

Two-Level Free-Form Deformation for High-Fidelity Aerodynamic Shape Optimization

Hugo Gagnon* and David W. Zingg†

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

A general and unified geometry framework based on non-uniform rational B-splines is presented. One key aspect of this framework is the use of free-form deformation where the control points defining the geometry itself are embedded rather than the usual surface grid points. This ensures that one maintains an analytical representation of the geometry as it deforms and, if required, also enables the use of an integrated algorithm for mesh movement. The flexibility and robustness of the proposed approach are demonstrated in the context of aerodynamic shape optimization by maximizing, using an Euler-based flow solver and gradient-based optimizer, the lift-to-drag ratio of an initially half-sphere-shaped geometry. Sensitivities are computed analytically via the discrete adjoint method.

I. Introduction

ANTHROPOGENIC climate change has awakened global awareness.¹ Its disastrous impacts on the environment have long been linked to oil, whose diminishing world reserves have led to a substantial rise in fuel prices. As far as the aviation industry is concerned, greener alternatives to kerosene will remain uncertain in the foreseeable future. Therefore, other means of reducing greenhouse gas emissions at reasonable cost must be sought.² One such prospect is through improved fuel efficiency via drag and weight reduction aided by aerodynamic shape optimization³ (ASO) and multidisciplinary design optimization (MDO), due to their potential for discovering novel, *optimal* aircraft configurations.

Until recently, high-fidelity ASO and MDO have been almost exclusively restricted to relatively simple geometries,^{4,5} e.g. airfoils and wings. Indeed, efforts in these fields have largely focused on accurately and efficiently evaluating aerodynamic functionals and their gradients with respect to a set of design variables.⁶ Instead, this paper addresses the fundamental need for an efficient shape control system capable of handling large surface deformations as well as accompanying volume mesh movement. The proposed methodology builds directly on the work of Hicken and Zingg,^{7,8} in view of incorporating an high-fidelity finite-element structural solver in the near future.

Historically, shape control systems for ASO have been subdivided into two broad categories: CAD-based and CAD-free.⁹ On the one hand, CAD-based systems usually benefit from well-developed and extensive toolkits while, on the other hand, due to their proprietary nature they often veil their core components from the user, making it difficult to compute, for example, sensitivities through the adjoint formulation,¹⁰ even with interfaces such as CAPRI.¹¹ Furthermore, when parallel considerations are accounted for, other practical issues such as the number of available CAD licenses may become problematic.¹²

In opting for a CAD-free system, one must also carefully choose a parameterization technique. A good parameterization is one that can represent a broad range of useful shapes using as few (design) variables as possible. Again, the developer is presented with two families of options: featured-based or free-form.¹³ While featured-based parameterizations, such as the CST method of Kulfan,¹⁴ are well suited to improving existing component-based aircraft, they generally assume prescribed geometries built from, but limited to, human imagination. Since this project is partly aimed at uncovering yet unknown unconventional configurations, a more flexible parameterization is required.

*PhD Candidate and AIAA Student Member

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics and Environmentally Friendly Aircraft Design, J. Armand Bombardier Foundation Chair in Aerospace Flight, AIAA Associate Fellow

Free-form parameterizations based on analytical bumps¹⁵ (Hicks and Henne, 1978), discrete points¹⁶ (Jameson and Reuther, 1994), and more recently B-splines¹⁷ (Nemec and Zingg, 2002) have all had their share of success. In the case of B-splines, their high flexibility and efficient approximation power¹⁸ still make them prime candidates when no a priori knowledge of the final geometry is known, even though continuity requirements at their boundaries can be fairly complicated.¹⁹ It is important to realize that, being surface-based parameterizations, none of these techniques can readily accommodate structural components.

In contrast, volume-based parameterizations can intrinsically account for any number of grids irrespective of their disciplines. Jakobsson and Amoignon²⁰ and later Morris et al.²¹ used radial basis functions to parameterize full computational domains, including the outer mold line of geometries, thereby perturbing both surface and internal grids in one shot. Other volumetric methods, mostly borrowed from the soft object animation industry, have also recently gained in popularity. These include cage-based deformation²² and free-form deformation²³ (FFD); see Ref. 24 for a comprehensive review of direct manipulation methods based on FFD and also Ref. 25 for applications to aerospace design.

Due to their attractive features, it is not surprising that a number of researchers have adopted some of these newer modeling techniques for ASO^{26–29} and MDO.^{30,31} In their work, however, it is presumed that a final geometry will have to undergo a reverse-engineering fit as a post-processing step (for say, manufacturing purposes), in which case geometric fidelity will invariably be lost. This unfortunate price stems from the fact that CAD-free, unlike CAD-based, shape optimization only requires an initial discrete geometry to operate. By parameterizing only the relevant information of a geometric model, it should be possible to reconcile the two worlds.

For these and other reasons we hope to clarify in this paper, a new CAD-free geometry framework has been developed. It is composed of a robust two-level FFD scheme that tightly integrates both surface warping and resulting volume mesh movement, all the while retaining exact geometric representation. Following recommendations from the NASA-IGES NURBS-Only standard,³² the entire framework is built onto non-uniform rational B-splines³³ (NURBS) objects.

The paper is divided as follows. Section II reviews essential concepts pertaining to NURBS. The ideas involved in FFD are introduced in the first half of Section III before presenting the novel two-level approach in the second half. Its coupling with a previously developed in-house aerodynamic optimizer is then briefly discussed in Section IV. Finally, the results of a gradient-based optimization example are the subject of Section V and concluding remarks are given Section VI.

II. NURBS Concepts

A NURBS curve of degree p is defined as

$$\mathbf{C}(\xi) = \sum_{i=0}^N \mathcal{R}_i^{(p)}(\xi) \mathbf{B}_i, \quad a \leq \xi \leq b. \quad (1)$$

A designer is free to choose the number, $N + 1$, and location of the control points $\{\mathbf{B}_i \in \mathbb{R}^3, i = 0, \dots, N\}$, as well as the degree, p , and knot vector,

$$\Xi = \{\underbrace{a, \dots, a}_{p+1}, \xi_{p+1}, \dots, \xi_N, \underbrace{b, \dots, b}_{p+1}\},$$

with which the rational basis functions,

$$\mathcal{R}_i^{(p)}(\xi) = \frac{\mathcal{N}_i^{(p)}(\xi) w_i}{\sum_{j=0}^N \mathcal{N}_j^{(p)}(\xi) w_j},$$

are defined. These basis are themselves composed of user-defined scalar weights $\{w_i\}_{i=0}^N$ and p th-degree B-spline functions $\{\mathcal{N}_i^{(p)}(\xi)\}_{i=0}^N$. The piecewise polynomials making up these splines are joined at the non-uniform and non-decreasing knot locations $\{\xi_i\}_{i=0}^{N+p+1}$ introduced above, such that the basis is C^{p-k} continuous at a given knot, where k is its multiplicity. Note that we choose a non-periodic knot vector, i.e. with end knot multiplicities equal to the order of the splines, which ensures that \mathbf{C} passes exactly through its end points \mathbf{B}_1 and \mathbf{B}_N . This property becomes immediately apparent after examining the recursive

relationships defining B-splines:

$$\mathcal{N}_i^{(0)}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathcal{N}_i^{(p)}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \mathcal{N}_i^{(p-1)}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \mathcal{N}_{i+1}^{(p-1)}(\xi).$$

Similarly, NURBS surfaces (also called patches) and volumes can be created using tensor products. For example, a NURBS volume mapping a cubic domain to \mathbb{R}^4 reads

$$\mathbf{V}^w(\boldsymbol{\xi}) = \sum_{i=0}^{N_i} \sum_{j=0}^{N_j} \sum_{k=0}^{N_k} \mathcal{N}_i^{(p_i)}(\xi) \mathcal{N}_j^{(p_j)}(\eta) \mathcal{N}_k^{(p_k)}(\zeta) \mathbf{B}_{i,j,k}^w, \quad (2)$$

where here homogeneous weighted coordinates $\mathbf{B}_{i,j,k}^w = (wx, wy, wz, w)_{i,j,k}$ are used to offer a non-rational, compact analogue to Eq. (1) that is considerably easier to implement. The indices p_i and N_i respectively refer to the degree and last control point in the ξ direction; similar definitions hold for the η and ζ directions. Projecting $\mathbf{V}^w(\boldsymbol{\xi})$ back onto the $w = 1$ plane retrieves the desired three-dimensional volume $\mathbf{V}(\boldsymbol{\xi})$. For the special case of $w_{i,j,k} = 1$, it can be shown that $\mathbf{V}^w(\boldsymbol{\xi})$ simply reduces to $\mathbf{V}(\boldsymbol{\xi})$, i.e. to the traditional B-spline volume definition. Thus, without loss of generality, all weights are now assumed to be unity, and the superscript w is dropped in the remainder of the manuscript.

III. Free-Form Deformation

Once an initial geometry has been defined with a network of watertight NURBS patches, it is customary in ASO to group some or all of their control points to form so-called feature-based design variables, e.g. based on span, dihedral, twist, etc. However, as discussed in Section I, it is often preferable to formulate the problem in a more general way, i.e. in a free-form fashion, or a combination thereof. This is precisely what FFD is capable of.

A. Definition

Conceptually, FFD²³ is best visualized as embedding a flexible, rubber-like object into a transparent material having the same constitutive properties. As the larger block deforms, so will the embedded object. In practice, any parametric representation can be utilized as the embedding material; here, the trivariate NURBS volumes expressed by Eq. (2) are used throughout this work. For this reason, the terms ‘‘NURBS volume’’ and ‘‘FFD volume’’ will now be used interchangeably.

Formally, FFD can be formulated as a mapping $\mathcal{F} : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \rightarrow \mathbb{R}^3$, from world space, \mathbf{t} , to parametric space, $\boldsymbol{\xi}$, to deformed world space, $\tilde{\mathbf{t}}$. This is achieved by two functions. The first one is the embedding function $F^{-1}(\mathbf{t}) = \boldsymbol{\xi}$, which associates a parametric value $\boldsymbol{\xi}$ to each vertex \mathbf{t} of an object. This is normally carried out by a Newton search procedure and needs only be performed once. Note that here the embedded vertices are taken to be the surface control points that define the object, not the surface points that discretize it as is usually the case. An obvious advantage of this approach is, assuming a geometry was initially described by NURBS entities only (or in fact any of its simplifications, e.g. Bézier, B-splines, etc.), then its analytical definition will always remain intact. The second function is the deformation function $\tilde{F}(\boldsymbol{\xi}) = \tilde{\mathbf{t}}$, which translates to simply re-evaluating Eq. (2) for each embedded vertex once the FFD volume’s control net $\{\mathbf{B}_{i,j,k}\}$, or lattice, has deformed. The composition of F^{-1} and \tilde{F} is what constitutes FFD:

$$\mathcal{F}(\tilde{\mathbf{t}}, \mathbf{t}) = \tilde{F}(F^{-1}(\mathbf{t})) - \tilde{\mathbf{t}} = \tilde{F}(\boldsymbol{\xi}) - \tilde{\mathbf{t}} = \mathbf{0}.$$

In the context of ASO, NURBS-based FFD offers many advantages. An obvious yet fundamental one is that a given geometry can be parameterized to machine accuracy provided the inverse mapping search is tightly converged. This remains true whether a geometry is available in discrete form, such as in a surface triangulation, or in analytical form, such as with NURBS patches. Either way, FFD allows simple sets of geometric design variables $\mathbf{v}_{\text{geo}} \in \{\mathbf{B}_{i,j,k}\}$ to manipulate complex geometries. For example, more than one FFD volume can be used simultaneously to optimize independent portions of an aircraft. Moreover, each

FFD volume is capable of both local and global deformation, e.g. an isolated bump on the surface of a wing can be produced by translating an individual FFD control point and large-scale twisting effects can be achieved by rotating entire slices of $\{\mathbf{B}_{i,j,k}\}$. Indeed, technically an entire wing can be twisted using only 8 control points, a difficult task for a B-spline surface parameterization alone where then the coupling of control points can become involved, especially when boundary requirements, such as geometric continuity along the leading edge, must also be accounted for. This capability of FFD to intrinsically retain topological features ensures continuity of the design space, an important property when working with gradient-based optimizers. This is also true for MDO applications where structural components can be simply embedded alongside a geometry without additional special treatment. Finally but just as importantly, the sensitivities of the embedded control points to the ones of the FFD lattice are exact and, as seen from Eq. (2), are easily derived:

$$\frac{\partial \mathbf{V}(\boldsymbol{\xi})}{\partial \mathbf{B}_{i,j,k}} = \mathcal{N}_i(\xi) \mathcal{N}_j(\eta) \mathcal{N}_k(\zeta) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

B. Two-Level Approach

In order to account for both geometry changes and resulting volume mesh movement, a two-level FFD scheme is introduced. At the first level, usually one FFD volume controls the surface control points of the underlying geometry, while, at the second level, more FFD volumes — as many as there are blocks decomposing a structured domain — control the computational grid points. The two levels of FFD volumes are connected by the surface control points embedded in the FFD volume(s) of the first level.

1. Control-point-based Surface Warping

In contrast to the work of others,^{26–28,30,31,34} here the *control points* of the NURBS surfaces defining the underlying geometry are embedded inside an FFD volume rather than the discrete grid points lying on those surfaces. This way the exact geometry representation is never lost. As previously described, a geometry can then be modified by perturbing either individual or groups of control points pertaining to a lattice \mathbf{B} , a task normally performed by the optimizer.

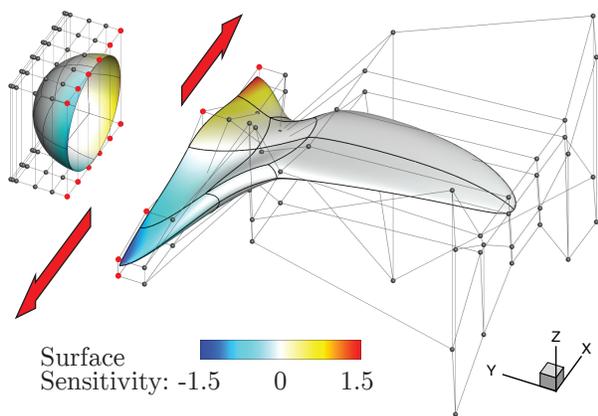


Figure 1. Control-point-based surface warping: from sphere to blended-wing-body. See text for an explanation of the red arrows. The embedded surface control points are not shown for improved clarity.

property of B-splines. Finally, observe how geometric continuity is naturally preserved at the boundaries of the deformed patches, without requiring any constraint whatsoever.

While FFD volumes are extremely malleable, the embedded objects they contain are not exempt from self-intersection(s). Such would be the case should an FFD volume’s control net become heavily intertwined. In order to prevent this, a simple injectivity test,³⁵ which amounts to ensuring the determinants of the FFD Jacobian are strictly positive, can be optionally carried out following each deformation \mathcal{F}_1 . Upon failure, a gradient-based optimizer could backtrack and reduce the current step-length or, if possible, try another design direction.

In particular, if \mathbf{A} denotes an xyz -coordinate block-column vector corresponding to all the control points that define a geometry, and \mathbf{A}_{emb} a portion of that vector matching the embedded control points only, then control-point-based surface warping can be succinctly expressed as

$$\mathcal{F}_1(\mathbf{A}_{\text{emb}}, \mathbf{A}_{\text{emb}}^{(0)}) = \mathbf{0}, \quad (4)$$

where the superscript 0 refers to the state preceding deformation. For example, Figure 1 on the left depicts a sphere morphed into a blended-wing-body. In that particular case, all surface control points are embedded, i.e. $\mathbf{A}_{\text{emb}} \equiv \mathbf{A}$. The sensitivity of each discrete surface node’s x coordinate to a global X -directional scaling variable (to be described in Section V) represented by the red arrows and associated to the highlighted control points is also shown. Note the local effect of that variable, a result of the local support

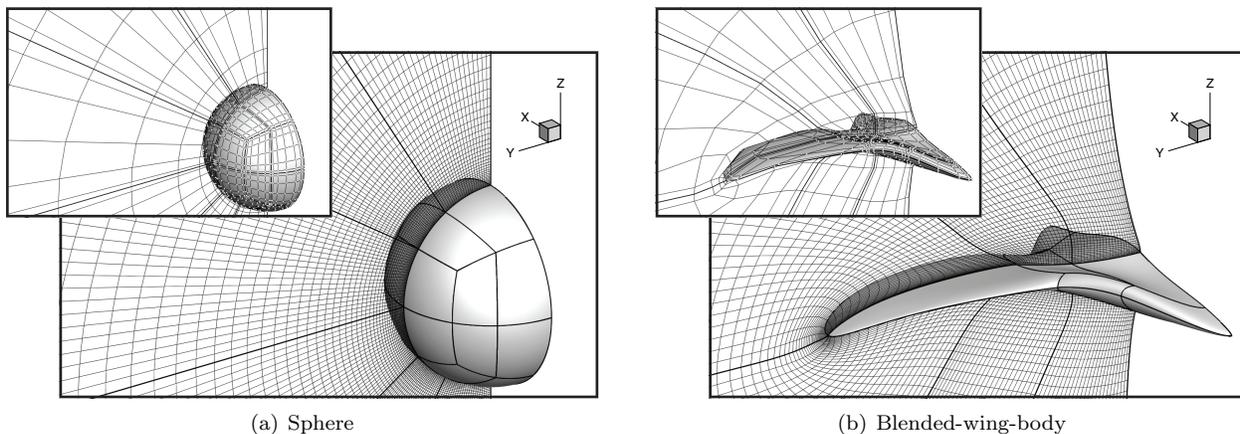


Figure 2. Grid-point-based volume mesh movement: (a) undeformed and (b) deformed grids. Control grids including surface control points are shown as insets.

2. Grid-point-based Volume Mesh Movement

Following Hicken and Zingg,⁷ each block in a three-dimensional structured domain is assigned a NURBS volume whose global control net acts as a coarse grid $\mathbf{b} \supset \mathbf{A} \supseteq \mathbf{A}_{\text{emb}}$,^a where here the symbol \supset denotes a strict superset while \supseteq implies the possibility $\mathbf{A}_{\text{emb}} \equiv \mathbf{A}$. Once again, these volumes can be regarded as FFD volumes where this time around the full set of volume *grid points* \mathbf{x} are embedded inside them;

$$\mathcal{F}_2(\mathbf{x}, \mathbf{x}^{(0)}) = \mathbf{0}, \quad (5)$$

either directly by discretizing the parametric domain of each NURBS volume³⁶ or, as performed in Ref. 7, by fitting B-spline volumes to an already existing multi-block mesh. Note that either approaches are as equally applicable to structured than unstructured grids.

In response to changes in \mathbf{A}_{emb} resulting from \mathcal{F}_1 , the control points \mathbf{b} of this second level of FFD volumes are updated according to the equations of linear elasticity:³⁷

$$\mathcal{M}(\mathbf{b}, \mathbf{b}^{(0)}) = \mathbf{K}\mathbf{u} - \mathbf{f} = \mathbf{0}, \quad (6)$$

where $\mathbf{K} = \mathbf{K}(\mathbf{b}^{(0)})$ is the stiffness matrix, \mathbf{f} is the block-column force vector (implicitly defined by boundary alterations $\Delta\mathbf{A}$) and $\mathbf{u} = \mathbf{b} - \mathbf{b}^{(0)}$ is the block-column displacement vector of the B-spline volume control point coordinates. After solving these, the entire computational mesh can be regenerated algebraically from the displaced control mesh \mathbf{b} through \mathcal{F}_2 . This last operation is almost instantaneous thanks to a very efficient implementation of de Boor's algorithm,¹⁸ and since there are typically two or three orders of magnitude fewer control points \mathbf{b} than grid points \mathbf{x} , the resources spent to converge Eq. (6) with a preconditioned conjugate gradient solver are only a few percent of what a grid-point-based deformation would cost.⁷

To be clear, the proposed two-level approach relies on a one-to-one correspondence between those control points in \mathbf{b} embracing the geometry with those in \mathbf{A} . Formally, if \mathbf{b} is ordered such that all internal control points are first, followed by non-surface and finally surface boundary points, \mathbf{A} can then be defined by the restriction operation $\mathbf{A} = [0 \ 0 \ \mathbf{I}]\mathbf{b}$, where the identity matrix \mathbf{I} is of size $3N_s \times 3N_s$, N_s being the total number of surface control points. Mesh movement is thus tightly integrated with geometry parameterization. Recall that the latter is itself constrained by the movement of the FFD lattice \mathbf{B} by virtue of Eq. (4).

Two stiffening mechanisms are enforced through \mathbf{K} in order to preserve mesh quality of the control mesh \mathbf{b} , which impacts the quality of the actual mesh \mathbf{x} . The first one allows the Young's modulus to vary spatially according to a metric based on cell orthogonality, while the second one helps smaller elements maintain their shape. Furthermore, a negative Poisson's ratio of 0.2 is chosen with the aim of further preventing the production of disproportionately high cell aspect ratios.

^aThe work presented in this section is only a slight generalization of that conducted by the cited authors. Only a brief description is given here for completeness.

The elasticity-based model is however only valid under a small-strain assumption. If shape changes are sufficiently large, the mesh movement is generally required to be broken up into m increments:

$$\mathbf{A}^{(i)} = \frac{i}{m} \left(\mathbf{A}^{(m)} - \mathbf{A}^{(0)} \right) + \mathbf{A}^{(0)}, \quad i = 1, 2, \dots, m. \quad (7)$$

While this approach helps maintain linearity, it also has important consequences on the evaluation of the objective function's gradients,³⁷ discussed in Section IV.

An example of grid-point-based volume mesh movement applied to the same half-sphere of Figure 1 is depicted in Figure 2. First, mesh movement based on linear elasticity with $m = 6$ is applied to a coarse grid of FFD control points $\mathbf{b}^{(0)}$, yielding a perturbed control grid \mathbf{b} (insets). Then, assuming a computational mesh $\mathbf{x}^{(0)}$ was embedded into $\mathbf{b}^{(0)}$ prior to mesh movement, the new mesh \mathbf{x} can be regenerated from \mathbf{b} through Eq. (5). Notice how the control grid mimics the computational one, and as such how the overall mesh quality is maintained.

IV. Gradient-based Optimization Suite

Design spaces in two-dimensional ASO based on the Reynolds-averaged Navier-Stokes (RANS) equations are typically convex – and thus amenable to gradient-based optimizers – while in three dimensions their multimodality is common.³⁸ Gradient-free methods would then seem the best approach; however their computational cost when applied to high-fidelity, three-dimensional ASO is still prohibitive.³⁹ In that regard, surrogates and other efficient global optimization methods, such as those presented by Chernukin and Zingg,³⁸ offer a good compromise between speed and depth of design space exploration.

For now, because the main focus of this paper is to demonstrate the applicability of the proposed two-level FFD formulation, only a straightforward gradient-based optimization will be conducted in Section V. Before then, a short presentation on the evaluation of the objective function, \mathcal{J} , and its gradient with respect to a set of design variables, $d\mathcal{J}/d\mathbf{v}$, is given. We also present two test cases validating the aerodynamic optimizer. The involved algorithms are of course the fruit of many researchers, and the reader is encouraged to consult the cited literature for more details.

A. Function Evaluation

In order to determine the discrete objective function \mathcal{J} , for example drag or C_L/C_D , the flow field surrounding an aerodynamic surface must first be solved for. While RANS-based ASO⁴⁰ will be performed in the future, only the Euler governing equations are considered here. Once discretized in space with finite-difference SBP-SAT operators,⁴¹ these read $\frac{d\mathbf{q}}{dt} + \mathcal{R} = \mathbf{0}$, from which

$$\mathcal{R}(\mathbf{q}, \mathbf{b}^{(m)}) = \mathbf{0} \quad (8)$$

at steady-state. Here, \mathbf{q} is a block-column vector of the conservative flow variables, while the dependence of the nonlinear flow residuals \mathcal{R} on $\mathbf{b}^{(m)}$, which is kept constant throughout a flow solve, is through the grid metrics. A major advantage of SBP-SATs over other discretization methods is that mesh lines meeting at block interfaces are only required to be C^0 continuous there, thus reducing inter-process communication and also eliminating the need for special treatments of exceptional points.

If the implicit Euler time-marching method is applied with local time linearization to the semi-discrete equations, then we obtain

$$\left(\mathcal{A}^{(n)} + \frac{\mathbf{I}}{\Delta t^{(n)}} \right) \Delta \mathbf{q}^{(n)} = -\mathcal{R}^{(n)},$$

where \mathcal{A} is the flow Jacobian $\partial\mathcal{R}/\partial\mathbf{q}$, and \mathbf{I} the identity matrix. At each iteration n , this system of linear equations is solved inexactly by a parallel Schur-preconditioned Jacobian-free Krylov iterative method. For even better performance, the time-step Δt is varied spatially and also quickly increases at each n thus recovering the quadratically converging Newton's method.

B. Gradient Evaluation

Evaluating the objective function's sensitivities to the design variables, $d\mathcal{J}/d\mathbf{v}$, in an efficient manner is equally important. The design variables considered here are $\mathbf{v} = [\mathbf{v}_{\text{geo}}^T, \alpha]^T$, where \mathbf{v}_{geo} is the vector of

geometric design variables, taken to be the xyz -coordinates of individual FFD control points involved in \mathcal{F}_1 (or any variable that handles groups of them simultaneously), and α is the freestream angle-of-attack. Since the number of design variables involved in problems of interest in ASO is typically on the order of $O(10^2)$, a logical choice for evaluating the total derivative $d\mathcal{J}/d\mathbf{v}$ is the adjoint method.³ Indeed, unlike finite-differencing, the computational cost of the adjoint method is virtually independent of the size of \mathbf{v} .

As pointed out in Ref. 37, it is preferable to augment the traditional adjoint formulation with grid variables. Indeed, as seen from Eqs. (6) and (8), it is to be expected that \mathcal{J} not only depends on the design variables \mathbf{v} and the flow variables \mathbf{q} , but also on the final control grid nodes $\mathbf{b}^{(m)}$. Likewise, the mesh residuals $\{\mathcal{M}^{(i)}\}_{i=1}^m$ are functions of both the previous $\mathbf{b}^{(i-1)}$ and current $\mathbf{b}^{(i)}$ control grids, as well as the perturbed surface control points $\mathbf{A}^{(i)}$, which are themselves functions of $\mathbf{A}^{(m)}$ through Eq. (7) and \mathbf{v}_{geo} through \mathcal{F}_1 . Finally, the flow residuals \mathcal{R} also depend on \mathbf{v} , \mathbf{q} and $\mathbf{b}^{(m)}$. Note that by directly accounting for the control grids $\{\mathbf{b}^{(i)}\}_{i=1}^m$, the only remaining explicit dependence of \mathcal{J} , \mathcal{M} , or \mathcal{R} , on \mathbf{v} is through α .

Once the relative dependencies between state variables have been established, an elegant formulation to the adjoint problem is:³⁷

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(\mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)}), \\ \text{w.r.t.} \quad & \mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)}, \\ \text{s.t.} \quad & \mathcal{M}^{(i)}(\mathbf{A}^{(i)}(\mathbf{v}), \mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) = \mathbf{0}, \quad i = 1, 2, \dots, m, \\ & \mathcal{R}(\mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)}) = \mathbf{0}. \end{aligned}$$

Typical of such a constrained optimization problem, a Lagrangian,

$$\begin{aligned} \mathcal{L} &= \mathcal{L}(\boldsymbol{\lambda}^{(i)}, \boldsymbol{\psi}, \mathbf{q}, \mathbf{b}^{(i)}, \mathbf{v}), \quad i = 1, 2, \dots, m, \\ &= \mathcal{J}(\mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)}) + \sum_{i=1}^m \boldsymbol{\lambda}^{(i)T} \mathcal{M}^{(i)}(\mathbf{A}^{(i)}(\mathbf{v}), \mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) + \boldsymbol{\psi}^T \mathcal{R}(\mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)}), \end{aligned}$$

is introduced. At optimality, this Lagrangian must at least obey the first order Karush-Kuhn-Tucker⁴² (KKT) conditions. The first two, $\partial\mathcal{L}/\partial\boldsymbol{\lambda}^{(i)} = \mathbf{0}$, $i = 1, 2, \dots, m$, and $\partial\mathcal{L}/\partial\boldsymbol{\psi} = \mathbf{0}$, are satisfied provided Eqs. (6) and (8) are solved sequentially. The following two conditions, $\partial\mathcal{L}/\partial\mathbf{q} = \mathbf{0}$ and $\partial\mathcal{L}/\partial\mathbf{b}^{(i)} = \mathbf{0}$, $i = 1, 2, \dots, m$, respectively lead to the equations for the Lagrange multipliers $\boldsymbol{\psi}$ and $\{\boldsymbol{\lambda}^{(i)}\}_{i=1}^m$, also known as the adjoint variables. These must be solved in reverse order relative to the ordering used to calculate \mathcal{J} :

$$\begin{aligned} \left(\frac{\partial\mathcal{R}}{\partial\mathbf{q}}\right)^T \boldsymbol{\psi} &= -\left(\frac{\partial\mathcal{J}}{\partial\mathbf{q}}\right)^T, \\ \left(\frac{\partial\mathcal{M}^{(m)}}{\partial\mathbf{b}^{(m)}}\right)^T \boldsymbol{\lambda}^{(m)} &= -\left(\frac{\partial\mathcal{J}}{\partial\mathbf{b}^{(m)}}\right)^T - \left(\frac{\partial\mathcal{R}}{\partial\mathbf{b}^{(m)}}\right)^T \boldsymbol{\psi}, \\ \left(\frac{\partial\mathcal{M}^{(i)}}{\partial\mathbf{b}^{(i)}}\right)^T \boldsymbol{\lambda}^{(i)} &= -\left(\frac{\partial\mathcal{M}^{(i+1)}}{\partial\mathbf{b}^{(i)}}\right)^T \boldsymbol{\lambda}^{(i+1)}, \quad i = m-1, m-2, \dots, 1. \end{aligned} \tag{9}$$

Finally, assuming all first four conditions are solved to a small enough tolerance, it follows that $\mathcal{L} = \mathcal{J}$, so the final condition, $\partial\mathcal{L}/\partial\mathbf{v} = \mathbf{0}$, is the total derivative of \mathcal{J} with respect to \mathbf{v} , i.e. it is the desired gradient:

$$\frac{d\mathcal{J}}{d\mathbf{v}} = \frac{\partial\mathcal{J}}{\partial\mathbf{v}} + \sum_{i=1}^m \left(\boldsymbol{\lambda}^{(i)T} \frac{\partial\mathcal{M}^{(i)}}{\partial\mathbf{A}^{(i)}} \frac{\partial\mathbf{A}^{(i)}}{\partial\mathbf{A}^{(m)}} \frac{\partial\mathbf{A}^{(m)}}{\partial\mathbf{v}} \right) + \boldsymbol{\psi}^T \frac{\partial\mathcal{R}}{\partial\mathbf{v}}, \tag{10}$$

and is non-zero in general; it must be driven to zero by an optimizer. For that purpose, we use the optimization package SNOPT,⁴³ capable of handling thousands of simultaneous general constraints. SNOPT is based on the sequential-quadratic-programming paradigm and approximates the Hessian of its own Lagrangian using the quasi-Newton method of Broyden, Fletcher, Goldfarb and Shanno.⁴²

A detailed description of the specialized solution strategies employed for solving Eqs. (9) is given in Ref. 7. New to this paper are however the sensitivities of the final surface control points to the design

variables, $\partial \mathbf{A}^{(m)}/\partial \mathbf{v}$, found in the second term of Eq. (10). In general, these surface sensitivities can be further decomposed:

$$\frac{\partial \mathbf{A}^{(m)}}{\partial \mathbf{v}} = \frac{\partial \mathbf{A}^{(m)}}{\partial \mathbf{B}} \frac{\partial \mathbf{B}}{\partial \mathbf{v}},$$

where \mathbf{B} are the control points of an FFD lattice, and \mathbf{v} any set of large-scale design variables moving parts of \mathbf{B} together; examples of such variables are given in Section V. The term $\partial \mathbf{A}^{(m)}/\partial \mathbf{B}$ can be directly computed using Eq. (3), while in this work we calculate $\partial \mathbf{B}/\partial \mathbf{v}$ using the complex-step method.⁴⁴ Note that both terms stay constant for all mesh increments $i = 1, 2, \dots, m$, the former throughout an entire optimization run and the latter during the course of a gradient evaluation; thus, in practice, $\partial \mathbf{A}^{(m)}/\partial \mathbf{v}$ is pulled out from the summation seen in Eq. (10).

C. Validation Test Cases

The following two test cases are based on the half-sphere geometry enclosed by the FFD lattice of Figure 1 and on the grids of Figure 2(a). In both cases, the freestream Mach number and angle-of-attack are set to 0.5 and 0, respectively.

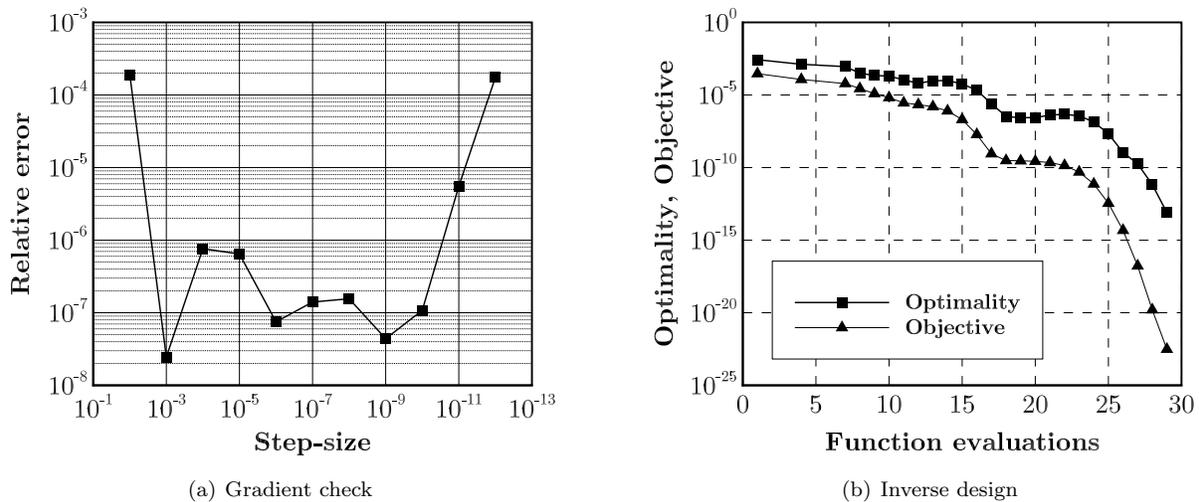


Figure 3. Validating the aerodynamic optimizer: (a) relative error between adjoint-based and finite-differenced directional derivative $D_{\mathbf{z}}\mathcal{J}$; (b) convergence history for the inverse design.

1. Gradient Check

As a means to verify gradient accuracy, the analytical directional derivative $D_{\mathbf{z}}\mathcal{J} = (\partial \mathcal{J}/\partial \mathbf{v}) \mathbf{z}$, where $\mathbf{z}_i = \text{sign}[(d\mathcal{J}/dv_i)]$, is well suited to compare against the centered, second-order finite-difference formula,⁷

$$D_{\mathbf{z}}\mathcal{J} \approx \frac{\mathcal{J}(\mathbf{v} + \epsilon \mathbf{z}) - \mathcal{J}(\mathbf{v} - \epsilon \mathbf{z})}{2\epsilon}.$$

The relative error between the two is plotted in Figure 3(a) over a range of step-sizes ϵ , for $\mathcal{J} = C_L/C_D$ and \mathbf{v} set to the design variables described in Section V. As expected, the truncation error of the finite-difference approximation decreases and then plateaus until round-off errors prevail. Thus, the adjoint-based $d(C_L/C_D)/d\mathbf{v}$ can be regarded as at least as accurate as finite-differences with optimal step-sizes, which is in any rate sufficient for the purpose of gradient-based optimization.

2. Inverse Design

The inverse design problem is based on surface pressure and is intended to validate SNOPT itself. First, the half-sphere is deformed by randomly perturbing the coordinates of four control points of its enclosing FFD

lattice. A flow solve is then performed, and the nodal pressure distribution recorded. Finally, SNOPT is given the unperturbed sphere and attempts to recover the perturbed variables using the objective

$$\mathcal{J} = \frac{1}{2} \sum_{i=1}^{N_{\text{surf}}} (p_i - p_{i,\text{rec}})^2 \Delta A_i,$$

where N_{surf} is the number of surface grid points, and ΔA_i , p_i and $p_{i,\text{rec}}$ are the surface area element, pressure and recorded pressure at node i , respectively. As seen from Figure 3(b), the optimizer could reduce \mathcal{J} by 18 orders and the optimality, a measure of the gradient, by 10 orders, thus successfully retrieving the perturbed coordinates of the sphere to almost machine accuracy in 29 function evaluations.

V. From Sphere to Aircraft

Both ASO and MDO are now used routinely to improve current aircraft designs. But can we do better than that? In particular, can these tools discover novel aircraft capable of superior performances compared with the conventional tube-and-wing? While we acknowledge that the full answer will undeniably involve highly-coupled MDO, as a step toward that direction we attempt, through aerodynamics only, to maximize the lift-to-drag ratio of an initially half-sphere-shaped geometry. A thorough analysis of this exploratory shape optimization problem is however beyond the scope of this paper, which is merely to demonstrate the flexibility and robustness of the proposed two-level FFD approach in a particularly demanding ASO setting. Thus, here we only perform a single-objective, gradient-based optimization in the hope of finding one local optimum out of the likely highly multimodal design space.

The half-sphere geometry used for this study is of radius $r = 1$ and corresponds to the union of 20 B-spline patches, each made up of a 6×6 control net, for a total of 720 non-unique surface control points. All of these are embedded inside the FFD volume depicted on the left-hand-side of Figure 1; it is in fact a $6 \times 6 \times 2$ NURBS volume, cubic in both streamwise (X) and spanwise (Y) directions, and linear in the vertical (Z) direction. The control and computational grids are those of Figure 2(a), which are respectively composed of 13,720 and 900,000 nodes, with respective off-wall spacings on the order of $O(10^{-2})$ and $O(10^{-3})$. The farfield velocity is kept subsonic at Mach 0.5 (to avoid the formation of shocks) with a fixed 1.2 degree angle of incidence. This choice of α is somewhat arbitrary; the idea is simply to discourage an overly pitched evolving geometry.

The number and choice of design variables in ASO can have a significant impact on both the final shape obtained and the convergence of the optimizer. For instance, experience suggests that planform-like variables can be an effective means to consolidate clouds of control points. Thus, instead of choosing the coordinates of every FFD lattice control point as design variables (of which there are 216), we group together all the control points lying on the same spanwise station of the FFD volume's control net. There are 6 such stations in the present case (see Figure 5 where the last one is showcased in the gray rectangle), and each one is free to translate (blue arrows) in a direction only if its corresponding bit (found in the lower left portion of Figure 5) is turned on; the rightmost bit refers to translation in X , the next one in Y and the leftmost one in Z . These so-called cross-section design variables are well suited for global deformations, but another mechanism able to develop airfoil-like profiles must also exist. For that purpose, we give each control point the freedom to scale vertically in its own cross-sectional plane (small red arrows). An additional scaling variable, this one global and perpendicular to the ones just described, is also used so as to allow variations in taper (big red arrows).

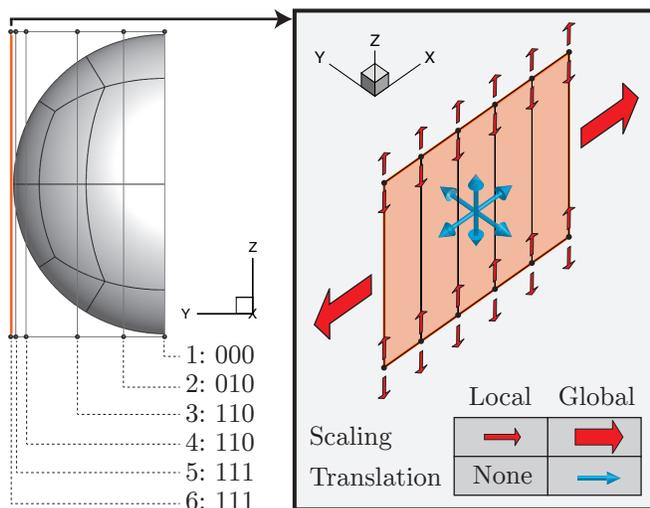


Figure 5. Cross-section design variables: groups of FFD control points are translated and/or scaled together. All cross-sections have their scaling design variables enabled.

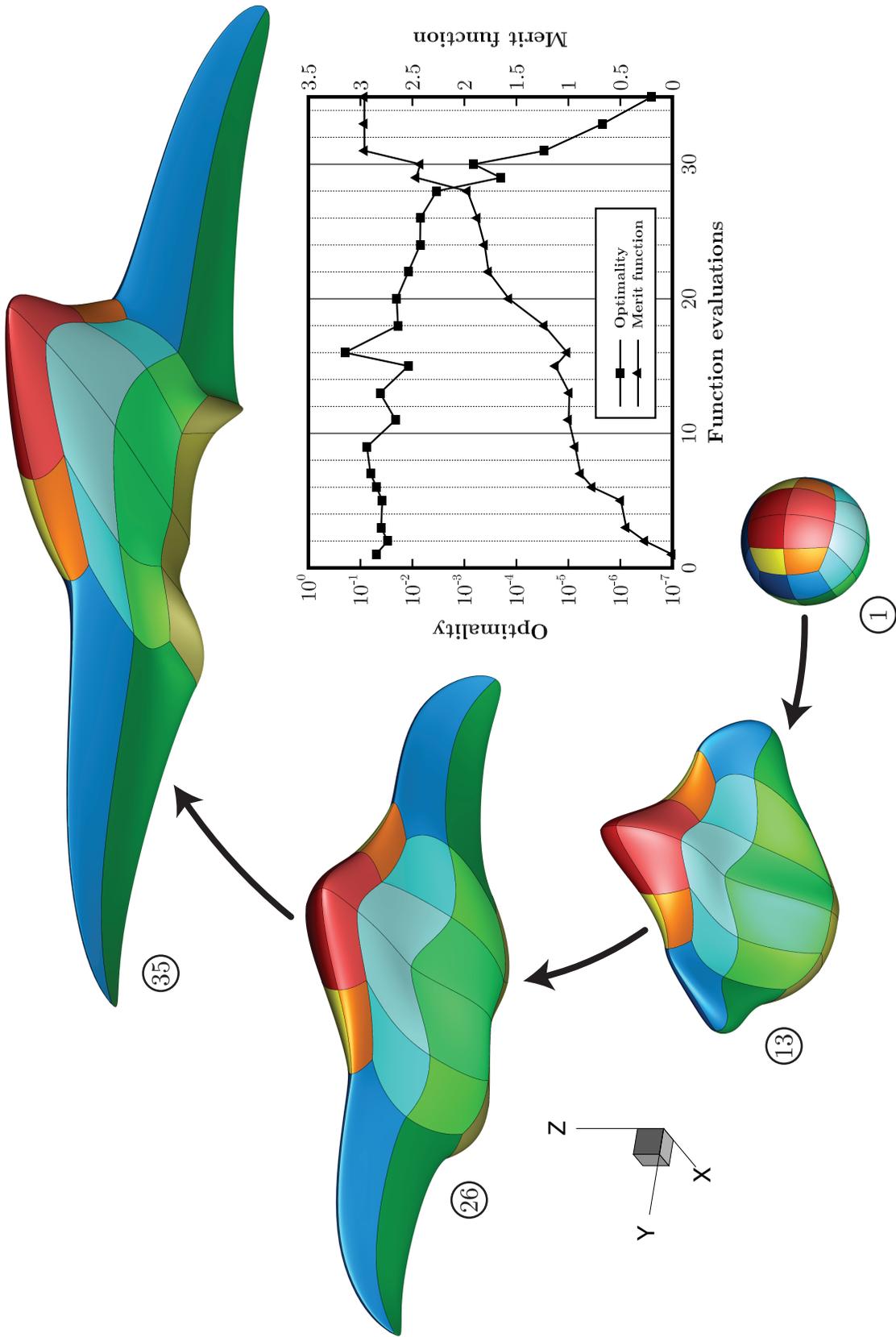


Figure 4. Shape evolution and associated convergence history of the sphere during the gradient-based optimization. Circled numbers refer to the number of times the optimizer has called the objective function evaluation procedure so far.

Summarizing, there are 11 translational and 78 scaling design variables, for a total of 89 overall design variables. As previously stipulated, α remains fixed at all time and is thus excluded from \mathbf{v} .

In order to ascertain the existence of a local optimum, SNOPT uses a Lagrangian in much the same way the gradient evaluation procedure does, where the first-order KKT conditions^b must be converged to within an “optimality” tolerance, here specified to $\epsilon = 10^{-6}$. SNOPT best satisfies these conditions if there are only relatively few degrees of freedom at a solution, i.e. many constraints are active. While constraints in a real-life problem are generally meaningful, such as manufacturing requirements would call for, here we use them solely as a means to enable SNOPT to converge. Specifically, we put quite restrictive bounds on the design variables, especially the ones related to scaling, while still allowing for blended-wing-body-like shapes to develop. We also impose two other set of linear constraints, one designed to maintain symmetry across the $Y = 0$ plane and the other to prevent two adjacent FFD stations to overlap. Finally, we force the evolving shape to maintain an internal volume of at least that of the initial half-sphere, i.e. $2\pi/3$, through a nonlinear inequality constraint.

The evolution of the geometry shown at key stages of this highly constrained problem along with its convergence history are pictured in Figure 4. Although not directly apparent from the diagram, the optimizer morphs the sphere to a blended-wing-body in three distinct phases. The first one (function evaluations 1 to 13) gives the sphere some way to produce lift by mostly varying local scaling variables. The second phase, spanning function evaluations 14 to 26, is characterized by a flattening of the contorted shape relative to the streamwise direction. At the end of this phase, the maximum bounds acting on the streamwise global scaling variables are already reached. Although wings have started protruding before then, it is only during the final phase (function evaluations 27 to 35) that the optimizer really exploits the spanwise translational design variables. This is also where most improvement is achieved. The final shape has a $C_L = 0.5386$, $C_D = 0.1818$, half-span of 2.8990 unit and projected surface area of 3.9299 unit² (recall that the sphere has an initial radius of 1 unit); C_L/C_D and the aspect ratio are thus 3.0 and 4.3, respectively.

We emphasize that the optimized shape is in no way the best possible; according to lifting-line theory, an optimally loaded planar wing with the same lift coefficient and aspect ratio has a drag coefficient of

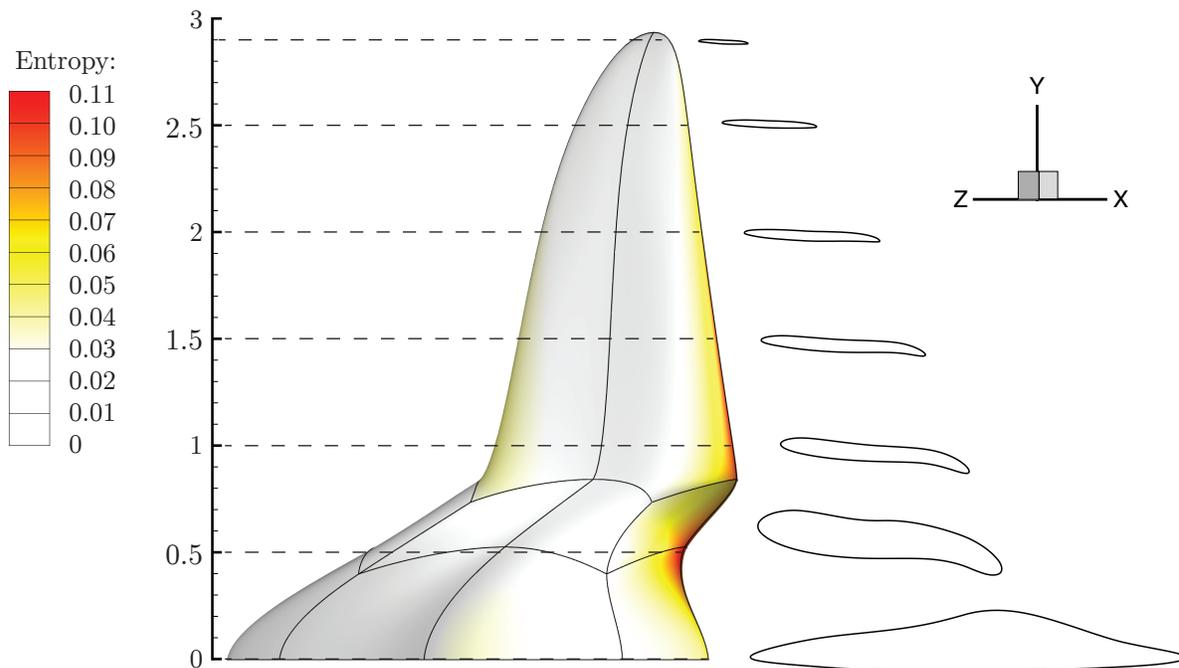


Figure 6. Entropy contours on the optimized shape along with spanwise airfoil profiles taken at 0.5 unit apart, except the last one which is located at 2.8 unit. Most entropy is generated at the trailing edge, especially inboard where radii of curvature are highest. Inviscid air flows from the $-X$ direction at Mach 0.5 with a 1.2 degree angle-of-attack.

^bThese are not the same as in the sequential approach used to determine the gradient. Rather, they emanate from the objective and constraints provided to SNOPT.

$C_{D,\text{ellip}} = 0.0401$, which is about 4.5 times less compared with the optimized shape. This poor performance can be partly explained by the optimizer failing to form a sharp trailing edge, which is an important feature of any high-lifting body. In fact, it was impossible for it to do so since the sphere was smooth in the first place. Instead, the final shape exhibits a small (in the scale of the grid resolution) rounded trailing edge where numerical dissipation prevails; see Figure 6 where entropy contours pertaining to the optimized shape are plotted. This artificial dissipation triggers (non-physical) local flow separation, which in turn increases induced drag. It is therefore reasonable to expect much smaller values of drag should the sphere be allowed to develop a sharp trailing edge, a task however difficult given the smooth properties of FFD volumes. In a sense, the final shape’s performance was thus limited more by the problem definition, i.e. starting off from a smooth sphere with a relatively coarse mesh, than by the aerodynamic optimizer itself. Indeed, by refining the final grid alone from 900,000 to 76,317,120 nodes (a task made easy given that the grid is described analytically by B-spline volumes, recall Eq. (5)), the induced drag reduces from 0.1818 to 0.1118 (which is about 2.8 times $C_{D,\text{ellip}}$), yielding a lift-to-drag ratio of 5.5.

VI. Conclusions

A robust and efficient geometry framework comprising a novel two-level FFD approach suitable for high-fidelity ASO and MDO has been presented. The result is a flexible technique that retains an analytical geometry representation throughout an optimization and provides an efficient integrated mesh movement capability that maintains mesh quality through large shape changes.

The framework proved to be a natural extension to an already existing gradient-based aerodynamic optimizer capable of accurate functional and gradient evaluations with fast turnaround. The augmented optimizer offers powerful tools to the aerodynamicist, a fact demonstrated by the challenging task of maximizing the lift-to-drag ratio of an initially spherical geometry. While the final shape was encouraged to evolve as it did through constraints, it is an optimized one nonetheless, and it therefore further demonstrates the suitability of the proposed two-level FFD approach to high-fidelity ASO. Future work will focus on incorporating these tools within a high-fidelity MDO framework.

Acknowledgments

The authors gratefully acknowledge funding from the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC). Computations were performed on the GPC supercomputer at the SciNet HPC Consortium and on the Guillimin supercomputer at McGill University.

References

- ¹IPCC, “Climate Change 2007: Synthesis Report,” Tech. rep., Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, R. K. Pachauri and A. Reisinger (eds.)], Geneva, Switzerland, 2007, 104 pp.
- ²Green, J. E., “Civil aviation and the environment - the next frontier for the aerodynamicist,” *The Aeronautical Journal*, Vol. 110, No. 1110, August 2006, pp. 469–486.
- ³Jameson, A., “Aerodynamic Design via Control Theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, September 1988, pp. 233–260.
- ⁴Nadarajah, S. K. and Tatossian, C., “Multi-objective aerodynamic shape optimization for unsteady viscous flows,” *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 67–106.
- ⁵Kim, J.-E., Rao, V. N., Koomullil, R. P., Ross, D. H., Soni, B. K., and Shih, A. M., “Development of an efficient aerodynamic shape optimization framework,” *Mathematics and Computers in Simulation*, Vol. 79, No. 8, April 2009, pp. 2373–2384.
- ⁶Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “A Scalable Parallel Approach for High-Fidelity Aerostructural Analysis and Optimization,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 2012-1922, Honolulu, Hawaii, 2012.
- ⁷Hicken, J. E. and Zingg, D. W., “Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement,” *AIAA Journal*, Vol. 48, No. 2, February 2010, pp. 400–413.
- ⁸Hicken, J. E. and Zingg, D. W., “Induced-drag minimization of nonplanar geometries based on the Euler equations,” *AIAA Journal*, Vol. 48, No. 11, November 2010, pp. 2564–2575.
- ⁹Fudge, D., Zingg, D. W., and Haimes, R., “A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimization,” *43rd AIAA Aerospace Sciences Meeting*, AIAA Paper 2005-0451, Reno, Nevada, January 2005.

- ¹⁰Yu, G., Müller, J.-D., Jones, D., and Christakopoulos, F., “CAD-based shape optimisation using adjoint sensitivities,” *Computers & Fluids*, Vol. 46, No. 1, July 2011, pp. 512–516.
- ¹¹Claus, R., Follen, G. J., and Haimes, R., “CAPRI: Computational Analysis PROGRAMMING Interface. ‘A CAD infrastructure for Aerospace Analysis and Design Simulations’,” Tech. rep., National Aeronautics and Space Administration, 2000.
- ¹²Nemec, M., Aftosmis, M. J., and Pulliam, T. H., “CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method,” *42nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2004-0113, 2004.
- ¹³Samareh, J. A., “Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization,” *AIAA Journal*, Vol. 39, No. 5, May 2001, pp. 877–884.
- ¹⁴Kulfan, B., “Universal Parametric Geometry Representation Method,” *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158.
- ¹⁵Hicks, R. M. and Henne, P. A., “Wing Design by Numerical Optimization,” *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.
- ¹⁶Jameson, A. and Reuther, J., “Control theory based airfoil design using the Euler equations,” *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City Beach, Florida, September 1994.
- ¹⁷Nemec, M. and Zingg, D. W., “Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations,” *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1146–1154.
- ¹⁸de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, 1978.
- ¹⁹Zhao, N. and Ma, W., “Properties of G1 continuity conditions between two B-spline surfaces,” *Advances in Computer Graphics*, Vol. 4035, 2006, pp. 743–752.
- ²⁰Jakobsson, S. and Amoignon, O., “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,” *Computers & Fluids*, Vol. 36, No. 6, July 2007, pp. 1119–1136.
- ²¹Morris, A. M., Allen, C. B., and Rendall, T. C. S., “Domain-Element Method for Aerodynamic Shape Optimization Applied to Modern Transport Wing,” *AIAA Journal*, Vol. 47, No. 7, 2009, pp. 1647–1659.
- ²²Ben-Chen, M., Weber, O., and Gotsman, C., “Variational harmonic maps for space deformation,” *ACM Transactions on Graphics*, Vol. 28, No. 3, Article 34, 2009, pp. 1–11.
- ²³Sederberg, T. W. and Parry, S. R., “Free-Form Deformation of Solid Geometric Models,” *SIGGRAPH ’86 Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, Dallas, Texas, August 1986.
- ²⁴Gain, J. and Bechmann, D., “A Survey of Spatial Deformation from a User-Centered Perspective,” *ACM Transactions of Graphics*, Vol. 27, No. 4, 2008, pp. 21 pp.
- ²⁵Anderson, G. R., Aftosmis, M. J., and Nemec, M., “Parametric Deformation of Discrete Geometry for Aerodynamic Shape Design,” *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0965, Nashville, Tennessee, January 2012.
- ²⁶Amoiralis, E. I. and Nikolos, I. K., “Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design,” *Journal of Computing and Information Science in Engineering*, Vol. 8, No. 2, June 2008, pp. 01–13.
- ²⁷Anderson, W. K., Karman, S. L., and Burdyslaw, C., “Geometry Parameterization Method for Multidisciplinary Applications,” *AIAA Journal*, Vol. 47, No. 6, 2009, pp. 1568–1578.
- ²⁸Brezillion, J. and Dwight, R. P., “Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations,” *CEAS Aeronautical Journal*, October 2011.
- ²⁹Anderson, G. R., Aftosmis, M. J., and Nemec, M., “Constraint-based Shape Parameterization for Aerodynamic Design,” *7th International Conference on Computational Fluid Dynamics*, ICCFD7 Paper 2001, Big Island, Hawaii, July 2012.
- ³⁰Samareh, J. A., “Novel Multidisciplinary Shape Parameterization Approach,” *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 1015–1024.
- ³¹Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA Paper 2010-9231, Fort Worth, Texas, September 2010.
- ³²Blake, M. W., Kerr, P. A., Thorp, S. A., and Chou, J. J., “NASA Geometry Data Exchange Specification for Computational Fluid Dynamics (NASA IGES),” Tech. Rep. 1338, National Aeronautics and Space Administration, 1994.
- ³³Piegl, L. A., “On NURBS: a survey,” *Computer Graphics and Applications*, Vol. 11, No. 1, 1991, pp. 55–71.
- ³⁴Morris, A. M., Allen, C. B., and Rendall, T. C. S., “High-fidelity aerodynamic shape optimization of modern transport wing using efficient hierarchical parameterization,” *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 3, May 2010, pp. 297–312.
- ³⁵Gain, J. E. and Dodgson, N. A., “Preventing Self-Intersection under Free-Form Deformation,” *Transactions on Visualization and Computer Graphics*, Vol. 7, No. 4, 2001, pp. 289–298.
- ³⁶Yu, T. and Soni, B. K., “Application of NURBS in numerical grid generation,” *Computer-Aided Design*, Vol. 27, No. 2, February 1995, pp. 147–157.
- ³⁷Truong, A. H., Oldfield, C. A., and Zingg, D. W., “Mesh Movement for a Discrete-Adjoint Newton-Krylov Algorithm for Aerodynamic Optimization,” *AIAA Journal*, Vol. 46, No. 7, July 2008, pp. 1695–1704.
- ³⁸Chernukhin, O. and Zingg, D. W., “An Investigation of Multi-Modality in Aerodynamic Shape Optimization,” *29th AIAA Applied Aerodynamics Conference*, AIAA Paper 2011-3070, Honolulu, Hawaii, June 2011.
- ³⁹Zingg, D. W., Nemec, M., and Pulliam, T. H., “A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization,” *European Journal of Computational Mechanics*, Vol. 17, No. 1-2, 2008, pp. 103–126.
- ⁴⁰Osusky, L. and Zingg, D. W., “A Novel Aerodynamic Shape Optimization Approach for Three-Dimensional Turbulent Flows,” *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0058, Nashville, Tennessee, January 2012.
- ⁴¹Hicken, J. E. and Zingg, D. W., “Parallel Newton-Krylov Solver for the Euler Equations Discretized Using Simultaneous-Approximation Terms,” *AIAA Journal*, Vol. 46, No. 11, November 2008, pp. 2273–2786.
- ⁴²Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer, 2nd ed., 2006.

- ⁴³Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.
- ⁴⁴Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., “The Complex-Step Derivative Approximation,” *ACM Transactions on Mathematical Software*, Vol. 29, 2003, pp. 245–262.