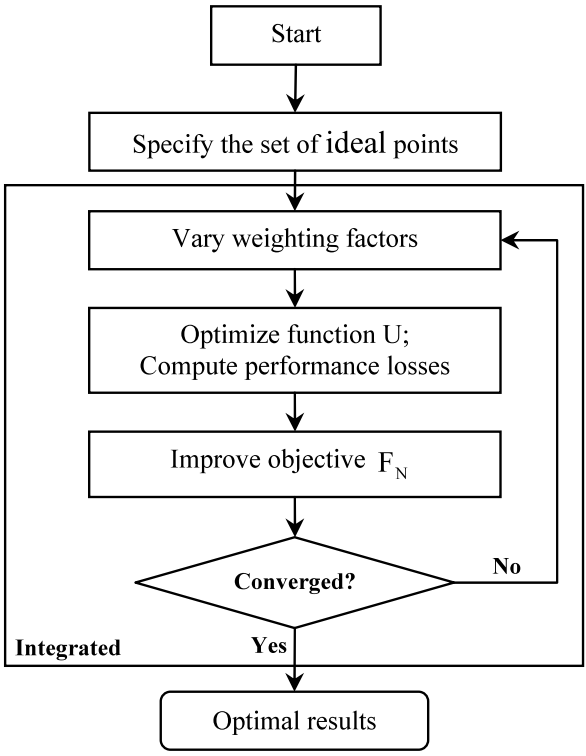


Fig. 5 Design procedure of the weighting factors

in comparison with its ideal point and the F_n objective function states the total mutual differences in the performance loss ratio between all optimal objectives. The set of weighting factors that minimizes the objective function F_n can improve the design evenly at all points and disciplines. The procedure for these weighting factors is summarized in the flow chart as shown in Fig. 5.

The entire process is an integration of the two optimization cycles. Firstly, the weighting factors are arbitrarily and continuously set by the integrated optimizer with the progress of the optimization process. The multiobjective function U is formed in according with each set of these factors. The optimum wing is then designed using the Genetic Algorithm optimizer. After executing the wing optimization, the performance losses of all objectives, which involve the multiobjective function, are computed and used to estimate the function value of F_n to be optimized. The above process will be enhanced again by the genetic algorithm optimizer until the convergent condition is satisfied. In general, the authors simply suggest a reasonable mode to retrieve a unique set of weighting factors relying on non-dominated solution for all objectives. No objective can dominate the others. Therefore, the design system will be improved evenly for all disciplines. However, the final decision in selection of this set of weighting factors for weighted-global-criterion objective function might depends on designer's preference in making trade-off without applying the above integrated algorithm.

4 Wing Design Case Study

A tested eight-variable wing design problem was executed to validate the MDO system. The design variables can be divided into aerodynamic and structure groups. In accordance with the scope of this paper, the wing planform, not the geometric shape of the airfoil, was utilized here owing to its remarkable influence on the aerodynamic performance lift/drag (L/D). The five aerodynamic variables are the break chord (C_B), tip chord (C_T), sweepback angle (Λ), break semi-span (S_B) and semi-span (S_S); note that the wing has 1 m root chord length. On the other hand, the wing structure has to incorporate the lowest weight and sufficient strength. The weight and strength of the wing structure essentially depend on the thickness of thin walls, such as sparwebs or skins, and the cross section of bending supported components like stringers or sparcabs. In this study, only three structural variables are included, involving the upper skin thickness (t_{us}), the lower skin thickness (t_{ls}) and the sparcabs cross-sectional area (A_{sc}). Finally, there are eight design variables in total and the ranges of these variables are summarized in Table 1.

After selecting feasible design variables, the multiple objectives are then adopted to improve the wing performance. Similarly, these objectives can be classified to aerodynamic objectives that are correlative to L/D and structural objectives that are correlative to the weight of the wing. A summary of the design objectives and flight condition is given in Table 2.

The multiobjective function and constraints used for the coupled aerostructural design optimization are defined as

$$\begin{aligned} \min \quad & U = w_1 \times (\text{Weight} - f^0_{struc}) + w_2 \times (f^0_{aero} - L/D), \\ \text{s.t.} \quad & C_L > C_{L_{base}}, \\ & C_D < C_{D_{base}}, \\ & \sigma < [\sigma_y]. \end{aligned} \tag{23}$$

Here, w_i $(i = 1, 2)$ are the weighting factors of a simplified function of the weighted global criterion, C_L is lift coefficient, C_D is drag coefficient, σ is maximum Von-Mises stress, and $[\sigma_y]$ is the yield stress of material. The wing is made

Table 1 Ranges of design variables

Variables	Minimum	Baseline	Maximum	Unit
C_B	0.60	0.75	0.90	m
C_T	0.20	0.35	0.50	m
Λ	30.0	35.0	40.0	deg
S_B	0.40	0.70	1.00	m
S_S	1.80	2.10	2.40	m
t_{us}	0.0015	0.00225	0.0030	m
t_{ls}	0.0015	0.00225	0.0030	m
A_{sc}	0.000153937	0.000430398	0.000706858	M^2

Fig. 6 Wing design variables

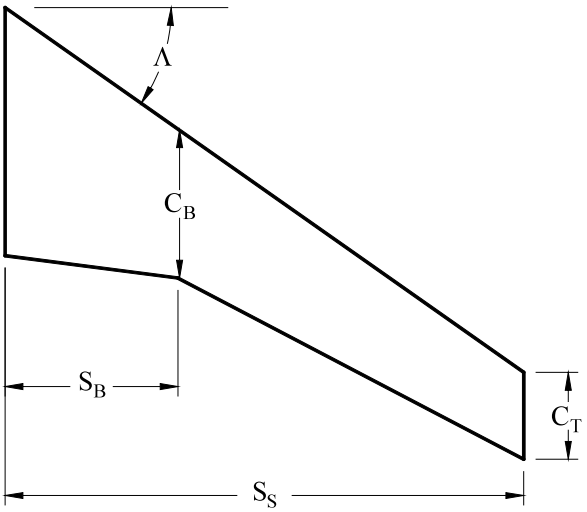
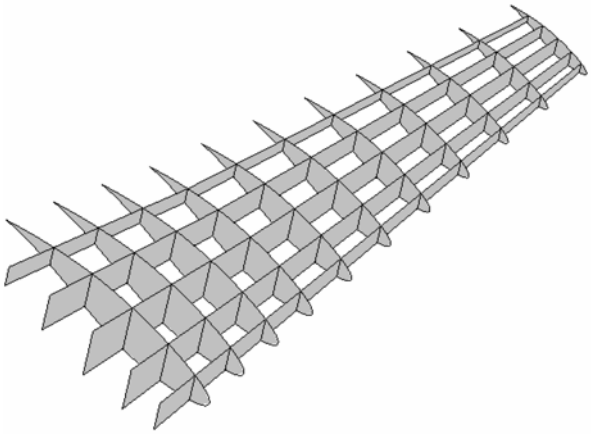


Fig. 7 Wing spars/ribs structure



from aluminum alloy Al 2024-T3. The subscript *base* is utilized to symbolize the baseline calculation. The above constraints ensure that the optimized wing outperforms the baseline wing. For instance, aerodynamic constraints are imposed on the lift and drag to meet the goal that the aerodynamic performance of the designed wing should be at least as good as that of the baseline wing. In addition, f^0_{aero} and f^0_{struc} are in turn the ideal points of the aerodynamic and structural objectives. These are

$$\begin{aligned} & f^0_{aero} = \max(\mathrm{L/D}), \\ \text{s.t.} \quad & C_L > C_{L_{base}}, \\ & C_D < C_{D_{base}}, \\ & \sigma < [\sigma_y], \end{aligned} \tag{24}$$

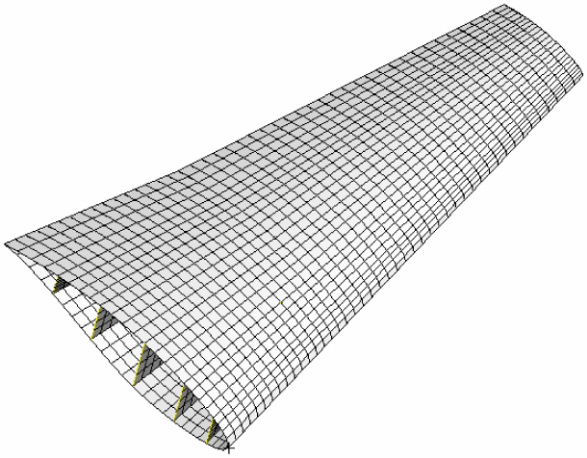


Fig. 8 Wing finite element model

Table 2 Design objectives and flight condition

Condition	Status	Unit
Mach number M	0.84 (cruising)	none
Angle of attack (AOA)	3	deg
Aerodynamic objective	Maximize L/D	none
Structural objective	Minimize (Weight)	kg

and

$$\begin{aligned} & f^0_{struc} = \min(\text{Weight}) \\ \text{s.t.} \quad & C_L > C_{L_{base}} \\ & C_D < C_{D_{base}} \\ & \sigma < [\sigma_y]. \end{aligned} \tag{25}$$

5 MDO Procedures

The MDO system begins with the selection of the design variables, constraints and objective functions. Eight variables are ultimately adopted to enhance the performance of the wing in terms of its aerodynamic and structural performance measures. To reduce the computational time, the Kriging metamodel is employed to approximate the design system. The candidate points for the eight-variable design problem were successfully retrieved using the MATLAB [69] Latin hypercube design function to produce an accurate Kriging approximation at last. The coupled aerodynamic and structural responses are in turn computed through the process of the aeroelastic analysis.

Firstly, the multiblock structured CFD grid of the wing geometry is analyzed via the CFD package coupled with the Spalart-Allmaras viscous model to the lift coef-

Table 3 Sampled Kriging errors

Error quantities	CVRMSE-70 (a)	MINEP (b)	(a)/(b)
Weight	0.237984	31.21563	0.00762
L/D	0.221904	13.79386	0.01609
C_D	0.000450	0.022389	0.02001

ficients C_L, the drag coefficients C_D and the values of L/D. Subsequently, the structural performances, such as the weight of the wing and the Von Mises stress, will be obtained by realizing CSM analyses. A highly stiff structural model of the wing is constructed with five spars distributed along the chord length. Ribs are arranged along the spanwise direction of the semispan length.

The total of 4400 three-dimensional quadrilateral and 440 three-dimensional truss finite elements were generated in the construction of the wing structural model. The appropriate thicknesses of the sparwebs, ribs, skins and the cross-sectional areas of the sparcabs were chosen to imitate the actual structure of the wing.

The loosely coupled static aeroelasticity is analyzed and repeated for all test points; the aerodynamic and structural data are obtained from these analyses. The overall procedure for carrying out the static aeroelastic computations can be divided into several steps: performing CFD computations, interpolating aerodynamic forces onto the structural mesh, realizing CSM computation to obtain the structural deformations, extrapolating these displacements onto the CFD grid, deforming the CFD grid, repeating the above steps for several iterations (commonly from four to nine iterations) using current solution as the starting point for the subsequent steps to reach to the convergent shape. For example, the aeroelastic results of the baseline wing are shown in Fig. 9 and Fig. 10. In this case, the tip incidence is reduced by approximately 0.859 degrees.

Based on the resulting data, the approximate Kriging model for each aerodynamic and structural performance are constructed, verified and prepared for the optimization process. The ideal points of the aerodynamic and structural objectives have to be determined; the optimal solutions are specified. In addition, the Kriging can be updated based on the modification of test points; the optimization cycle is then repeated until the desired solutions are accepted. In summary, the overall optimization process is presented in Fig. 11.

For instance, CVRMSE-70 is the sampled cross-validation root mean square error of the Kriging method using 70 experimental points and MINEP of a quantity is the minimum value of that quantity in the set of experimental points. The accuracy of Kriging in this case is acceptable. Naturally, increasing sample points will lead to the enhancement of Kriging accuracy. Nevertheless, the computational cost also rises significantly due to the expense of computer simulation. Therefore, the selection of the number of test points normally depends on the accuracy of approximate model, computational expense and designer's preference.

In general, the Kriging method proves to be a good metamodel due to its cheapness and efficiency. To determine the global optimum of an eight-variable design problem, the global optimization algorithms, e.g., genetic algorithm or simulated annealing, may request at least 4000 function evaluations while the Kriging requires around

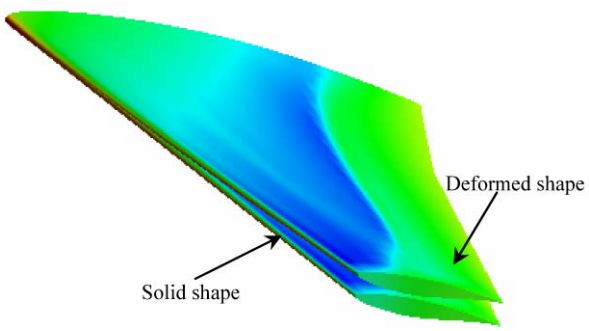


Fig. 9 Static aeroelasticity

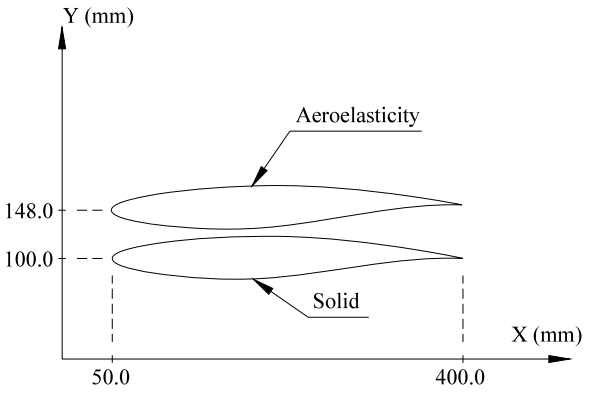


Fig. 10 Deflection of the wing tip

100 function evaluations (or more). Assume that one function evaluation obtained from fluid/structure interface simply takes around 18 hours, the disparity in computational cost between approximate Kriging and global optimization is extremely big. The employment of Kriging metamodel is basically reasonable since the trade-off between the computational expense and the accuracy of optimal solution is acceptable. According to our experience, apart from ERBF, SVR and MLS, KM is also one of the state-of-the-art metamodelings in the world of optimization at this time.

The preferred optimum results of the wing, which are done by realizing the aforementioned suggested multiobjective optimization algorithm and Kriging model, are summarized in Table 4. The micro GA with population of 5 individuals, uniform crossover with probability of 0.05 and jump mutation with probability of 0.02 were utilized for the design problem.

The optimized wing shows better performances than the baseline wing; the aerodynamic performance is improved and the structural weight is decreased. The value of L/D is reduced by an amount of 7.438% and the wing weight is decreased by an amount of 7.563% in comparison with the aerodynamic and structural ideal points. The aerodynamic and structural performances are improved by nearly an equal amount. The multidisciplinary aero-structural design is desirable and practical as it enhances the aerodynamic and structural performances simultaneously. Moreover, the values 0.479 and 0.521 of the optimized aerodynamic and structural weighting factors show the high efficiency of the proposed multiobjective optimization algo-

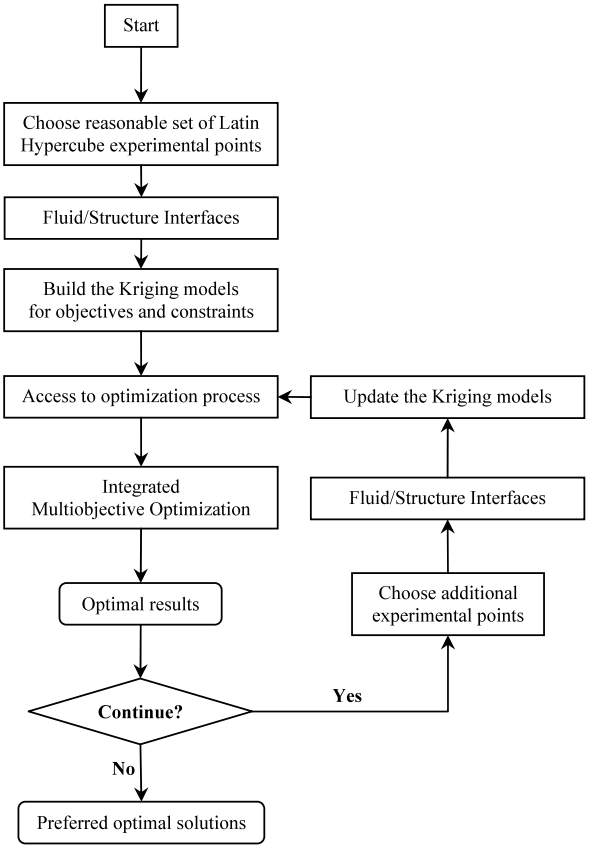


Fig. 11 Overall optimization process

Table 4 MDO results

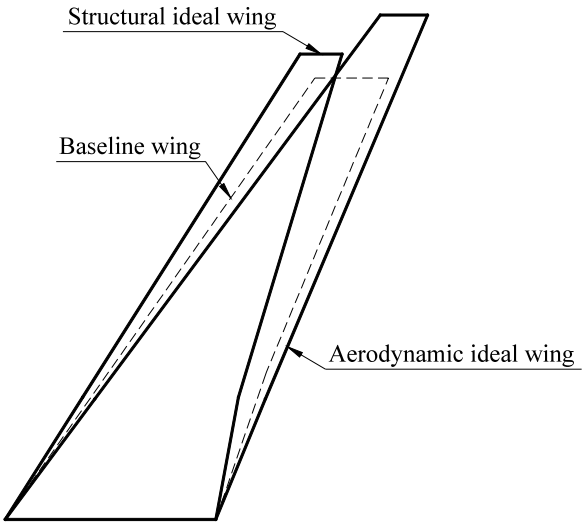
Case	Values	Baseline	Aero-ideal	Struc-ideal	Optimum	Unit
Cruising	C_L	0.6366	0.6362	0.6357	0.6363	none
$M = 0.84$	C_D	0.0315	0.0255	0.0315	0.0276	none
AOA $= 3°$	L/D	20.223	24.940	20.179	23.085	none
	Weight	56.044	56.967	29.658	31.901	kg

rithm. It is shown that the algorithm can be applied effectively for any optimization and design problem.

6 Conclusions

This research is motivated by our interest in developing and improving computational capability of MDO system. Considerable MDO work was successfully performed for a tested wing to validate several suggested algorithms that can be easily applied for

Fig. 12 Ideal points



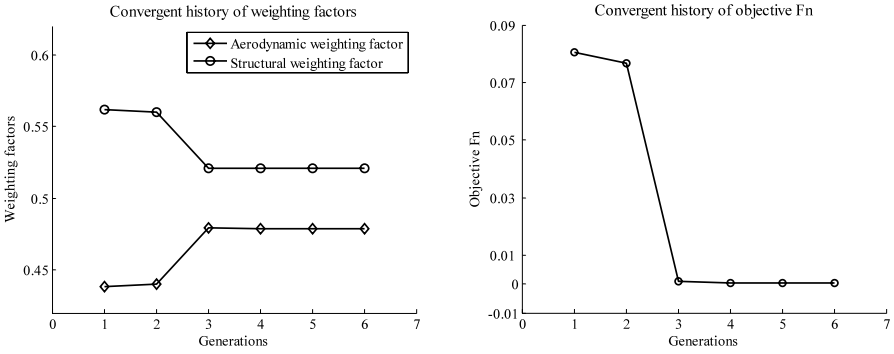


Fig. 13 Convergent history

more complex and practical problems. The high-fidelity structural analysis code was coupled with the commercial CFD code and robust fluid/structure coupling algorithm to realize the aeroelastic analyses. The aerodynamic and structural meshes were well-managed by using a robust grid deformation algorithm and a CSM mesh generator. The design system was subsequently approximated by utilizing a robust Kriging interpolative model. An efficient multiobjective optimization algorithm was also investigated and can be easily extended to a wide range of optimization problems. The use of equal weighting factors does not yield even increases of performances for all disciplines. Thus, an automatic method that determines appropriate weighting factors which improves the design for each disciplines evenly was proposed. Through the use of this method, the aerodynamic and structural performances can be improved evenly from its ideal points which are the optimum points of single discipline. Therefore, the Multidisciplinary aerostructural design is desirable and practical.

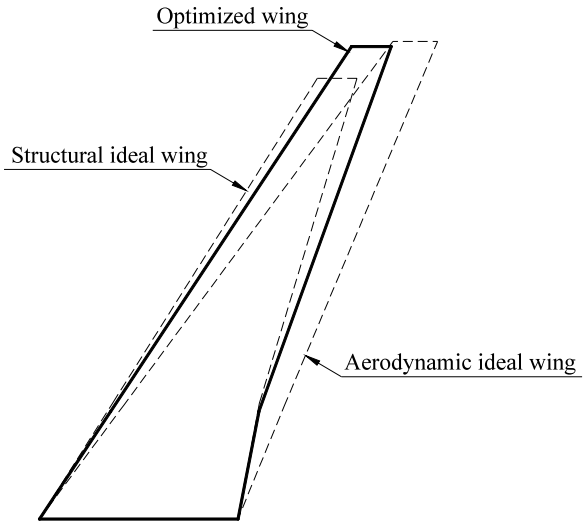


Fig. 14 Optimum wing

References

1. Sobieski, J.S., Haftka, R.T.: Multidisciplinary aerospace design optimization: survey of recent developments. AIAA J., AIAA-96-0711 (1996)
2. Wakayama, S.R.: Lifting surface design using multidisciplinary optimization. Ph.D. Thesis, Stanford University (1997)
3. Walsh, J.L., Townsend, J.C., Salas, A.O., Samareh, J.A., Mukhopadhyay, V., Barthelemy, J.-F.: Multidisciplinary high-fidelity analysis and optimization of aerospace vehicles. AIAA J., AIAA-2000-0418 (2000)
4. Martins, R.R.A.: A coupled-adjoint method for high-fidelity aero-structural optimization. Ph.D. Thesis, Stanford University (2002)
5. Venkataraman, S., Haftka, R.T.: Structural optimization complexity: what has Moore's law done for us. J. Struct. Multidiscip. Optim. **28**, 375–387 (2004)
6. Kim, Y., Kim, J., Jeon, Y., Bang, J., Lee, D.-H., Kim, Y., Park, C.W.: Multidisciplinary aerodynamic-structural design optimization of supersonic fighter wing using response surface methodology. AIAA J., AIAA-2002-0322 (2002)
7. Giunta, A.A.: Aircraft multidisciplinary design optimization using design of experiments theory and response surface modeling methods. Ph.D. Thesis, University of Virginia (1997)
8. Giunta, A.A., Balabanov, V., Haim, D., Grossman, B., Mason, W.H., Watson, L.T., Haftka, R.T.: Wing design for a high-speed civil transport using a design of experiments methodology. AIAA J., AIAA-96-4001 (1996)
9. Joaquim, R.R., Alonso, J.J., Reuther, J.: Aero-Structural Wing Design Optimization using high-fidelity sensitivity analysis. In: Proceeding to CEAS Conference on Multidisciplinary Aircraft Design Optimization, Germany. Confederation of European Societies (2001)
10. Chittick, I.R., Martins, J.R.R.A.: Aero-structural optimization using adjoint coupled post-optimality sensitivities. J. Struct. Multidiscip. Optim. DOI 10.1007/s00158-007-0200-9 (2007)
11. Gumbert, C.R., Newman, P.A.: High-fidelity computational optimization for 3-D flexible wings. J. Optim. Eng. **6**, 117–156 (2005)
12. Kumano, T., Jeong, S., Obayashi, S., Ito, Y., Hatanaka, K., Morino, H.: Multidisciplinary design optimization of wing shape with nacelle and pylon. In: European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006, TU Delft, The Netherlands (2006)
13. Weck, O.D., Agte, J., Sobieski, J.S., Arendsen, P., Morris, A., Spieck, M.: State-of-the-art and future trends in multidisciplinary design optimization. In: 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Hawaii, USA. AIAA-2007-1905 (2007)

14. Martins, J.R.R.A., Marriage, C.: An objective-oriented framework for multidisciplinary design optimization. In: 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Hawaii, USA. AIAA-2007-1906 (2007)
15. Kamakoti, R., Shyy, W.: Fluid-structure interaction for aeroelastic applications. Prog. Aerospace Sci. **40**, 535–558 (2005)
16. Guruswamy, G.P.: A review of numerical fluids/structures interface methods for computations using high-fidelity equations. J. Comput. Struct. **80**, 31–41 (2001)
17. Hounjet, M.H.L., Meijer, J.J.: Evaluation of elastomechanical and aerodynamic data transfer methods for non-planar configurations in computational aeroelastic analysis. National Aerospace Laboratory NRL, NLR-TP-95690 U (1995)
18. Bhadra, S., Ganguli, R.: Aeroelastic optimization of a helicopter rotor using orthogonal array-based metamodels. AIAA J. **44**(9), 1941–1951 (2006)
19. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1996)
20. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, New Jersey (1999)
21. Hagan, M.T., Demuth, H.B., Beale, M.: Neural Network Design. Massachusetts (1996)
22. Friedman, J.H.: Multivariate adaptive regression splines, invited paper. Ann. Stat. **19**(1), 1–67 (1991)
23. Turner, C.J., Crawford, R.H., Campbell, M.I.: Multidimensional sequential sampling for NURBs-based metamodel development. J. Eng. Comput. **23**, 155–174 (2007)
24. Mullur, A.A., Messac, A.: Extended radial basis functions: More flexible and effective metamodeling. AIAA J. **43**(6), 1306–1315 (2005)
25. Mullur, A.A., Messac, A.: Metamodeling using extended radial basis functions: A comparative approach. J. Eng. Comput. **21**, 203–217 (2006)
26. Koehler, J.R., Owen, A.B.: Computer Experiments. Handbook of Statistics 13: Design and Analysis of Experiments. Elsevier, Amsterdam (1996)
27. Giunta, A.A., Watson, L.T.: A comparison of approximation modeling techniques: Polynomial versus interpolating models. AIAA J., AIAA-98-4758 (1998)
28. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. J. Stat. Sci. **4**(4), 409–423 (1989)
29. Jeong, S., Murayama, M., Yamamoto, K.: Efficient optimization design method using Kriging model. AIAA J., AIAA-2004-118 (2004)
30. Simpson, T.W., Dennis, L., Chen, W.: Sampling strategies for computer experiments: design and analysis. Int. J. Reliab. Appl. **23**(2), 209–240 (2001)
31. Simpson, T.W., Booker, A.J., Ghosh, D., Giunta, A.A., Koch, P.N., Yang, R.-J.: Approximation methods in multidisciplinary analysis and optimization: A panel discussion. J. Struct. Multidiscip. Optim. **27**, 302–313 (2004)
32. Martin, J.D., Simpson, T.W.: Use of Kriging models to approximate deterministic computer models. AIAA J. **43**(4), 853–863 (2005)
33. Clarke, S.M., Griebsch, J.H., Simpson, T.W.: Analysis of support vector regression for approximation of complex engineering analyses. ASME J. **127**, 1077–1087 (2005)
34. Maisuradze, G.G., Thompson, D.L.: Interpolating moving least-squares methods for fitting potential energy surfaces: illustrative approaches and applications. J. Phys. Chem. A **107**(37), 7118–7124 (2003)
35. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman Inc, Cambridge (1989)
36. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin (1996)
37. Yang, G., Reinstein, L.E., Pai, S., Xu, Z.: A new genetic algorithm technique in optimization of permanent prostate implants. J. Med. Phys. **25**(12), 2308–2315 (1998)
38. Carroll, D.L.: Chemical laser modeling with genetic algorithms. AIAA J. **34**(2), 338–346 (1996)
39. Arora, J.S.: Introduction to optimum design. Elsevier Academic, San Diego (2004)
40. Arora, J.S., Elwakeil, O.A., Chahande, A.I., Hsieh, C.C.: Global optimization methods for engineering applications: A review. J. Struct. Optim. **9**, 137–159 (1995)
41. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. J. Sci. **220**(4598), 671–680 (1983)
42. Goffe, W.L., Ferrier, G.D., Rogers, J.: Global optimization of statistical functions with simulated annealing. J. Econom. **60**(1/2), 65–100 (1993)
43. Corana, A., Marchesi, M., Martini, C., Ridella, S.: Minimizing multimodal functions of continuous variables with the 'Simulated Annealing' algorithm. ACM Trans. Math. Softw. **13**(3), 262–280 (1987)

44. Yao, X.: Simulated annealing with extended neighbourhood. Int. J. Comput. Math. **40**, 169–189 (1991)
45. Coello Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York (2007)
46. Deb, K.: Current trends in evolutionary multi-objective optimization. Int. J. Simul. Multidiscip. Des. Optim. **1**, 1–8 (2007)
47. Marler, R.T.: A study of multi-objective optimization methods for engineering applications. Ph.D. Thesis, University of Iowa (2005)
48. FLUENT INC: Fluent User's Manual. Fluent Inc, New Hampshire (2005)
49. Blom, F.J.: Considerations on the spring analogy. Int. J. Numer. Methods Fluids **32**, 647–668 (2000)
50. Tsai, H.M., Wong, A.S.F., Cai, J., Zhu, Y., Liu, F.: Unsteady flow calculations with a parallel multi-block moving mesh algorithm. AIAA J. **39**(6), 1021–1029 (2000)
51. Dubuc, L., Cantariti, F., Woodgate, M., Gribben, B., Badcock, K.J., Richards, B.E.: A grid deformation technique for unsteady flow computations. Int. J. Numer. Methods Fluids **32**, 285–311 (2000)
52. Spekreijse, S.P., Prananta, B.B., Kok, J.C.: A simple, robust and fast algorithm to compute deformations of multi-block structured grids. National Aerospace Laboratory NLR, NLR-TP-2002-105 (2002)
53. Thompson, J.F., Soni, B.K., Weatherill, N.P.: Handbook of Grid Generation. CRC Press LLC, Boca Raton (1999)
54. Sadeghi, M., Liu, F., Lai, K.L., Tsai, H.M.: Application of three-dimensional interfaces for data transfer in aeroelastic computations. AIAA J., AIAA-2004-5376 (2004)
55. Dowell, E.H., Hall, K.C.: Modeling of fluid-structure interaction. J. Fluid Mech. **33**, 445–490 (2001)
56. Hirsch, C.: Numerical Computation of Internal and External Flows. Butterworth-Heinemann, Oxford (2007)
57. Blazek, J.: Computational Fluid Dynamics: Principles and Applications. Elsevier Science Ltd, Oxford (2001)
58. Chung, T.J.: Computational Fluid Dynamics. Cambridge University Press, Cambridge (2002)
59. Ferziger, J.H., Peric, M.: Computational Methods for Fluid Dynamics. Springer, Berlin (2002)
60. Anderson, J.D.: Computational Fluid Dynamics: The Basics with Applications. McGraw-Hill, Columbus (1995)
61. Pointwise: Gridgen User's Manual. Pointwise Inc, Texas, USA (2005)
62. Zienkiewicz, O.C., Taylor, L.R.: The Finite Element Method, 5th edn. Butterworth-Heinemann, Oxford (2000)
63. Bathe, K.-J.: Finite Element Procedures. Prentice-Hall, Englewood Cliffs (1996)
64. Smith, I.M., Griffiths, D.V.: Programming the Finite Element Method. Wiley, Chichester (2004)
65. Reddy, J.N.: An introduction to the Finite Element Method, 3rd edn. McGraw-Hill, New York (2006)
66. Liu, G.R., Quek, S.S.: The Finite Element Method—A Practical Course. Butterworth-Heinemann, Oxford (2003)
67. Ribo, R., Pasenau, M.D.R., Escolano, E., Ronda, J.S.P., Sans, A.C., Gonzalez, L.F.: GiD User's Manual. CIMNE, Barcelona, Spain (2007)
68. Mitchell, T.J., Morris, M.D.: Bayesian design and analysis of computer experiments: Two examples. J. Stat. Sinica **2**, 359–379 (1992)
69. The Mathworks: Matlab User'S Manual. The MathWorks Inc, Massachusetts, USA (2007)