AIAA JOURNAL
Vol. 48, No. 8, August 2010

Geometry Parameterization and Computational Mesh Deformation by Physics-Based Direct Manipulation Approaches

Wataru Yamazaki,* Sylvain Mouton,† and Gérald Carrier†
ONERA, 92190 Meudon, France

DOI: 10.2514/1.J050255

Consistent geometry parameterization of different models used in multidisciplinary optimization is one of the major issues for efficient design optimization. In this research, physics-based free-form deformation and radial-basis-function-based deformation approaches that allow a direct manipulation of an arbitrary object are proposed for efficient geometry manipulation and parameterization in the optimization processes. The developed methods can deform any computational mesh mapped on the deformable geometry at once. The methods have several advantages for geometry parameterization, such as easy handling of complex objects, direct manipulation, local controllability, and intuitive deformation as real elastic objects. In addition, they offer a straightforward solution for constructing a hierarchical/adaptive parameterization that can benefit the optimization algorithmic performances. Application to an aerodynamic design optimization showed that the methods were effective for efficient design optimization.

Nomenclature

$\mathbf{B}$	=	blending matrix
$\mathbf{B}^+$	=	pseudoinverse matrix of $\mathbf{B}$
$\mathbf{b}$	=	blending vector
$\mathbf{D}$	=	constitutive matrix
E	=	modulus of elasticity
$\mathbf{K}$	=	stiffness matrix
n	=	number of control points/centers
$\mathbf{P}$	=	original coordinates of control points/centers
q	=	number of pilot points
$\mathbf{U}$	=	local curvilinear coordinates, $(s \quad t \quad u)$
w	=	deformation energy
$\mathbf{x}$	=	original coordinates of a node point, $(x_1 \quad x_2 \quad x_3)$
β, γ	=	unknown coefficients of radial basis function interpolation
$\Delta\mathbf{P}$	=	displacement of control points/centers
$\Delta\mathbf{X}$	=	displacement of pilot points
$\Delta\mathbf{x}$	=	displacement of a node point, $(\delta x_1 \quad \delta x_2 \quad \delta x_3)$
ε	=	strain matrix
λ, μ	=	Lamé's constants
υ	=	Poisson ratio
ϕ	=	radial basis function
ω	=	damping function

I. Introduction

AERODYNAMIC design optimization is a discipline at the crossroads of the latest mathematical, numerical, and fluid dynamics methods. It facilitates the design process by automating both the performance analysis and the search for optimum. Subsequently, the recent maturity of computational fluid dynamics (CFD) and progress of computer performance, the aerodynamic design optimization has attracted much attention. Nowadays, it has been widely explored in academic research groups and industrial concerns. It has contributed to the design of numerous recent aerospace products, such as launch vehicles, airplanes, helicopters and turbomachinery. The design optimization procedures rely on several tools and methods, among which geometry parameterization, computational mesh regeneration or deformation, numerical simulation and optimization algorithm. In the past years, both optimization algorithms and CFD methods have reached a high level of maturity in terms of efficiency and robustness and are now routinely used in the aerospace industry [1–3]. At the same time, two major bottlenecks for the more systematic use of aerodynamic design optimization in the industrial design context have been pointed out: the efficient parameterization of complex objects and the deformation of the computational meshes around those complex objects.

The geometry parameterization is the crucial part for a flexible design and usually impacts both the speed and the result of optimizations. Therefore, shape parameterization methods should enable a robust, flexible, and detailed definition for an arbitrary object. Many geometry parameterization methods have been developed and applied in aerodynamic design optimizations, such as the Hicks–Henne function [4], parametric-section airfoil parameterization [5], Bézier curve [6,7], nonuniform rational B-spline (NURBS) [8], analytical method [9], CAD-based approach [10–12], class/shape function transformation method [13,14], free-form deformation (FFD) [15–20], radial basis function (RBF) approach [21,22], mesh point method [1,23] and so on. Because of the recent increase of interest in parameterization methods, the effect of shape parameterization methods on the results of design optimizations has been investigated vigorously [24–28]. In these parameterization methods, the FFD and RBF approaches have recently attracted attention because they can easily handle any complex three-dimensional objects.

The mesh regeneration or deformation after the geometry definition has been set up is another crucial step in the optimization procedure. For efficient aerodynamic design optimizations, mesh deformation techniques are preferred to regeneration techniques in order to avoid introducing numerical error due to the change in discretization and also because of the lower computational cost. Moreover, in cases of gradient-based optimizations using an adjoint formulation, mesh deformation methods that enable the calculation of mesh sensitivity are preferable. Several approaches tolerating large geometry displacements have been proposed to increase the variety of attainable designs without losing robustness [29–33]. However, the quality of the resulting deformed meshes is still a crucial issue affecting the reliability of the optimization, because the

Presented as Paper 2008-5953 at the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, Canada, 10–12 September 2008; received 28 September 2009; revision received 5 March 2010; accepted for publication 31 March 2010.

*Postdoctoral Researcher, Applied Aerodynamics Department, 8, rue des Vertugadins.

†Research Scientist, Applied Aerodynamics Department, 5 Boulevard Painlevé, 59045 Lille.

variation of mesh quality affects the accuracy of the flow solution. Therefore, the preservation of the mesh quality is essential for accurate and reliable aerodynamic design optimizations.

One of the advantages of the RBF approach over FFD is that it can be easily used for both geometry parameterization and computational mesh deformation. In [22], the standard RBF approach was successfully applied to a design optimization of a 3-D configuration as both geometry parameterization method and computational mesh deformation strategy. However, the mesh quality deformed by the standard RBF approach just relies on the smoothness of the radial basis function. Therefore, new approaches that can be used for both efficient geometry parameterization and computational mesh deformation while preserving the mesh quality are desirable.

In this research, FFD and RBF parameterization approaches are extended to have better capabilities for both geometry parameterization and computational mesh deformation. In detail, physics-based direct-manipulation FFD/RBF methods are developed that are grounded on an elastic analogy. These methods enable both intuitive geometry parameterization and computational mesh deformation while preserving the mesh quality.

II. Physics-Based Direct-Manipulation FFD/RBF

A. Free-Form Deformation

FFD is a versatile approach that expresses object deformations only by the displacement of a parallelepiped volume domain surrounding an object (we call the volume domain *control box* or *control lattice*), instead of the displacement of the object surface itself. Design variables in FFD are the node points' coordinates of the control box. Therefore, it can be easily applied to an arbitrary complex object. In this context, FFD will be applied to deform any surface or three-dimensional volume meshes for CFD, computational structural mechanics, and other numerical disciplines such as acoustics or heat transfer.

The original Bézier-based FFD method is defined in terms of a tensor product trivariate Bernstein polynomial [34]. The displacement $\Delta\mathbf{x}$ of any node point $\mathbf{x}(s,t,u)$ inside the control box is defined as follows:

$$\mathbf{x}(s,t,u)+\Delta\mathbf{x}(s,t,u)=\sum_{i=0}^{l_1}\sum_{j=0}^{l_2}\sum_{k=0}^{l_3}[B^i_{l_1}(s)B^j_{l_2}(t)B^k_{l_3}(u)]\cdot[\mathbf{P}_{i,j,k}+\Delta\mathbf{P}_{i,j,k}] \tag{1}$$

where $\mathbf{P}_{i,j,k}$ and $\Delta\mathbf{P}_{i,j,k}$ are the line matrices representing the original coordinates and the displacement of the node point (i,j,k) of the control box, respectively; (s,t,u) are the local curvilinear coordinates mapped into the control box, also called lattice coordinates; and $B^i_l(s)$ is the ith Bernstein polynomial of the degree l, which is defined as follows:

$$B^i_l(s)=\frac{(l)!}{(i)!(l-i)!}s^i(1-s)^{l-i} \tag{2}$$

In the matrix form, Eq. (1) can be written as follows:

$$\begin{pmatrix}\delta x_1\\ \delta x_2\\ \delta x_3\end{pmatrix}^T=(B_{0,0,0}\quad\cdots\quad B_{l_1,l_2,l_3})\times\begin{pmatrix}\Delta P_{0,0,0,1} & \Delta P_{0,0,0,2} & \Delta P_{0,0,0,3}\\ \vdots & \vdots & \vdots\\ \Delta P_{l_1,l_2,l_3,1} & \Delta P_{l_1,l_2,l_3,2} & \Delta P_{l_1,l_2,l_3,3}\end{pmatrix}$$
$$\Delta\mathbf{x}=\mathbf{b}(s,t,u)\cdot\Delta\mathbf{P} \tag{3}$$

where

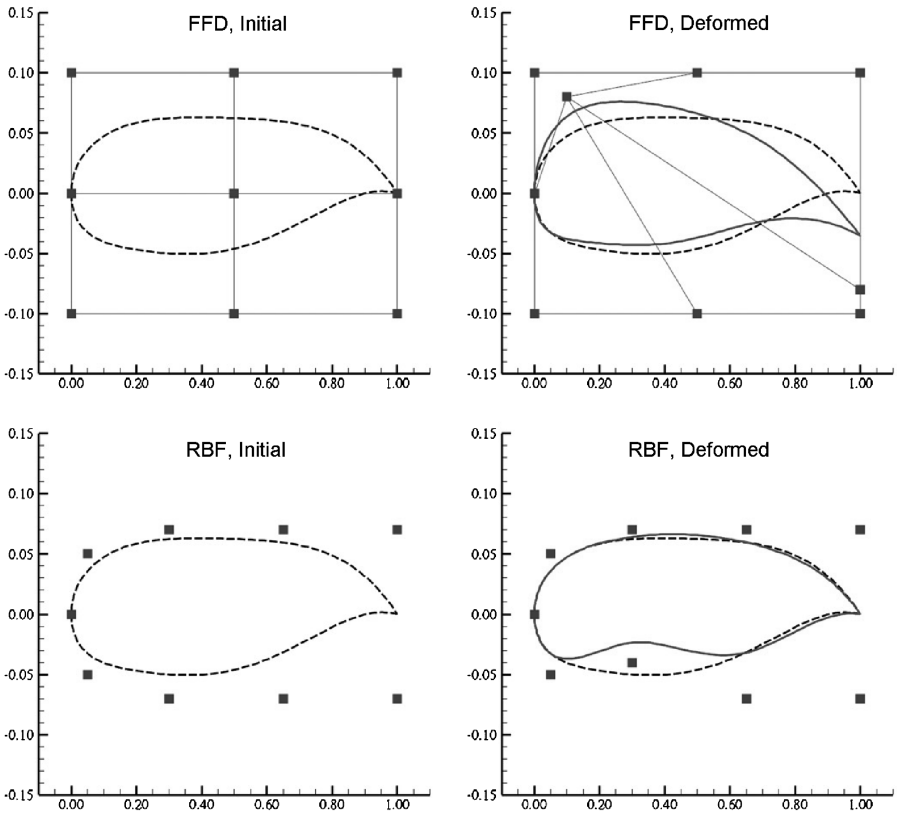

Fig. 1 Conventional FFD/RBF for a 2-D airfoil; upper: FFD, lower: RBF, left: initial set of control box/centers, right: deformed, broken line: initial airfoil, and solid line: deformed airfoil.

$$B_{i,j,k} = B^{i}_{l_1}(s) B^{j}_{l_2}(t) B^{k}_{l_3}(u) \tag{4}$$

Therefore, the FFD process is implemented by the following three steps. First, the local curvilinear coordinates (s, t, u) are determined for each mesh points lying within the control box and the blending vector $\mathbf{b}(s, t, u)$ is computed. The search for the local curvilinear coordinates is implemented by applying Newton's method. Second, the control box is deformed based on design variables that express $\Delta\mathbf{P}$. Finally, the new location of each mesh point is calculated by Eq. (3). In Fig. 1, a simple application of the FFD to a 2-D airfoil is shown. This standard FFD method has the following specific merits:

1) It can be used for an arbitrary complex object.

2) The deformed objects can be fairly general and continuous.

3) The differentiability and degree-elevation property of the Bézier curve [6] are preserved.

4) The displacement of arbitrary points within the control box (e.g., for CFD meshes, volume grid points around the deformed object) is naturally included in the procedure.

5) B-spline or NURBS-based FFD can be formulated easily by using the B-spline/NURBS basis functions instead of the Bernstein polynomials.

The main advantage of the B-spline and NURBS basis functions compared to the Bézier function is the locality of the deformation induced by the displacement of a control point. The B-spline basis function is given as follows:

$$\begin{aligned} B^{i,0}(s) &= \begin{cases} 1 & \text{if } s_i \le s < s_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B^{i,p}(s) &= \frac{s - s_i}{s_{i+p} - s_i} B^{i,p-1}(s) + \frac{s_{i+p+1} - s}{s_{i+p+1} - s_{i+1}} B^{i+1,p-1}(s) \end{aligned} \tag{5}$$

where p is the degree and s_i is the component of a knot vector.

B. Radial-Basis-Function-Based Deformation

The parameterization method based on RBF is called the domain element method [22], and it has properties similar to the FFD method. The parameterization based on RBF is represented by the displacements of centers that are set around an object. The formula of the interpolation by RBF at a node point $\mathbf{x}$ is expressed as follows:

$$f_{(\mathbf{x})} = \sum_{i=1}^{n} \beta_i \phi_{(\|\mathbf{x} - \mathbf{P}_i\|)} + [\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3] \tag{6}$$

where $\mathbf{P}_i$ is the original coordinates of a center and n is the number of centers. ϕ is a radial basis function. In this research, the following Wendland's radial basis function [21] is used because of the best combination of deformation quality and matrix conditioning:

$$\phi_{(\|\mathbf{r}\|)} = \begin{cases} (1 - \alpha\|\mathbf{r}\|)^4 (4\alpha\|\mathbf{r}\| + 1) & (\|\mathbf{r}\| \le 1/\alpha) \\ 0 & \text{else} \end{cases} \tag{7}$$

where α is a positive constant that determines the deformable range by the displacement of a center. In our approach, α is empirically determined to take the value of 0.2 at the location of each second nearest center, and β_i and γ_i are the unknown coefficients for the interpolation problem. The first term on the right-hand side (RHS) of Eq. (6) is the RBF term that expresses local deformations around the centers. On the other hand, the second term on the RHS is the linear polynomial term that is used to express a simple mode of deformation as parallel shift and/or rotation. The Eq. (6) has to be satisfied at all centers with given displacements as follows:

$$\begin{aligned} \mathbf{P}_i + \Delta\mathbf{P}_i = (\,1 \quad P_{i1} \quad P_{i2} \quad P_{i3} \quad \phi_{\|\mathbf{P}_i - \mathbf{P}_1\|} \quad \cdots \quad \phi_{\|\mathbf{P}_i - \mathbf{P}_n\|}\,) \\ \times \begin{pmatrix} \gamma_{01} & \gamma_{02} & \gamma_{03} \\ \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \\ \beta_{11} & \beta_{12} & \beta_{13} \\ \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \beta_{n3} \end{pmatrix} \qquad (i = 1, \cdots, n) \end{aligned} \tag{8}$$

To complete the system, the following side conditions are added:

$$\sum_{i=1}^{n} \beta_{ij} = 0, \qquad \sum_{i=1}^{n} \beta_{ij} \mathbf{P}_{ik} = 0 \qquad (j, k = 1, 2, 3) \tag{9}$$

These conditions enable to recover simple modes of deformation as parallel shift/rotation. Combining Eqs. (8) and (9),

$$\begin{aligned} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{P}_1 + \Delta\mathbf{P}_1 \\ \vdots \\ \mathbf{P}_n + \Delta\mathbf{P}_n \end{pmatrix} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & P_{11} & \cdots & P_{n1} \\ 0 & 0 & 0 & 0 & P_{12} & \cdots & P_{n2} \\ 0 & 0 & 0 & 0 & P_{13} & \cdots & P_{n3} \\ 1 & P_{11} & P_{12} & P_{13} & \phi_{\|\mathbf{P}_1 - \mathbf{P}_1\|} & \cdots & \phi_{\|\mathbf{P}_1 - \mathbf{P}_n\|} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & P_{n1} & P_{n2} & P_{n3} & \phi_{\|\mathbf{P}_n - \mathbf{P}_1\|} & \cdots & \phi_{\|\mathbf{P}_n - \mathbf{P}_n\|} \end{pmatrix} \\ &\times \begin{pmatrix} \gamma_{01} & \gamma_{02} & \gamma_{03} \\ \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \\ \beta_{11} & \beta_{12} & \beta_{13} \\ \vdots & \vdots & \vdots \\ \beta_{n1} & \beta_{n2} & \beta_{n3} \end{pmatrix} \end{aligned} \tag{10}$$

Equation (10) can be expressed as follows in the matrix form:

$$\mathbf{P}_{rbf} + \Delta\mathbf{P}_{rbf} = \mathbf{C}\mathbf{a}_{xyz} \tag{11}$$

The matrix $\mathbf{C}$ can be expressed as follows in block matrix form:

$$\mathbf{C} = \begin{bmatrix} 0 & \bar{\mathbf{P}} \\ \bar{\mathbf{P}}^T & \mathbf{M} \end{bmatrix} \tag{12}$$

The solvability of Eq. (11) mainly depends on the matrix conditioning of $\mathbf{M}$. Finally, the unknown matrix $\mathbf{a}_{xyz}$ is determined as follows with the block matrix form:

$$\begin{aligned} \mathbf{a}_{xyz} &= \mathbf{H} \cdot (\mathbf{P} + \Delta\mathbf{P}) \\ \mathbf{H} &= \begin{bmatrix} (\bar{\mathbf{P}}\mathbf{M}^{-1}\bar{\mathbf{P}}^T)^{-1}\bar{\mathbf{P}}\mathbf{M}^{-1} \\ \mathbf{M}^{-1} - \mathbf{M}^{-1}\bar{\mathbf{P}}^T(\bar{\mathbf{P}}\mathbf{M}^{-1}\bar{\mathbf{P}}^T)^{-1}\bar{\mathbf{P}}\mathbf{M}^{-1} \end{bmatrix} \end{aligned} \tag{13}$$

Then the interpolation at a node point $\mathbf{x}$ is calculated as follows:

$$\begin{aligned} \mathbf{x} + \Delta\mathbf{x} &= \mathbf{A}\mathbf{a}_{xyz} = \mathbf{A}\mathbf{H}(\mathbf{P} + \Delta\mathbf{P}) \\ \mathbf{A}_{(\mathbf{x})} &= (\,1 \quad x_1 \quad x_2 \quad x_3 \quad \phi_{\|\mathbf{x} - \mathbf{P}_1\|} \quad \cdots \quad \phi_{\|\mathbf{x} - \mathbf{P}_n\|}\,) \end{aligned} \tag{14}$$

Finally, the formula of RBF-based deformation can be summarized as follows:

$$\Delta\mathbf{x} = (\mathbf{A}\mathbf{H})\Delta\mathbf{P} = \mathbf{b}_{rbf}\Delta\mathbf{P} \tag{15}$$

The readers can notice the similarity of the formulations of the FFD and RBF methods [Eqs. (3) and (15)]. A simple application of the RBF approach to a 2-D airfoil is also included in Fig. 1. The major differences between the FFD and RBF are summarized as follows:

1) The deformation is limited within the control box in FFD, whereas the deformation can extend to the whole computational domain in RBF, due to the effect of the linear polynomial term.

2) In RBF, a simple mode of deformation such as parallel shift or rotation can be expressed for a whole domain without any displacements of centers, due to the linear polynomial term.

The main drawback of the standard FFD/RBF methods is that there is no direct (intuitive) relationship between the displaced control points and the deformed object [15]. Therefore, designers have to struggle with the unintuitive design parameters, whose bounds are difficult to define in design problems, and that can lead to ill-conditioned optimization problems. This led us to introduce direct-manipulation approaches described in the next section.

C. Direct Manipulation

In the field of computer graphics, the manipulation of curves/surfaces is classified into control point-based method and direct-manipulation methods. The control point-based method is the traditional way to manipulate curves/surfaces, due to the displacements of their control points. For example, in Bézier/B-spline curves, users specify the displacements of the control points to deform an object. In FFD, users specify the displacements of the points of control box. One of the limitations of this control point-based method is that the design variables may have no physical significance for designers.

Direct manipulation [35–38] is a generic method to automatically find the displacements of control points that satisfy user-specified constraints. The user-specified constraints are usually defined as the displacements of representative points defined on an object. (We call these points *pilot points*.) In this method, users can pick any points on the object as the pilot points, and then move them to desired locations. In Fig. 2, the flowchart of the two methods is shown. In general, the direct-manipulation method leads to a better and easier control of the deformation and provides users with an interface that is more intuitive than the control point-based method. So it is considered to be a promising method for our field of application.

In the direct-manipulation approach, the number of control points corresponds to the degree of freedom, and the number of pilot points corresponds to the number of user-specified constraints. If the number of control points is greater than the number of pilot points, which is usual in our actual applications, the direct-manipulation problem is under-constrained. In this condition, an infinite number of solutions can satisfy the user-specified displacements on the pilot points. Among these solutions, we consider in this paper the one obtained by direct linear method [35,36] and another one obtained by physics-based methods [37,38] that are grounded on an elastic analogy. The direct linear method is the straightforward application of a linear algebra method to invert the system of equations. The physics-based method has been developed in the field of computer graphics during the last 10 years. In this method, at the crossroads of computer graphics and solid mechanics disciplines, the new locations of the control points are determined along with the minimization of a deformation energy, which includes physical properties as stretching/bending stiffnesses. Therefore, objects can be manipulated more intuitively as real elastic objects.

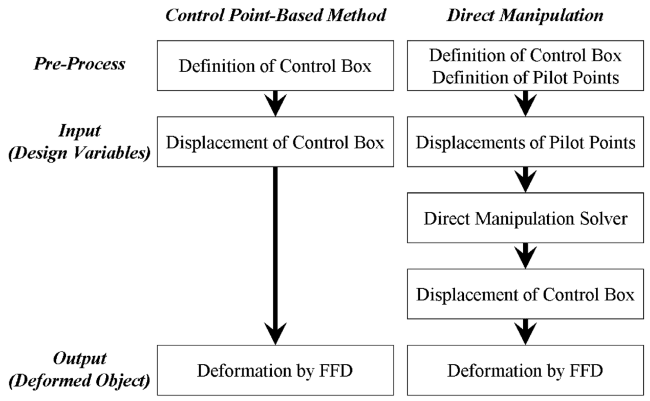

Fig. 2 Flowchart of control point-based method and direct manipulation in FFD.

D. Direct Manipulation for FFD

Since the conventional (control point-based/center-based) FFD/RBF approaches have several advantages over other parameterization methods, the introduction of the direct manipulation in FFD/RBF is considered as promising parameterization methods. The principles of the introduction of the direct manipulation in FFD [35,36] are recalled below. The direct linear method, which is the classical solution to solve this problem is presented in the next section, and then a new formulation of the physics-based approach is introduced.

When one assigns the displacements $\Delta\mathbf{X}$ to the pilot points located on an object, the goal is to determine the displacements of the points of control box $\Delta\mathbf{P}$, which satisfy the user-specified displacements. Suppose that q is the number of pilot points on which user specifies the displacements, n is the number of control points of the control box. Considering the formula of FFD, Eq. (3), the displacements on the control points have to satisfy the following equation for any pilot point i with the local curvilinear coordinates $(s,t,u)_i$:

$$\Delta\mathbf{X}_i = \mathbf{b}(s,t,u)_i \Delta\mathbf{P} \tag{16}$$

For all pilot points,

$$\begin{pmatrix} \Delta X_{11} & \Delta X_{12} & \Delta X_{13} \\ \vdots & \vdots & \vdots \\ \Delta X_{q1} & \Delta X_{q2} & \Delta X_{q3} \end{pmatrix} = \begin{pmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & & \vdots \\ B_{q1} & \cdots & B_{qn} \end{pmatrix} \times \begin{pmatrix} \Delta P_{11} & \Delta P_{12} & \Delta P_{13} \\ \vdots & \vdots & \vdots \\ \Delta P_{n1} & \Delta P_{n2} & \Delta P_{n3} \end{pmatrix}$$

$$\Delta\mathbf{X} = \mathbf{B}\Delta\mathbf{P} \tag{17}$$

where $\Delta\mathbf{X}$ is the matrix expressing the user-specified displacements on the pilot points, $\Delta\mathbf{P}$ is the unknown matrix, and $\mathbf{B}$ is a $(q \times n)$ irregular blending matrix. The subscript for the indexes of control points was changed hereafter for more concise description. When the number of control points is greater than the number of pilot points, an infinite number of solutions can generally satisfy this system.

E. Direct Linear Method

The straightforward way to determine $\Delta\mathbf{P}$ is to use the Moore–Penrose pseudoinverse matrix $\mathbf{B}^+$ of $\mathbf{B}$ [35]. The pseudoinverse solution satisfies the following properties when $\mathbf{B}$ is a $(q \times n)$ matrix:

If $q = n$, the pseudoinverse matrix is exactly the inverse matrix $\mathbf{B}^{-1}$ if the matrix $\mathbf{B}$ is invertible.

If $q > n$, it is generally impossible to find an exact solution that satisfies the user-specified constraints. The pseudoinverse solution is the one that minimizes the norm $\|\Delta\mathbf{X} - \mathbf{B}\Delta\mathbf{P}\|$.

If $q < n$ (this is usually the case in the present applications), there are generally an infinite number of solutions, and the pseudoinverse solution is the one that minimizes the quantity $\|\Delta\mathbf{P}\|$.

Once the pseudoinverse matrix is determined, the unknown matrix $\Delta\mathbf{P}$ can be solved as follows:

$$\Delta\mathbf{P} = \mathbf{B}^+\Delta\mathbf{X} \tag{18}$$

As a way to elude the implementation of the pseudoinversion, another methodology using a Lagrange function has been developed [36], and the solution also minimizes the quantity $\|\Delta\mathbf{P}\|$. These

methodologies that search a solution minimizing $\|\Delta\mathbf{P}\|$ are called direct linear method and they make use of the conventional linear algebra. The main drawback of this method is that the minimized quantity offers no physical meaning. The obtained solution may therefore exhibit unintuitive interpolation properties between pilot points, and has poor ability to preserve volume mesh quality around deformed objects. This led us to develop a new physics-based method described in the next section.

F. Physics-Based Method

In the physics-based method, the solution is determined by minimizing a deformation energy while satisfying the user-specified displacements [37,38]. In this research, the deformation energy w is defined as an isotropic elastic material:

$$w = \frac{1}{2}\int_{\Theta}[\lambda(\varepsilon_1+\varepsilon_2+\varepsilon_3)^2 + 2\mu(\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2+2(\varepsilon_4^2+\varepsilon_5^2+\varepsilon_6^2))]\,\mathrm{d}Q \tag{19}$$

where the integral region Θ and the elementary volume $\mathrm{d}Q$ can be defined in many ways that are investigated later. λ and μ are called Lamé's constants. In our field of application, it is more meaningful to express them as functions of the modulus of elasticity E and Poisson ratio υ:

$$\lambda = \frac{\upsilon E}{(1+\upsilon)(1-2\upsilon)}, \qquad \mu = \frac{E}{2(1+\upsilon)} \tag{20}$$

In this research, $\upsilon = 0.25$ is used in general. E is simply a scaling factor of the above defined energy and can be set arbitrarily. It can also be set variably in space, and the effect is discussed later. In the matrix form, Eq. (19) is written as follows:

$$w = \frac{1}{2}\int_{\Theta}[\varepsilon^T\mathbf{D}\varepsilon]\,\mathrm{d}Q \tag{21}$$

where ε and $\mathbf{D}$ correspond, respectively, to the strain matrix and constitutive matrix as follows:

$$\varepsilon = \begin{pmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\\ \varepsilon_5\\ \varepsilon_6\end{pmatrix} = \begin{pmatrix}\partial(\delta x_1)/\partial x_1\\ \partial(\delta x_2)/\partial x_2\\ \partial(\delta x_3)/\partial x_3\\ 0.5[\partial(\delta x_2)/\partial x_1+\partial(\delta x_1)/\partial x_2]\\ 0.5[\partial(\delta x_3)/\partial x_1+\partial(\delta x_1)/\partial x_3]\\ 0.5[\partial(\delta x_2)/\partial x_3+\partial(\delta x_3)/\partial x_2]\end{pmatrix} \tag{22}$$

$$\mathbf{D} = \begin{pmatrix}\lambda+2\mu & \lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0\\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0\\ 0 & 0 & 0 & 4\mu & 0 & 0\\ 0 & 0 & 0 & 0 & 4\mu & 0\\ 0 & 0 & 0 & 0 & 0 & 4\mu\end{pmatrix} \tag{23}$$

In cases of the FFD, the derivative of displacement $\partial(\Delta\mathbf{x})/\partial\mathbf{x}$ can be expressed as follows:

$$\frac{\partial(\Delta\mathbf{x})}{\partial\mathbf{x}} = \Delta\mathbf{P}^T\frac{\partial\mathbf{b}}{\partial\mathbf{x}} = \Delta\mathbf{P}^T\frac{\partial\mathbf{b}}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial\mathbf{x}} \tag{24}$$

where $\mathbf{U}$ is the local curvilinear coordinates in the control box. $\partial\mathbf{b}/\partial\mathbf{U}$ is the $(n\times 3)$ matrix whose column vector is the derivatives of each $\mathbf{b}$ component with respect to a component of $\mathbf{U}$. By using the inverse function theorem,

$$\frac{\partial\mathbf{U}}{\partial\mathbf{x}} = \left(\frac{\partial\mathbf{x}}{\partial\mathbf{U}}\right)^{-1} = \left(\mathbf{P}^T\frac{\partial\mathbf{b}}{\partial\mathbf{U}}\right)^{-1} \tag{25}$$

Combining Eqs. (24) and (25),

$$\frac{\partial(\Delta\mathbf{x})}{\partial\mathbf{x}} = \Delta\mathbf{P}^T\frac{\partial\mathbf{b}}{\partial\mathbf{U}}\left(\mathbf{P}^T\frac{\partial\mathbf{b}}{\partial\mathbf{U}}\right)^{-1} \equiv \Delta\mathbf{P}^T\cdot\mathbf{R} \tag{26}$$

where $\mathbf{R}$ is the $(n\times 3)$ matrix defined at each point in the control box. Then the strain matrix ε can be expressed as

$$\varepsilon = (\,\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3 \quad \varepsilon_4 \quad \varepsilon_5 \quad \varepsilon_6\,)^T = \mathbf{S}\cdot\Delta\bar{\mathbf{P}}$$

$$\mathbf{S} = \begin{pmatrix}R_{11} & 0 & 0 & \cdots & R_{n1} & 0 & 0\\ 0 & R_{12} & 0 & \cdots & 0 & R_{n2} & 0\\ 0 & 0 & R_{13} & \cdots & 0 & 0 & R_{n3}\\ 0.5R_{12} & 0.5R_{11} & 0 & \cdots & 0.5R_{n2} & 0.5R_{n1} & 0\\ 0.5R_{13} & 0 & 0.5R_{11} & \cdots & 0.5R_{n3} & 0 & 0.5R_{n1}\\ 0 & 0.5R_{13} & 0.5R_{12} & \cdots & 0 & 0.5R_{n3} & 0.5R_{n2}\end{pmatrix}$$

$$\Delta\bar{\mathbf{P}} = (\,\Delta P_{11} \quad \Delta P_{12} \quad \Delta P_{13} \quad \cdots \quad \Delta P_{n1} \quad \Delta P_{n2} \quad \Delta P_{n3}\,)^T \tag{27}$$

Finally, the deformation energy w can be expressed as follows:

$$w = \Delta\bar{\mathbf{P}}^T\left[\int_{\Theta}\left(\frac{1}{2}\mathbf{S}^T\mathbf{D}\mathbf{S}\right)\mathrm{d}Q\right]\Delta\bar{\mathbf{P}} = \Delta\bar{\mathbf{P}}^T\mathbf{K}\Delta\bar{\mathbf{P}} \tag{28}$$

where $\mathbf{K}$ is the $(3n\times 3n)$ stiffness matrix of the region Θ with respect to the degree of freedom $\Delta\bar{\mathbf{P}}$.

The physics-based direct-manipulation problem can be defined as follows:

Find $\Delta\bar{\mathbf{P}}$ that minimizes $w = \Delta\bar{\mathbf{P}}^T\mathbf{K}\Delta\bar{\mathbf{P}}$ while satisfying $\Delta\mathbf{X} = \mathbf{B}\Delta\mathbf{P}$.

By the introduction of the Lagrange multipliers, the Lagrangian can be defined as

$$J = \Delta\bar{\mathbf{P}}^T\mathbf{K}\Delta\bar{\mathbf{P}} + \sum_{i=1}^{q}(\Delta\mathbf{X}_i - \mathbf{B}_i\Delta\mathbf{P})\lambda_i \tag{29}$$

where $\lambda_i = (\,\lambda_{i1} \quad \lambda_{i2} \quad \lambda_{i3}\,)^T$. A new unknown matrix ξ is now introduced, instead of $\Delta\bar{\mathbf{P}}$, as

$$\xi = (\,\Delta P_{11} \quad \Delta P_{12} \quad \Delta P_{13} \quad \cdots \quad \Delta P_{n3} \quad \lambda_{11} \quad \lambda_{12} \quad \lambda_{13} \quad \cdots \quad \lambda_{q3}\,)^T \tag{30}$$

Then Equation (29) can be expressed as follows:

$$J = \xi^T\bar{\mathbf{K}}\xi + \xi^T\Delta\bar{\mathbf{X}} - \xi^T\bar{\mathbf{B}}\xi \tag{31}$$

where the $\bar{\mathbf{K}}$ and $\bar{\mathbf{B}}$ are $(3n+3q)$ symmetric matrices and $\Delta\bar{\mathbf{X}}$ is the $(3n+3q)$ column matrix with respect to the $\mathbf{K}$, $\mathbf{B}$ and $\Delta\mathbf{X}$, respectively. To minimize the deformation energy, $\partial J/\partial\xi = 0$ is required and the unknown matrix ξ is evaluated as follows:

$$\xi = -\tfrac{1}{2}(\bar{\mathbf{K}} - \bar{\mathbf{B}})^{-1}\Delta\bar{\mathbf{X}} \tag{32}$$

Equation (32) means that the physics-based method requires the inversion of a $(3n+3q)$ regular matrix, and the direct linear method requires the pseudoinversion of a $(q\times n)$ irregular matrix.

The two direct-manipulation methods for FFD, direct linear method and physics-based method, were discussed in the sections II.D–II.F. The direct-manipulation methods can be introduced to the RBF approach in the same manner because of the similarity of the formulations.

III. Verification of the Developed FFD/RBF

In this section, the verification of the developed FFD/RBF direct-manipulation methods is conducted for a simple 2-D airfoil manipulation. Although the FFD method can be used with Bézier, B-spline, or NURBS blending functions, the B-spline blending function is generally employed in this study because of the locality. A structured CFD mesh around a unit chord NACA0012 airfoil is used for this study. The control box is defined by a simple Cartesian mesh with 100 control points (10×10). Two pilot points are defined on the leading/trailing edges, and the pilot point on the leading edge is displaced upward, whereas the trailing-edge location is fixed in this

study, as shown in Fig. 3. The readers can guess that the most straightforward solution in this case is a rigid rotating airfoil. Therefore, the flexibilities of the developed FFD/RBF methods for such a simple mode of deformation can be also discussed.

In Fig. 4, the object deformed by the direct linear FFD method is presented. In this computation, the pseudoinversion of a 2×100 irregular matrix is required as understood from Eq. (17). The root-mean-squared error (RMSE) on the pilot points (the error between the user-specified displacements and actual ones) was 0.59×10^{-16}, and the value of $\|\Delta \mathbf{X}\|$ is 0.2. This result witnesses the accuracy of the matrix numerical inversion. But it was also confirmed that the volume mesh around the leading edge was overlapped because the formulation of the direct linear method did not take the volume mesh quality into consideration.

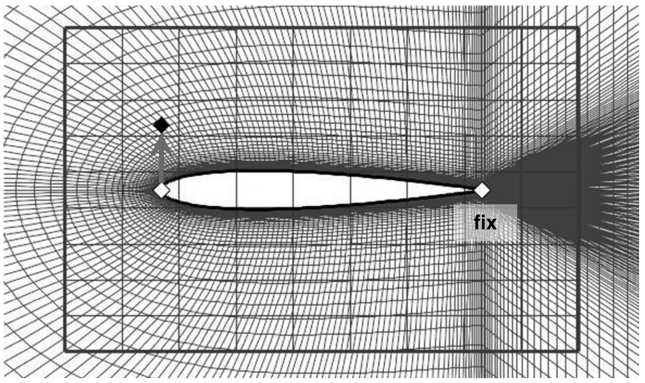

Fig. 3 Setting of the Cartesian control box and two pilot points on the leading/trailing edges of an airfoil; white points: initial pilot points and black points: displaced pilot points.

In Fig. 5, the objects deformed by the physics-based FFD method with different definitions of the deformation energy are shown. In the first case, the deformation energy per unit thickness is defined on the airfoil surface as follows:

$$w = \frac{1}{2} \int_{\text{Airfoil}} [\varepsilon^T \mathbf{D} \varepsilon] \, \mathrm{d}S \tag{33}$$

In the second case, the deformation energy is defined for the whole control box as follows:

$$w = \frac{1}{2} \int_{\text{Control Box}} [\varepsilon^T \mathbf{D} \varepsilon] \, \mathrm{d}V \tag{34}$$

In these computations, the inversion of a regular matrix (306×306) is required as understood from Eq. (32). The RMSEs on the pilot points were 0.27×10^{-13} and 0.13×10^{-12} in these cases. Although the airfoil geometry was relevant at the case of Eq. (33), the volume mesh quality was not. This is because the deformation energy was minimized on the airfoil only. At the case of Eq. (34), on the other hand, the deformed control box (and the deformed airfoil) was obtained just as a simple rotation. This is because this simple rotation is the *physical* solution that minimizes the deformation energy in the control box, so that this result showed the correctness of the developed physics-based direct-manipulation method.

As the reader may notice, the mesh connectivity at the border of control box is still incorrect, so that this issue has to be improved to enable the deformation of CFD volume meshes. This issue can be solved by adding a constraint on the control points located on the border of lattice that freezes their positions [35]. In Fig. 6, the objects deformed by the physics-based FFD method with the constraint on the border of control box are indicated. In this computation, the inversion of a 309×309 regular matrix is required. The increase in the size of the matrix compared with the previous case corresponds to

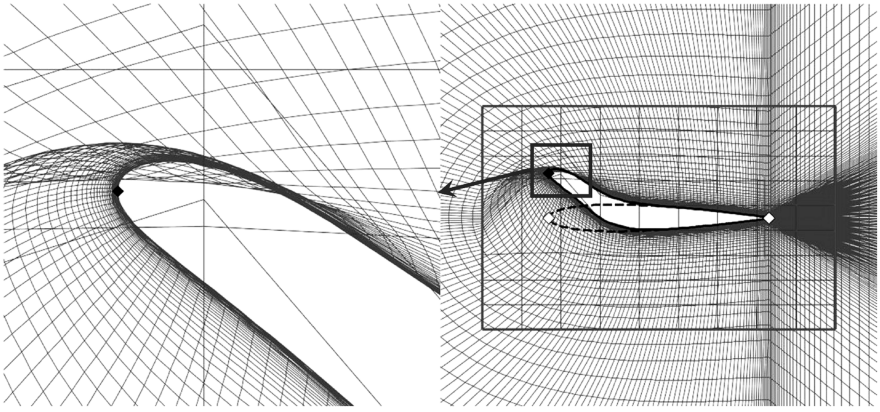

Fig. 4 Object deformation by direct linear FFD method (volume mesh is irrelevant); left: enlarged view around the leading edge and right: general view for the deformed airfoil.

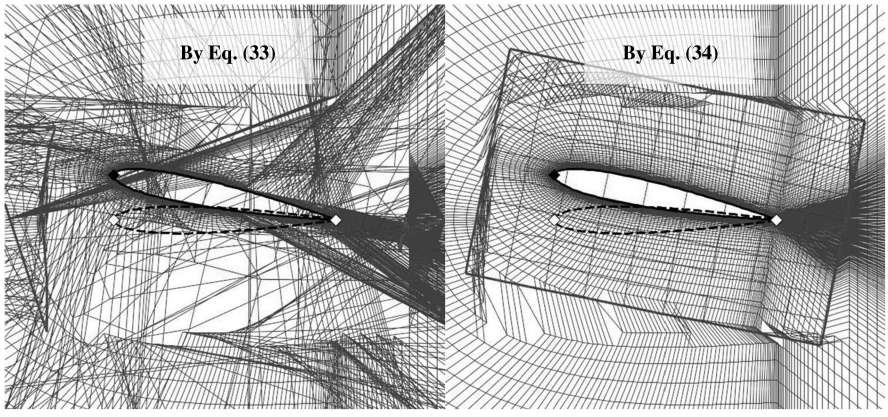

Fig. 5 Objects deformation by physics-based FFD method (volume meshes are irrelevant); left: by Eq. (33) and right: by Eq. (34).

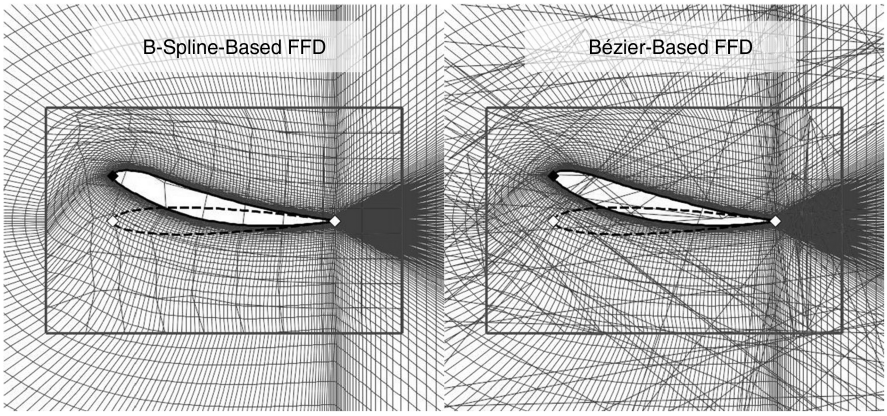

Fig. 6 Objects deformation by physics-based FFD method with a constraint on the border of control box (volume meshes are relevant); left: B-spline-based FFD and right: Bézier-based FFD.

the border constraints handling. The RMSE on the pilot points was 0.73×10^{-11} in this case, and an acceptable deformed mesh was obtained by adding the constraint. Although the mesh quality at the upstream boarder was not good yet, this issue can be reduced to set the border of control box further away. The effect of the blending function is also examined in Fig. 6 between the B-spline-based and Bézier-based blending functions. If the control box is sufficiently refined, both cases should converge toward the same result, corresponding to the solution of the elastic deformation problem that is independent of the chosen volume discretization. However, the B-spline-based blending function is generally employed in this study because of the locality.

The deformed (rotated) airfoil was distorted by adding the constraint on the border of control box (comparison between Figs. 5 and 6). It means that simple transformations of objects, such as the rotation and/or parallel shift, are difficult to handle by the developed FFD method with the constraint on the border. To handle these kinds of deformations, a technique using a damping function [7] is preferable to the addition of the constraint on the border. The damping function ω is defined in each direction of the local curvilinear coordinates. This function should be equal to 1 in the major part of the local coordinate (objects should exist in this region), and equal to zero at the border of the local coordinate and should have high order continuity properties between them, as shown in Fig. 7. Then the formula of FFD, Eq. (3), can be modified as follows:

$$\Delta \mathbf{x} = (\omega(s) \cdot \omega(t) \cdot \omega(u)) \cdot \mathbf{b}(s, t, u) \cdot \Delta \mathbf{P} \tag{35}$$

The damping function does not affect the major part of the control box, whereas it enables a smooth damping of the mesh displacement at the border. Although the appropriate choice and setting of the damping function are essential and they require trial-and-error procedures, this approach can handle the rotation/parallel shift correctly, as shown in Fig. 7. Although the deformed mesh was highly distorted around the leading edge, this issue can be reduced to set the border of control box further away.

In Fig. 8, the object deformed by the physics-based RBF method without the border constraint is shown. The 100 control points used in the above FFD cases are now used as the centers. The deformation can reach the far field in the RBF method, so the deformation energy can be defined for the whole computational mesh as follows:

$$w = \frac{1}{2} \int_{\text{Mesh}} [\varepsilon^T \mathbf{D} \varepsilon] \, \mathrm{d}V \tag{36}$$

In this computation, the inversion of a matrix of size 306×306 is required. The RMSE on the pilot points was 0.29×10^{-12}. Again, the deformed centers and airfoil were obtained as a simple rotation. In addition, the effect of the rotation extended to the outer boundary of the computational mesh, due to the effect of the linear polynomial term of the RBF method. It is understood therefore that the RBF method can handle these simple modes of deformation easily without any special treatment. However, the propagation of the deformation to far field may be undesirable particularly for unsteady flow cases. Then users have several options, such as to use a damping function as Fig. 7 only for the far-field direction to reduce the propagation. It is also possible to eliminate the linear polynomial term from the RBF formulation although it cannot express the simple mode of deformation anymore.

IV. Properties of the Developed FFD/RBF

In this section, the properties of the developed FFD/RBF methods are examined. Although the properties described in Secs. IV.D and IV.E only concern the developed FFD method, the other properties are common to both developed methods.

A. Application to Arbitrary Objects and Meshes

The most crucial advantage of the developed FFD/RBF methods is the applicability to arbitrary objects. With these methods, what users have to set is only to define a control box (or a set of centers) that surrounds an object and pilot points at any positions on the object.

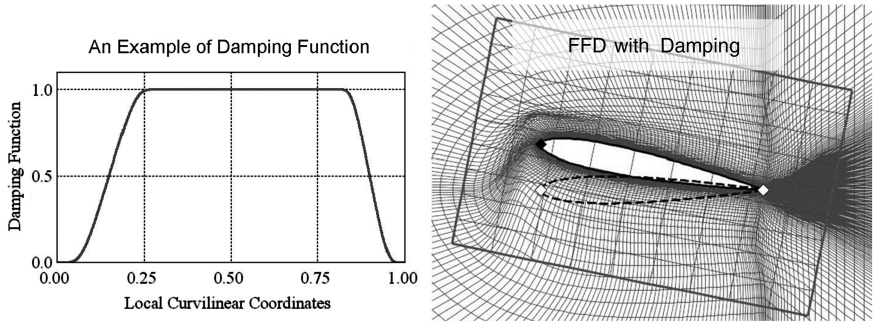

Fig. 7 Application of a damping function to physics-based FFD without the constraint on boarder (volume mesh is relevant); left: an example of damping function and right: object deformation by physics-based FFD with the damping function.

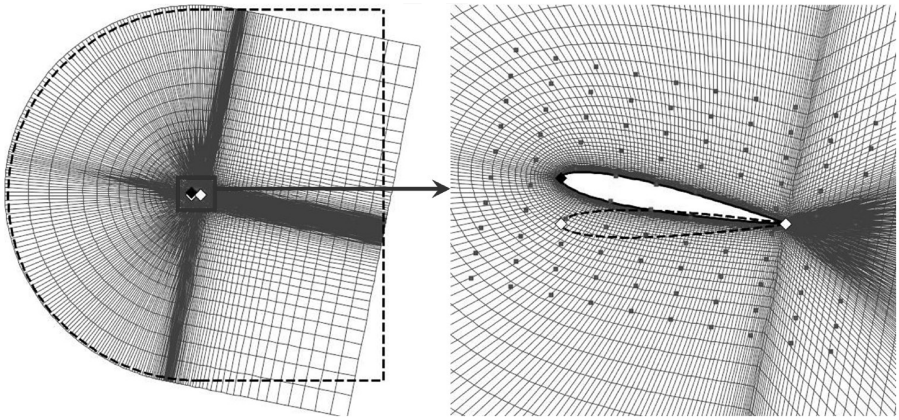

Fig. 8 Object deformation by physics-based rbf method without any constraints (volume mesh is relevant); left: far-field visualization and right: deformed object visualization.

In addition, the mesh deformation procedures are independent from the mesh topological information. Therefore, these methods can be used as a mesh deformation strategy for arbitrary mesh topologies without any modification in the computational code.

In Fig. 9, the application of the physics-based FFD method to a 59-block structured mesh around a wing–body configuration is indicated. Five pilot points were defined on the kink and tip of the wing, and the pilot points on the tip were displaced. In conventional mesh deformation strategies, some information has to be exchanged with the neighboring mesh blocks and some appropriate boundary conditions have to be constructed. In the developed FFD/RBF methods, on the other hand, such procedures are not necessary and they can be easily implemented. This flexibility is particularly important since presently existing CFD software tends to include more and more types of cells and connectivity in order to ease the mesh generation process (e.g., Chimera method, polygonal meshes, meshless methods, mixture of structured, unstructured and Cartesian meshes). In addition, it could be confirmed that the developed methods were robust for the general range of deformations that will be given in optimization process. In Fig. 10, the application of the physics-based RBF method to an unstructured mesh around a soccer ball is indicated. Four pilot points were defined and one was displaced. The intuitive deformation as a real elastic object was confirmed.

B. Analytical Mesh Sensitivity

In adjoint-based gradient optimization methods, the calculation of mesh sensitivities with respect to design variables is essential. Traditionally, the mesh sensitivity is evaluated by the finite difference, which requires many executions of a mesh deformation code (grows linearly with the number of design variables). Although a method to eliminate the mesh sensitivity from adjoint problems has been recently developed [39], it requires the modification of adjoint codes and takes an additional computational cost. In the developed FFD/RBF methods, the mesh sensitivities of any node points with respect to a pilot point displacement $\partial(\mathbf{x} + \Delta\mathbf{x})/\partial(\Delta\mathbf{X})$ can be evaluated analytically by using the following formula:

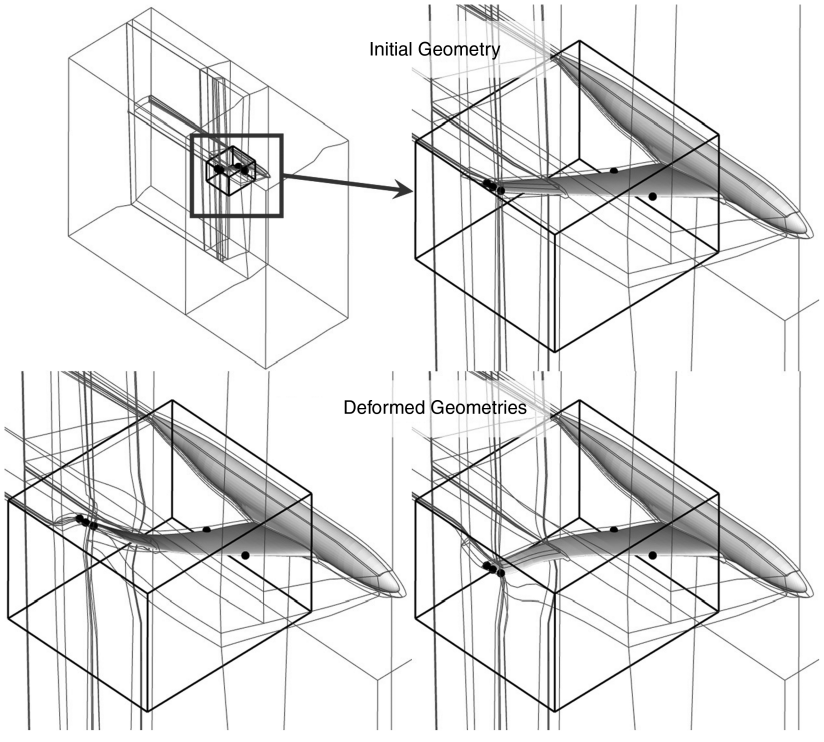

Fig. 9 Application to a multiblock structured mesh around a wing/body configuration; upper: initial geometry and lower: deformed geometries.

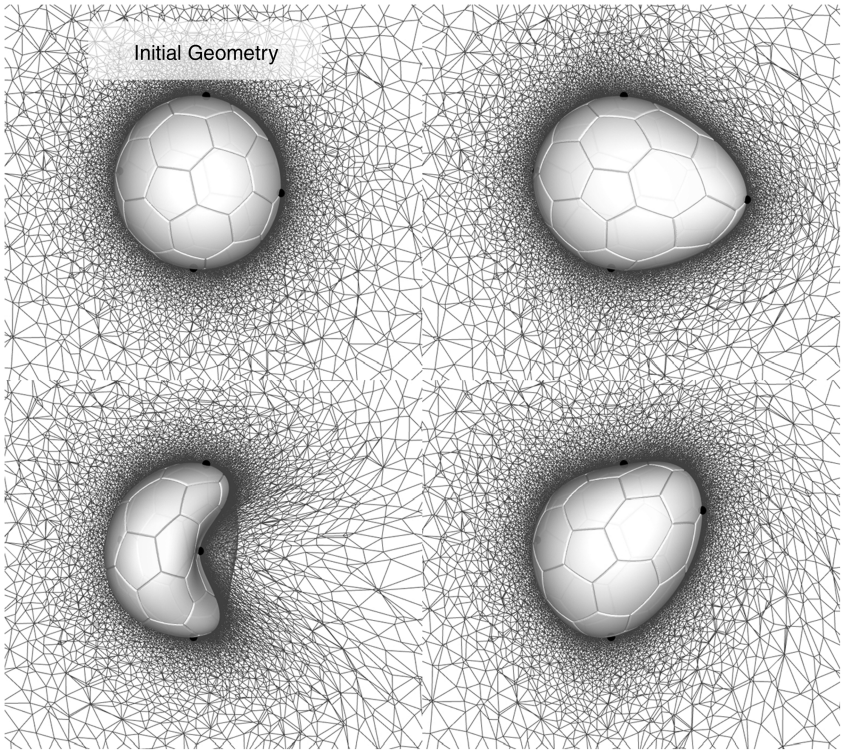

Fig. 10 Application to an unstructured mesh around a soccer ball; left/upper: initial geometry and others: deformed geometries.

$$\frac{\partial(\mathbf{x}+\Delta\mathbf{x})}{\partial(\Delta\mathbf{X})}=\frac{\partial(\Delta\mathbf{x})}{\partial(\Delta\mathbf{X})}=\frac{\partial(\mathbf{b}\Delta\mathbf{P})}{\partial(\Delta\mathbf{X})}=\mathbf{b}\cdot\frac{\partial(\Delta\mathbf{P})}{\partial(\Delta\mathbf{X})} \tag{37}$$

The matrix **b** depends only on the local coordinates. For the direct linear method, the derivative $\partial(\Delta\mathbf{P})/\partial(\Delta\mathbf{X})$ is given by the pseudoinverse matrix $\mathbf{B}^{+}$ as understood from Eq. (18). For the physics-based method, the derivative can be obtained from the last $3q$ column vectors of the inverse matrix $(\bar{\mathbf{K}}-\bar{\mathbf{B}})^{-1}$ as understood from Eq. (32). Although the same kind of analytical evaluation formulas may be possible for other conventional mesh deformation methods, the size of matrix that has to be inverted is proportional to the number of grid points in these methods. So it is almost impossible in real problems to obtain the inverted matrix explicitly. On the other hand, it is possible with the developed methods since the size of matrix is perfectly independent of the number of grid points.

In Fig. 11, the analytical mesh sensitivity of the physics-based FFD method is compared with that obtained by the finite difference for the 2-D airfoil case discussed in the previous section. The mesh sensitivities with respect to the vertical displacement of a pilot point located at the leading edge were visualized. The distributions by the finite difference and by the analytical evaluation are in perfect agreement, and this property is considered to be useful for adjoint-based optimization methods.

C. Mesh-Quality Preservation and Computational Cost

To investigate the quality of deformed mesh, the mesh deformed by the developed FFD method was compared with that by a tension/torsion spring-analogy-based mesh deformation method [29,31]. In the FFD method, a 10×10 control box was defined as a Cartesian mesh, and two pilot points were defined at the leading/trailing edges similarly to the previous section. To increase the robustness of the FFD method, variable definition of the modulus of elasticity E, which appears in the deformation energy is adopted [32]. This is capable of reproducing solid body motions for stiff (higher E) regions, so that the variable definition of E is considered to be effective for the physics-based direct-manipulation approach. In this research, the modulus of elasticity of node i is taken as inversely proportional to the control volume as follows:

$$E_i = E_0\left(\frac{V_0}{V_i}\right)^p \tag{38}$$

where E_0 and V_0 are the constants, and V_i is the control volume of node i. p is a factor controlling the stiffness distribution. At $p=0$, the stiffness is homogeneously distributed in the volume. At $p<0$, the stiffness of larger mesh cells is larger, and at $p>0$ the stiffness of smaller mesh cells is larger. In general, $p \sim 1$ is considered as the

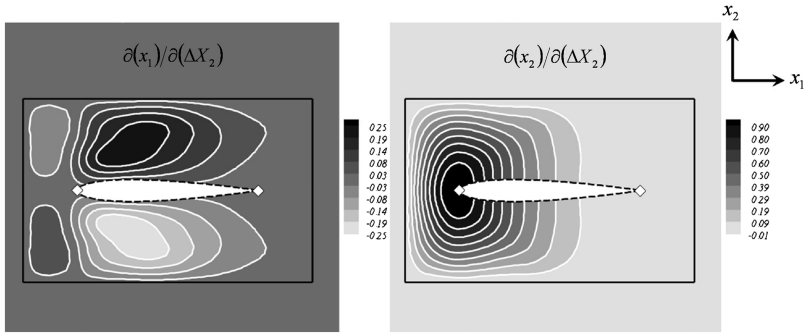

Fig. 11 Mesh sensitivity with respect to the vertical displacement of a pilot point on the leading edge; gray scale: analytical mesh sensitivity, white solid line: mesh sensitivity by finite differencing, black solid line: border of control box, and black broken line: initial geometry of airfoil.

robust set for CFD meshes. Then the stiffness matrix can be defined by integrating on the mesh points that are within the control box:

$$\mathbf{K} = \int_{\Theta} \left(\frac{1}{2} \mathbf{S}^T \mathbf{D} \mathbf{S} \right) \mathrm{d}Q = \frac{1}{2} \sum_{i=1}^{\text{node}} [\mathbf{S}_i^T \mathbf{D}_{(E_i)} \mathbf{S}_i] \cdot V_i \tag{39}$$

An alternative definition of this variable stiffness is to use the distance from the object d_i as the weighting criteria as follows:

$$E_i = E_0 \left(\frac{d_0}{d_i} \right)^p \tag{40}$$

Then the stiffness matrix can be defined by integrating on arbitrary points within the control box (e.g., uniformly distributed points in the control box). The advantage of this definition is that the deformed object becomes absolutely independent from the CFD mesh. However, Eqs. (38) and (39) are used in this research since the mesh quality can be better preserved by using the mesh points directly to construct the stiffness matrix.

In the spring-analogy-based method, the displacements on the airfoil surface obtained by the FFD method were employed as the input displacements. The tension and torsion spring stiffness of each edge were taken as inversely proportional to the edge length and the sine of the angle between two edges, respectively. Then the static equilibrium equations were solved by a Jacobi iteration method. In Fig. 12, the deformed-mesh distributions of both methods are compared. As understood from the enlarged view around the airfoil surface, the mesh orthogonality, which is one of the major mesh-quality measures for CFD applications was better preserved by the physics-based method. For more detailed comparison of the deformed-mesh quality, the following mesh-quality measures for smoothness/orthogonality were evaluated on all mesh points:

$$M_{\text{smt}} = \sum_{i=1} \sum_{j=1} \| \mathbf{r}_{i+1,j} + \mathbf{r}_{i,j+1} + \mathbf{r}_{i-1,j} + \mathbf{r}_{i,j-1} \| \tag{41}$$

$$M_{\text{ort}} = \sum_{i=1} \sum_{j=1} [(\mathbf{r}_{i+1,j} \cdot \mathbf{r}_{i,j+1})^2 + (\mathbf{r}_{i+1,j} \cdot \mathbf{r}_{i,j-1})^2 + (\mathbf{r}_{i-1,j} \cdot \mathbf{r}_{i,j+1})^2 + (\mathbf{r}_{i-1,j} \cdot \mathbf{r}_{i,j-1})^2] \tag{42}$$

where $\mathbf{r}$ expresses the vector from a node point (i, j) to a neighboring node as $\mathbf{r}_{i+1,j} = \mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}$. The comparison of the mesh-quality measures is shown in Table 1. It was confirmed that the developed method showed better mesh quality, especially in terms of mesh orthogonality.

Table 1 Comparison of mesh quality measures

	$M_{\text{smt}}/M_{\text{smt}_{\text{init}}}$	$M_{\text{ort}}/M_{\text{ort}_{\text{init}}}$
Initial mesh	1.000	1.000
Spring analogy	1.020	1.091
Physics-based ffd	1.022	1.030

The computational cost of the physics-based approaches mainly depends on the number of control points (number of centers in RBF case). It is more expensive than the computational cost of the spring-analogy-based method in total. The most costly procedures of the developed methods are the construction of the stiffness matrix $\mathbf{K}$ and the inversion of the matrix $\bar{\mathbf{K}} - \bar{\mathbf{B}}$. But these matrices depend only on the original positions of the pilot points (in fact, their local coordinates) and not on their displacements $\Delta\mathbf{X}$. This implies that the inverse matrix $(\bar{\mathbf{K}} - \bar{\mathbf{B}})^{-1}$ can be shared for the same set of pilot points (same parameterization), whatever their displacements. For design optimizations, therefore, the costly procedures have to be computed only once before the optimization loop, and then the inverse matrix can be used repeatedly in the optimization loop as long as the parameterization is not changed. In Table 2, the computational times for the case of Fig. 12 are compared. The computations were executed by an Intel Core 2 Duo 2.13 GHz processor (single run). In the *full mode* (it includes the construction of the stiffness matrix and inversion of the matrix $\bar{\mathbf{K}} - \bar{\mathbf{B}}$), the computational cost was much more expensive than the spring-analogy-based method. On the other hand, in the *skip mode* (which skips the costly procedures and is applicable during optimization loops), it was confirmed that the cost was reduced remarkably and was the same order as the spring-analogy-based method.

D. Deformation of the Symmetry Plane

This property holds for the FFD method only. In conventional mesh deformation methods, when a deformation is given on a symmetry plane of the mesh, its deformation is addressed first as a 2-D mesh deformation problem on the symmetry plane, and then the result is used as a part of input displacements for a 3-D volume mesh deformation, which leads to a quite complicated procedure. On the contrary, in the developed FFD method, this kind of problem can be treated easily by adding a constraint to the control points lying on the symmetry plane. Their out-of-plane displacements are forced to zero, whereas they can move freely in the two other directions. In Fig. 13, the application result to a wing/body configuration is indicated. In this case, four pilot points were set on the nose/tail of the fuselage and

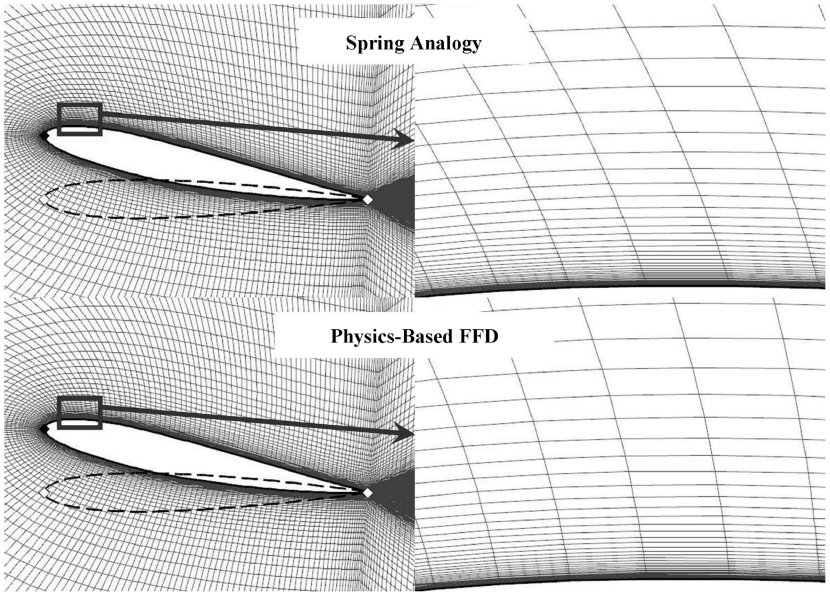

Fig. 12 Comparison of deformed-mesh distributions; upper: spring analogy and lower: physics-based FFD.

Table 2 Comparison of computational times for mesh deformation

	Physics-based FFD	
Spring analogy	Full mode	Skip mode
10 s	360 s	8 s

the leading/trailing edges of the wingtip. The pilot point on the nose was displaced upward, and the other points were fixed. It can be confirmed that the symmetry plane deformation can be achieved surely.

E. Use of a Nonparallelepiped Control Box

This property also holds for the FFD method only. Conventional rectangular control boxes may be difficult to adapt to some particular problems, especially with a small number of control points. This lack of freedom of rectangular control boxes has already been pointed out and some proposals to overcome this drawback were made [40]. The presently developed FFD method has the ability to handle arbitrary control boxes, due to an efficient algorithm to search the local curvilinear coordinates. Therefore, a C-type control box, as shown in Fig. 14, can be defined around the airfoil to increase the local controllability of the parameterization. The deformed results are compared in Fig. 14 between a rectangular control box and the C-type: they have the same number of points. In this case, a displacement was given only on a pilot point on the upper surface, and the other points were fixed. Although both cases gave possible computational meshes, the rectangular control box does not permit to handle the upper and lower surfaces independently, which is considered a disadvantage from the designer's point of view.

F. Addition of Geometrical Constraints

In the developed FFD/RBF methods, users can add some geometrical constraints by modifying the optimization problem for the physics-based method as follows:

Find $\Delta\bar{\mathbf{P}}$ that minimizes $w = \Delta\bar{\mathbf{P}}^T\mathbf{K}\Delta\bar{\mathbf{P}}$ while satisfying $\Delta\mathbf{X} = \mathbf{B}\Delta\mathbf{P}$ and some geometrical constraints $\Delta\mathbf{V} = \mathbf{W}\Delta\mathbf{P}$, where the RHS and left-hand side of the geometrical constraints equation indicate the variation of some geometrical parameters and the user-specified variations, respectively. For example, if the matrix $\mathbf{W}$ is defined for the sectional area of an airfoil and $\Delta\mathbf{V}$ is set to zero, a volume preserving parameterization of the airfoil can be realized, as

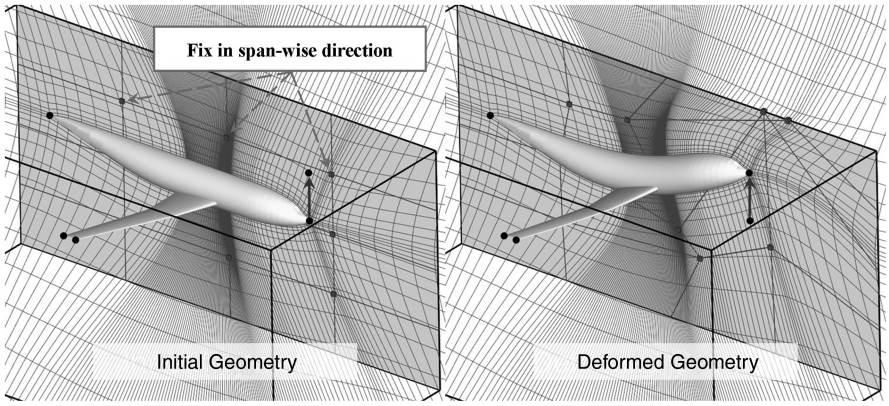

Fig. 13 Application to a symmetry plane deformation of a wing/body configuration; left: initial set and right: deformed geometry and mesh.

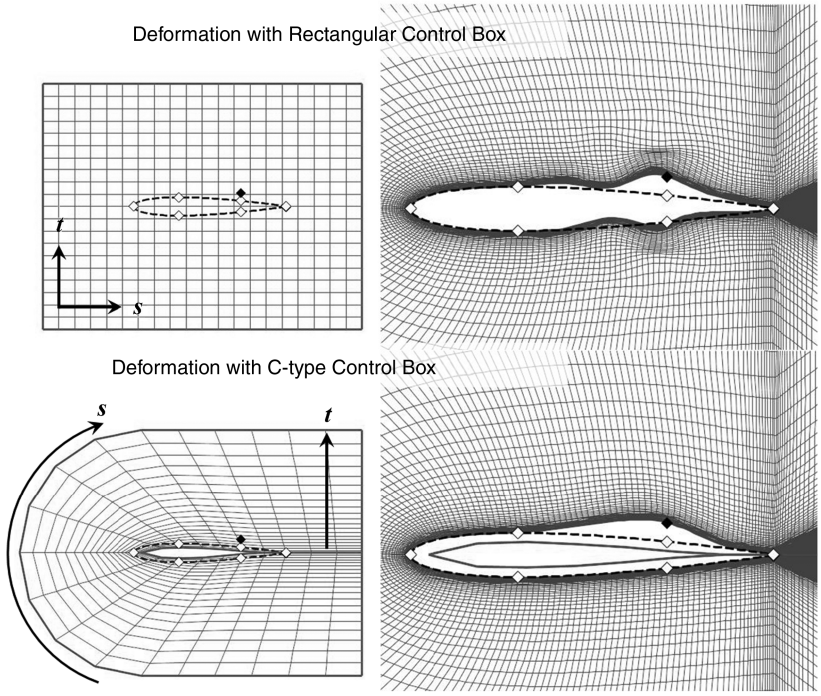

Fig. 14 Comparison of FFD deformations with different control boxes; upper: rectangular control box, lower: C-type control box, left: initial set of control boxes, and right: deformed airfoils.

illustrated in Fig. 15. In this calculation, the C-type control box introduced in the previous section was employed to increase the local controllability. It was confirmed that the sectional area of the airfoil could be preserved sufficiently. This technique can be applied not only to the sectional area, but also to the airfoil thickness at spar locations, wing volume, or any other geometrical parameters.

G. Enrichment of Design Space

Another advantage of the developed methods is that the addition of new pilot points (design variables) is possible with preserving deformed geometries, as shown in Fig. 16. This property of design space enrichment makes it possible to be applied to construct hierarchical/adaptive design optimizations. In these types of optimizations, new design variables can be added in the course of the optimization process, either uniformly or based on the significance of the design variables. In [7,16,18,19], the hierarchical/adaptive design optimizations were executed with the control point-based FFD method by using the degree-elevation property of the Bézier curves. This obliges to refine whole areas of the control box. By using the developed physics-based FFD/RBF methods, more intuitive hierarchical/adaptive design optimizations will be achieved.

Meanwhile, designers have to consider the number and relative locations of pilot points carefully since to set plural pilot points closely may provide bumpy deformed objects. In Fig. 17, a NACA0012 airfoil was deformed toward a RAE2822 airfoil. A set of 23 pilot points was used to represent the RAE2822 airfoil. It was confirmed that the upper surface was well represented, whereas the lower surface had small oscillations around the trailing edge. This problem tends to occur when the curvature of object is largely changed. It should be noted that the developed methods are not suitable to deform largely with a large number of pilot points, whereas the methods are suitable to deform an arbitrary object locally/globally by using a certain number of pilot points.

V. Hierarchical Design Optimization for 3-D Wing/Body Configuration

In this section, an example of the hierarchical design optimization using the developed FFD method is conducted for a 3-D wing/body configuration.

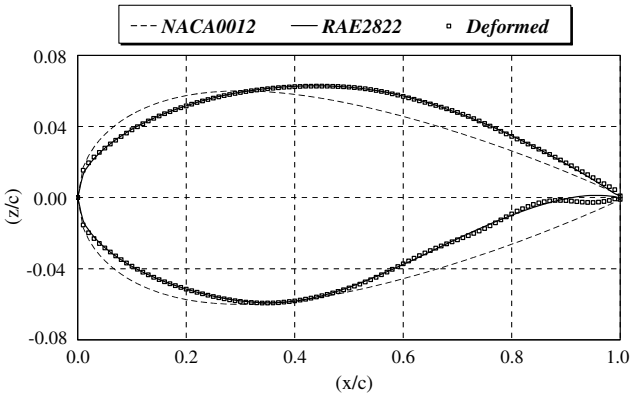

Fig. 17 Deformed airfoil from NACA0012 to represent RAE2822.

A. Design Problem

The design problem is to minimize the total drag of a wing/body configuration at an upstream Mach number of 0.80 and a Reynolds number of 5×10^6. An aerodynamic constraint is given for the lift coefficient, which has to be greater than 0.5. The flow evaluation is conducted by elsA [41,42]. Jameson's central difference scheme with artificial viscosity is used for space discretization. The Spalart–Allmaras turbulent model [43] is adopted with the assumption of fully turbulent flow. The initial configuration is DLR-F6 wing/body configuration with FX2B fairing. A multiblock structured mesh that was generated at the Third AIAA CFD Drag Prediction Workshop [44] is used. The number of mesh points (26 blocks) is about 2.6 million. The computational time for one flow evaluation is within 1 h by the parallel execution on four processors of NEC SX-8.

An optimization method based on a kriging surrogate model is employed in this optimization problem [3]. The initial sample points are generated by using Latin hypercube sampling (LHS) techniques. Once the kriging model is constructed, new sample points are explored on which exact (CFD) evaluations are executed to increase the accuracy of the surrogate model around the optimal region. In this research, the positions of new sample points are determined by a multi-objective optimization in which the estimated objective function (total drag) is minimized and the expected improvement (EI) value of total drag is maximized. To the author's best knowledge,

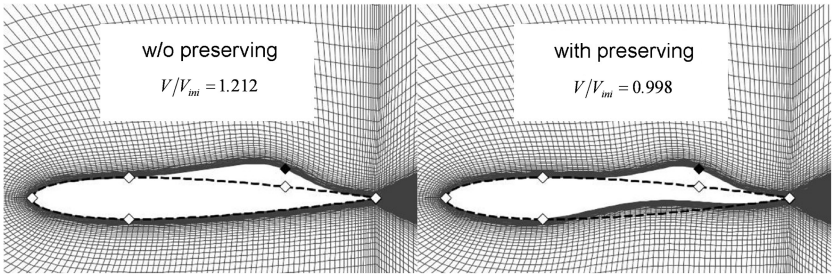

Fig. 15 Demonstration of volume preserving parameterization for a 2-D airfoil; left: conventional physics-based FFD and right: that with volume preserving.

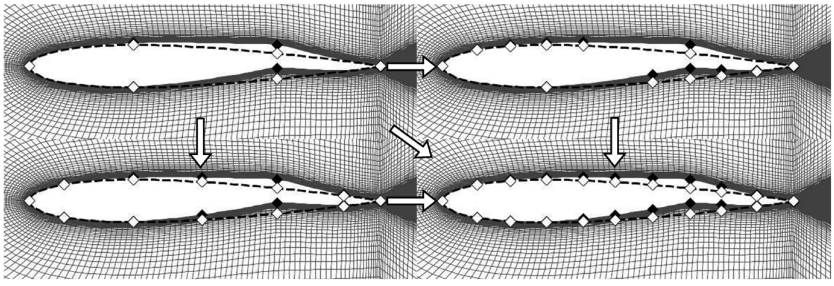

Fig. 16 Addition of new pilot points with the preservation of a deformed airfoil geometry; left/upper: initial set of pilot points for a deformed airfoil, others: addition of pilot points, white points: initial locations of pilot points, and black points: displaced pilot points.

the maximization of EI is good criteria to decide the location of new sample points, and it is difficult to be maximized by using any kind of optimizers. The EI value is zero in the major field of design space, and has many steep peaks, especially with a smaller estimated objective function. By including the minimization of estimated objective function, the optimization search can automatically proceed toward the regions of smaller estimated objective function. This property can increase the possibility finding the steep peaks of EI. The multi-objective optimization problem is solved by a multi-objective genetic algorithm [2]. The whole optimization procedures are depicted schematically in Fig. 18.

B. Parameterizations

The geometry parameterization and the structured mesh deformation are treated by the physics-based FFD method simultaneously. The control box is defined, as shown in Fig. 19, which includes the main wing of the DLR-F6 configuration. In this design problem, two parameterization levels are defined. In the first (coarser) case, six pilot points are defined on the leading and trailing edges at the 20%, 40% (kink), and tip sections of the main wing, as shown in Fig. 19. The pilot points at the 20% semispan section are specified to fix the wing geometry at their locations. The displacements for x and z (main flow and vertical) directions of the other four pilot points are defined as design variables. Therefore, the semispan length of the main wing is fixed within the optimization. In addition, the wing volume is also forced to be fixed within the optimization by applying the volume preservation technique discussed in Sec. IV.F. The aerodynamic angle of attack is also defined as another design variable, so the number of design variables in the coarser parameterization is nine in total. In the second (finer) parameterization, additional pilot points are defined at the 60 and 80% sections in the same manner as the coarser one, as shown in Fig. 20. The number of design variables is 17 in the finer level. The typical computational time for both geometry parameterization and mesh deformation is within 1 min in the *skip* mode, and it is over 1 h in the *full* mode (see Sec. IV.C for the italic terms).

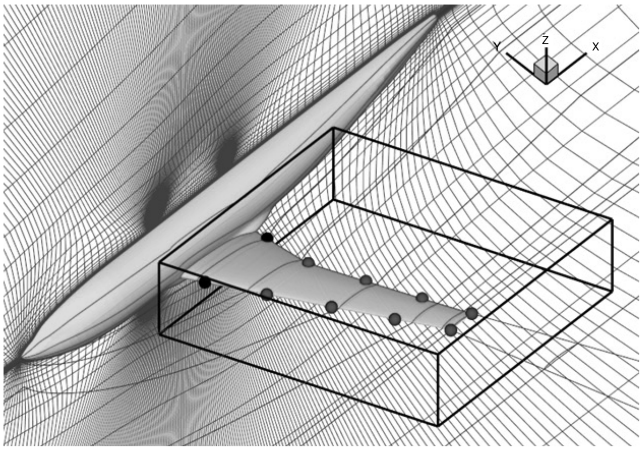

Fig. 20 Finer parameterization settings for DLR-F6 configuration.

C. Results and Discussion

A set of 81 initial sample points was generated in the coarser level by LHS and was evaluated to construct the initial kriging model. Then 80 new sample points were searched and evaluated iteratively. After the convergence of the objective function, the degree elevation from the coarser to finer level was executed. The information of all previous sample points is reconstructed for the finer parameterization. Then additional optimization iterations were executed in the finer parameterization level. The hierarchical optimization history is indicated in Fig. 21. It was confirmed that drag was reduced by the optimization in the coarser parameterization, and then further drag reduction was achieved in the finer parameterization level. In Table 3, those performances are summarized with the results of a drag decomposition analysis [45]. In Fig. 22, the geometries of the two optimal designs are compared with the DLR-F6 configuration. The increase of the sweep angle of the main wing can be confirmed in both optimal designs, and it is considered to reduce the wave drag. The increase of the dihedral angle at the wingtip can be also observed,

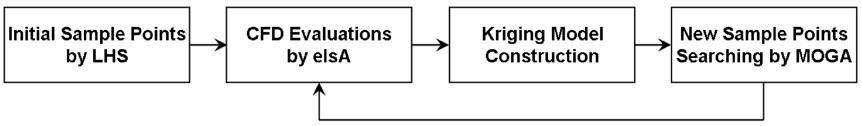

Fig. 18 Optimization flowchart.

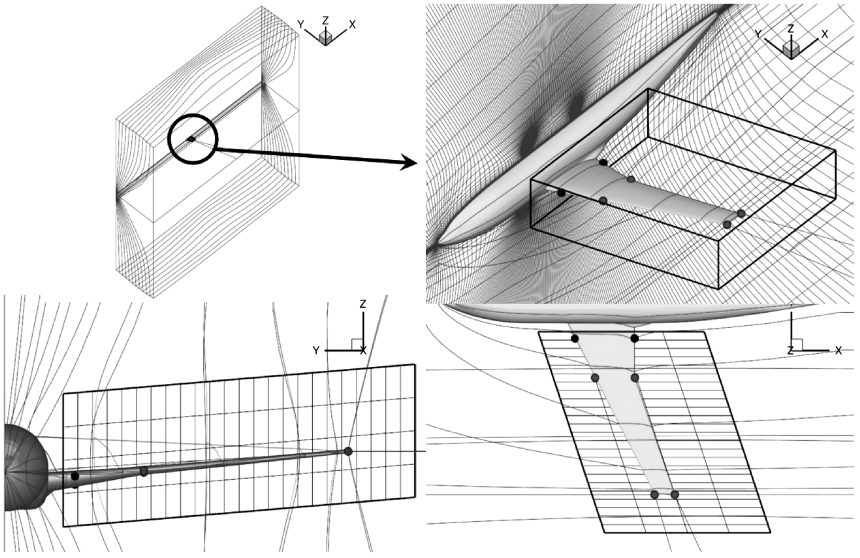

Fig. 19 Coarser parameterization settings for DLR-F6 configuration.

Table 3 Performances of DLR-F6 and two optimal designs (1 ct = 1 × 10^{-4})

	Wing volume	C_L	C_D, cts	$C_{D\text{wave}}$, cts	$C_{D\text{viscous}}$, cts	$C_{D\text{induced}}$, cts
DLR-F6	1.00	0.50	310.6	10.7	195.6	94.8
Coarser optimal	1.01	0.50	285.0	1.7	182.3	90.7
Finer optimal	1.00	0.50	281.3	0.8	180.5	90.4

and it is considered to reduce the induced drag as the effect of a winglet. The optimization in the finer level gave smoother geometry than that in the coarser level. In Fig. 23, the shock regions are visualized for the initial and two optimal designs. In Fig. 24, the surface-pressure coefficients at 66% semispan section are also compared. The reduction of shock waves on the outboard wing is remarkable in both optimal designs, and it contributed to the drag reduction achieved in these optimizations.

In this research, the two parameterization levels were chosen by human user, judging from the aerodynamic knowledge and from the result of optimization on coarser levels. Since the significance of each design variable can be estimated by analysis of variance with the kriging surrogate model [3], an automatic adaptive parameterization can be envisaged, which adds new design variables based on the observed significance of each design variable on coarser levels.

VI. Conclusions

In this research, the development of physics-based direct-manipulation free-form deformation and radial-basis-function-based deformation methods was described. With these methods, users define a control box or a set of centers surrounding an object to deform. Then users can define pilot points at any location on the object surface (actually anywhere in the volume) where displacements are imposed. The displacements of these pilot points are then naturally used as design variables for design optimization problems.

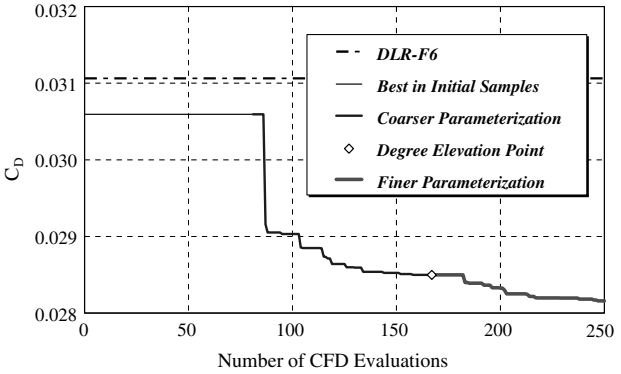

Fig. 21 Hierarchical optimization history.

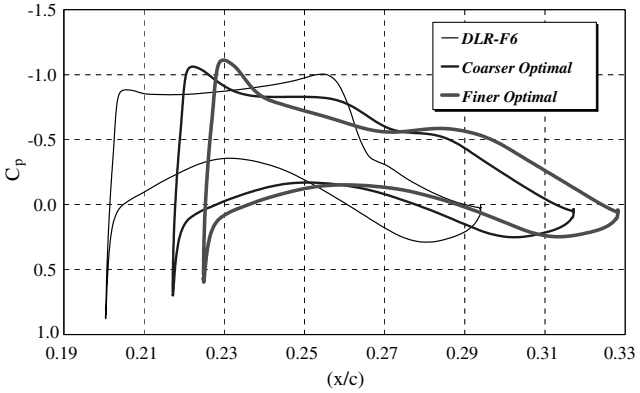

Fig. 24 Comparison of c_p distributions at 66% semispan section.

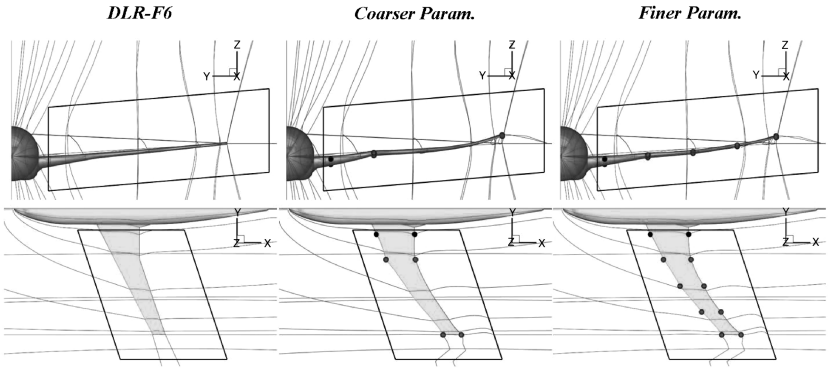

Fig. 22 Geometrical comparison of optimal designs.

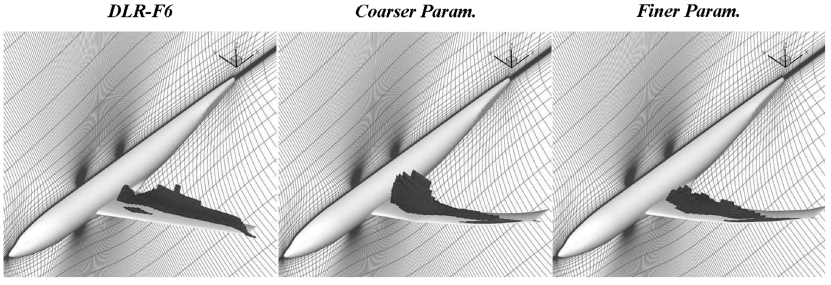

Fig. 23 Shock wave visualization of optimal designs.

The new locations of the control points/centers are automatically determined both to satisfy the user-specified deformations on the pilot points and to minimize a deformation energy, so that intuitive geometry deformation as real elastic objects is realized. In addition, these methods have the capability to deform, together with the geometry, any computational meshes mapped on the geometry, while preserving most of its mesh quality.

The developed methods were applied to several geometry manipulation and mesh deformation problems, and their efficiency for both geometry parameterization and mesh deformation was clearly demonstrated. These methods could be easily applied to arbitrary objects and mesh topologies. They also enable the analytical evaluation of mesh sensitivity, which is useful for adjoint-based optimizations. The meshes deformed by these methods exhibited better quality than a spring-analogy-based method. Furthermore, the developed methods allow enriching the design space easily, and this capability was successfully applied to the hierarchical design optimization of a 3-D wing/body configuration.

For efficient and detailed design optimizations, the physics-based FFD/RBF methods can bring significant benefits. The developed methods will be further improved and validated in the ongoing research project Développement Technique Probatoire (DTP) Optimization 2, conducted in collaboration with Airbus.

References

[1] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285

[2] Fonseca, C. M., and Fleming, P. J., "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423.

[3] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492. doi:10.1023/A:1008306431147

[4] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412. doi:10.2514/3.58379

[5] Sobieczky, H., "Parametric Airfoils and Wings," *Notes on Numerical Fluid Mechanics*, Vol. 68, 1998, pp. 71–88.

[6] Desideri, J. A., "Hierarchical Optimum Shape Algorithms Using Embedded Bézier Parameterizations," *Numerical Methods for Scientific Computing, Variational Problems and Applications*, Barcelona, 2003.

[7] Desideri, J. A., Abou el majd, B., and Janka, A., "Nested and Self-Adaptive Bézier Parameterizations for Shape Optimization," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 117–131. doi:10.1016/j.jcp.2006.12.016

[8] Painchaud-Ouellet, S., Tribes, C., Trepanier, J.-Y., and Pelletier, D., "Airfoil Shape Optimization Using a Nonuniform Rational B-Splines Parameterization Under Thickness Constraint," *AIAA Journal*, Vol. 44, No. 10, 2006, pp. 2170–2178. doi:10.2514/1.15117

[9] Carrier, G., and Blondeau, C., "DTP Optimisation, Optimisation Multidisciplinaire du Winglet de la Voilure d'un Avion de Transport," ONERA Rept. RT 7/08657 DAAP, Meudon, France, 2006.

[10] Nemec, M., Aftosmis, M. J., and Pulliam, T. H., "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method," AIAA Paper 2004-0113, 2004.

[11] Jameson, A., Shankaran, S., Martinelli, L., and Haimes, B., "Aerodynamic Shape Optimization of Complete Aircraft Configuration," AIAA Paper 2004-0533, 2004.

[12] Kleinveld, S., Roge, G., Daumas, L., and Dinh, Q., "Differentiated Parametric CAD Used within the Context of Automatic Aerodynamic Design Optimization," AIAA Paper 2008-5952, 2008.

[13] Kulfan, B. M., and Bussoletti, J. E., "Fundamental Parametric Geometry Representations for Aircraft Component Shapes," AIAA Paper 2006-6948, 2006.

[14] Kulfan, B. M., "Universal Parametric Geometry Representation Method," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158. doi:10.2514/1.29958

[15] Samareh, J. A., "Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD)," AIAA Paper 2000-4911, 2000.

[16] Desideri, J. A., and Janka, A., "Multilevel Shape Parameterization for Aerodynamic Optimization—Application to Drag and Noise Reduction of Transonic/Supersonic Business Jet," *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Jyväskyla, 2004.

[17] Samareh, J. A., "Aerodynamic Shape Optimization Based on Free-Form Deformation," AIAA Paper 2004-4630, 2004.

[18] Majd B. A. E., Duvigneau, R., and Desideri, J. A., "Aerodynamic Shape Optimization Using a Full and Adaptive Multilevel Algorithm," *European Research Community on Flow Turbulence and Combustion (ERCOFTAC 2006)*, Gran Canaria, Spain, 2006.

[19] Duvigneau, R., "Adaptive Parameterization Using Free-Form Deformation for Aerodynamic Shape Optimization," Institut National de Recherche en Informatique et en Automatique, Rept. 5949, Sophia Antipolis, France, 2006.

[20] Anderson, W. K., Karman, S. L., and Burdyshaw, C., "Geometry Parameterization Method for Multidisciplinary Applications," *AIAA Journal*, Vol. 47, No. 6, 2009, pp. 1568–1578. doi:10.2514/1.41101

[21] Rendall, T. C. S., and Allen, C. B., "Unified Fluid-Structure Interpolation and Mesh Motion Using Radial Basis Functions," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559. doi:10.1002/nme.2219

[22] Morris, A. M., Allen, C. B., and Rendall, T. C. S., "Domain-Element Method for Aerodynamic Shape Optimization Applied to a Modern Transport Wing," *AIAA Journal*, Vol. 47, No. 7, 2009, pp. 1647–1659. doi:10.2514/1.39382

[23] Jameson, A., "Efficient Aerodynamic Shape Optimization," AIAA Paper 2004-4369, 2004.

[24] Samareh, J. A., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 877–884. doi:10.2514/2.1391

[25] Wu, H. Y., Yang, S., Liu, F., and Tsai, H. M., "Comparison of Three Geometric Representations of Airfoils for Aerodynamic Optimization," AIAA Paper 2003-4095, 2003.

[26] Song, W., and Keane, A. J., "A Study of Shape Parameterisation Methods for Airfoil Optimization," AIAA Paper 2004-4482, 2004.

[27] Castonguay, P., and Nadarajah, S. K., "Effect of Shape Parameterization on Aerodynamic Shape Optimization," AIAA Paper 2007-0059, 2007.

[28] Mousavi, A., Castonguay, P., and Nadarajah, S. K., "Survey of Shape Parameterization Techniques and Its Effect on Three-Dimensional Aerodynamic Shape Optimization," AIAA Paper 2007-3837, 2007.

[29] Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis," *AIAA Journal*, Vol. 29, No. 3, 1991, pp. 327–333. doi:10.2514/3.10583

[30] Crumpton, P. I., and Giles, M. B., "Implicit Time-Accurate Solutions on Unstructured Dynamic Grids," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 11, 1997, pp. 1285–1300. doi:10.1002/(SICI)1097-0363(19971215)25:11<1285::AID-FLD607>3.0.CO;2-M

[31] Murayama, M., Nakahashi, K., and Matsushima, K., "A Robust Method for Unstructured Volume/Surface Mesh Movement," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 46, No. 152, 2003, pp. 104–112.

[32] Yang, Z., and Mavriplis, D. J., "Unstructured Dynamic Meshes with Higher-order Time integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2005-1222, 2005.

[33] Yang, Z., and Mavriplis, D. J., "Mesh Deformation Strategy Optimized by the Adjoint Method on Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 12, 2007, pp. 2885–2896. doi:10.2514/1.30592

[34] Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160. doi:10.1145/15886.15903

[35] Hsu, W. M., Hughes, J. F., and Kaufman, H., "Direct Manipulation of Free-Form Deformations," *Computer Graphics*, Vol. 26, No. 2, 1992, pp. 177–184. doi:10.1145/142920.134036

[36] Hu, S. M., Zhang, H., Tai, C. L., and Sun, J. G., "Direct Manipulation of FFD: Efficient Explicit Solutions and Decomposible Multiple Point Constraints," *The Visual Computer*, Vol. 17, No. 6, 2001, pp. 370–379.

[37] Pourazady, M., and Xu, X., "Direct Manipulations of B-spline and NURBS Curves," *Advances in Engineering Software*, Vol. 31, No. 2, 2000, pp. 107–118.
doi:10.1016/S0965-9978(99)00026-5

[38] Pourazady, M., and Xu, X., "Direct Manipulations of NURBS Surfaces Subjected to Geometric Constraints," *Computers & Graphics*, Vol. 30, No. 4, 2006, pp. 598–609.

[39] Nielsen, E. J., and Park, M. A., "Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948–953.
doi:10.2514/1.16052

[40] Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *SIGGRAPH '90*, Computer Graphics, Vol. 24, No. 4, 1990, pp. 187–196.

[41] Gazaix, M., Jolles, A., and Lazareff, M., "The elsA Object-Oriented Computational Tool for Industrials Applications," 23rd Congress of ICAS, Paper 2002-1.10.3.1, Toronto, 2002.

[42] Cambier, L., and Veuillot, J. P., "Status of the elsA CFD Software for Flow Simulation and Multidisciplinary Applications," AIAA Paper 2008-664, 2008.

[43] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, 1992.

[44] Laflin, K. R., "AIAA CFD Drag Prediction Workshop: An Overview," 25th Congress of ICAS, Paper 2006-2.4.1.1, Hamburg, Germany, 2006.

[45] Esquieu, S., "Reliable Drag Extraction from Numerical Solutions: Elimination of Spurious Drag," *Computational Uncertainty in Military Vehicle Design*, RTO-MP-AVT-147, NATO Research and Technology Organisation, Paper 42, Athens, 2007.

A. Messac
Associate Editor