Two-Level Free-Form and Axial Deformation for Exploratory Aerodynamic Shape Optimization

Hugo Gagnon[1] and David W. Zingg[2]
University of Toronto, Toronto, Ontario M3H 5T6, Canada

An intuitive shape parameterization and control technique suitable for high-fidelity aerodynamic shape optimization is presented. It relies on the principles of free-form and axial deformation, enabling thorough exploration of the design space while keeping the number of design variables manageable. Surface sensitivities to the design variables are readily available; their inclusion in a highly efficient and robust adjoint-based optimization methodology involving linearly elastic volume mesh deformation and a Newton-Krylov solver for the Euler equations is described. The flexibility of the proposed approach is demonstrated through the exploratory shape optimization of a three-pronged feathered winglet, leading to a span efficiency of 1.19 under a height-to-span ratio constraint of 0.1, and an optimization of a regional jet wing at transonic speed where a winglet is allowed to develop starting from a planar wingtip extension, leading to an 18.8% reduction in drag.

Nomenclature

$\boldsymbol{A}$	surface control points
$\boldsymbol{b}, \boldsymbol{B}$	volume control points
b	wing span
c	chord length
c_l, c_d	coefficients of lift and drag (section)
C_L, C_D	coefficients of lift and drag

[1] Ph.D. Candidate, Institute for Aerospace Studies. AIAA Student Member.

[2] Professor and Director, Institute for Aerospace Studies, Tier 1 Canada Research Chair in Computational Aerodynamics and Environmentally Friendly Aircraft Design, J. Armand Bombardier Foundation Chair in Aerospace Flight. AIAA Associate Fellow.

C_p	coefficient of pressure
e	span efficiency
h	wing vertical extent
H	projective mapping
$\mathcal{J}$	aerodynamic functional (L/q_∞ or D/q_∞)
L, D	lift and drag
$\mathcal{L}$	Lagrangian
$\boldsymbol{\mathcal{M}}$	mesh residuals
N	number of control points
$\mathcal{N}^{(p)}$	Basis spline (degree p)
$\boldsymbol{P}$	control point coordinates (physical space)
$\boldsymbol{P}^w$	control point coordinates (homogeneous space)
$\boldsymbol{q}$	flow solution
q_∞	freestream dynamic pressure
$\boldsymbol{\mathcal{R}}$	flow residuals
S	reference area
$\boldsymbol{v}$	geometric design variables
α	freestream angle of attack
Λ	aspect ratio
$\boldsymbol{\lambda}, \boldsymbol{\psi}$	mesh- and flow-adjoint variables

I. Introduction

The impact of aviation on climate change is projected to quadruple by the year 2050 [1]. Since emissions are directly related to fuel burn, and fuel burn to drag, one promising avenue to mitigate aviation's impact on climate change is to minimize drag (for a fixed mission). Numerical tools built upon optimal control theory are extremely promising in this regard [2], yet radically new configurations can only be discovered through flexible and efficient shape parameterizations [3]. Without a flexible parameterization, the true optimal shape might not be found; if inefficient, the parameterization may produce more design variables than necessary and slow down convergence.

Computer-aided design (CAD) packages are, at first glance, well-suited for the task [4, 5]. Sophisticated interfaces help capture and store design intent in the form of a centralized master model, which can be reused at various fidelity levels, thus seamlessly bridging the gap with manufacturing [6, 7]. Unfortunately, many difficulties related with interfacing a high-end CAD package with external optimizers persist, mainly because the sensitivity of a surface mesh to the geometric design variables cannot be retrieved by usual means such as automatic differentiation [8, 9]. This forces costly and potentially inaccurate finite-difference approximations [10], or, if at all possible, the daunting task of identifying and differentiating individual geometric entities implicitly defined by the associativity tree of a master model [11].

Due to the integration issues related with commercial CAD, it is not surprising that many authors have turned to CAD-free parameterizations that are specifically tailored to aerodynamic design. In his early work, Jameson [2] parameterized airfoils with individual grid points, an approach that however requires gradient filtering in order to maintain smooth geometries. Hicks and Henne [12] proposed sinusoidal shape functions; although smooth, they generally fail to form a complete basis, which might preclude an optimizer from finding the best possible shape. Robinson and Keane [13] derived orthogonal basis functions using state-of-the-art supercritical airfoils. Sobieczky's PARSEC method [14] characterizes airfoils with practical parameters such as leading edge radius and trailing edge angle. Kulfan's CST method [15] extends this idea to a set of fundamental "classes" of aerodynamic bodies. The lesser known PDE method of Bloor and Wilson [16] also seeks to minimize the number of design parameters through the solution of boundary value problems.

Perhaps the best compromise between tailored parameterizations and the flexibility of CAD packages is the one offered by splines [17]. In fact, CAD kernels are based on a generalization of splines called non-uniform rational B-splines (NURBS) [18]. Unlike polynomial series, B-splines and NURBS alike can capture complicated geometries by using knot insertion rather than degree elevation. Their so-called "control points" are intuitive to work with, providing a natural selection of design variables. Braibant and Fleury [19] were among the first to use B-splines in shape optimization; Cosentino and Holst [20] optimized sections of a transonic wing parameterized with cubic splines; Anderson and Venkatakrishnan [21] implemented B-splines in an aerodynamic unstructured code coupled with a continuous adjoint approach.

Regardless of their mathematical representation, curves and surfaces alone cannot naturally account for volumetric changes, as would be desirable in certain applications involving say, an aircraft outer mold line enclosing internal structures. Although for different motives, the soft object animation industry (a subset of computer graphics) has long recognized this limitation by adopting a number of spatial deformation techniques [22]. One such technique, called free-form deformation (FFD) [23], has been steadily gaining popularity in the aerodynamic shape optimization (ASO) and multidisciplinary design optimization (MDO) communities [24–28]. Despite its many advantages, FFD has one major drawback: the cubic rate at which the number of control points required to define a volume rises. Cage-based deformation can alleviate this by defining only the exterior points of a lattice; the interior domain can then be deformed, for example, with a Laplacian operator [29]. Subdivision surfaces also start from a cage lattice and recursively subdivide each face to produce up to C^2 continuous surfaces; see [30] for a rare application in shape optimization.

Volume-based parameterizations, on the other hand, intrinsically lack surface awareness: real world requirements such as thickness and curvature constraints are difficult to achieve. Hsu *et al.* [31] reconciled the two worlds by allowing direct manipulation of any point on an object and subsequently solving for the corresponding FFD lattice, a process commonly known as "inverse FFD". Yamazaki *et al.* [32] successfully applied this technique to ASO through the use of "pilot points" while also minimizing an energy functional for even more realistic surface deformation. A promising alternative to FFD was presented by Jakobsson and Amoignon [33], who achieved intuitive and robust global deformation with radial basis functions (RBF). Morris *et al.* [34] later used similar techniques to perturb both surface and volume meshes in one shot. Yet another user-centered alternative to FFD and RBF consists of using lower dimensional objects such as points, curves, and even surfaces to quickly deform ambient three-dimensional space [22]. In this work, we combine the intuitiveness of one such technique with the strength of FFD.

We present here a CAD-free parameterization that is specifically tailored to wing systems of arbitrary topology. It includes a two-level FFD scheme [28], in which both the geometry and surrounding volume grid are parameterized with NURBS-based FFD volumes [35]. Since the entire formulation is in fact solely built around NURBS entities, the connection with CAD is maintained [36]. A salient feature of the present methodology is that it is the control points of surfaces, as opposed to grid points, that are embedded inside FFD volumes, an approach that leverages an extremely robust and efficient volume mesh movement algorithm [37, 38]. Another novel aspect of the present work lies in the use of *axial curves* [39] to achieve wing deformation by intuitively reorienting spanwise FFD volume cross-sections of control points [40]. We opt for NURBS curves as axial curves [40], not only making the parameterization both flexible and efficient, but also amenable to dynamic adaptation [41] using for example knot insertion [42].

The remainder of this paper is divided as follows. In Section II, a short exposé on NURBS and FFD lead to a description of the two-level FFD scheme responsible for surface and volume mesh deformation. The axial deformation technique augmenting these tools is also formally introduced. Section III gives a brief overview of the overall aerodynamic optimizer, with emphasis on the incorporation of the surface sensitivities within the adjoint framework. The flexibility of the proposed parameterization is then demonstrated in Section IV, which contains the exploratory aerodynamic shape optimization of a feathered wingtip and a regional jet transonic wing. Finally, concluding remarks and future directions are the subject of Section V.

II. Shape Parameterization and Control

A. Non-Uniform Rational B-Spline Volumes

A Basis spline $\mathcal{N}^{(p)}(\xi)$, or B-spline, is a piecewise nonrational polynomial of degree p (order $p+1$) joined at nondecreasing (and in general nonuniform) $m+1$ knot locations given by

$$\Xi = \{\underbrace{0,\ldots,0}_{p+1},\xi_{p+1},\ldots,\xi_{m-p-1},\underbrace{1,\ldots,1}_{p+1}\} \tag{1}$$

(a zero-based indexing is used throughout this paper, unless noted otherwise). The knot vector expressed by Eq. (1) is here conveniently normalized and assumed nonperiodic, i.e. with repetitive first and last knots. This choice ensures that $\mathcal{N}_0^{(p)}(0)$ and $\mathcal{N}_{m-p-1}^{(p)}(1)$ are both equal to unity, as can be seen by examining the following recursive definition of B-splines:

$$\mathcal{N}_i^{(0)}(\xi) = \begin{cases} 1 & \text{if } \xi_i \le \xi < \xi_{i+1} \\ \\ 0 & \text{otherwise} \end{cases}$$
$$\mathcal{N}_i^{(p)}(\xi) = \left(\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i}\right)\mathcal{N}_i^{(p-1)}(\xi) + \left(\frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}}\right)\mathcal{N}_{i+1}^{(p-1)}(\xi). \tag{2}$$

B-splines are infinitely differentiable on the real line, except at the knots where they are C^{p-k} continuous, k being the knot multiplicity.

Given three normalized, nonperiodic knot vectors, each associated with a different dimension in $\mathbb{R}^3$, then the mapping $D = \{\boldsymbol{\xi} = (\xi,\eta,\zeta) \in \mathbb{R}^3 \mid \boldsymbol{\xi} \in [0,1]^3\} \to P \subset \mathbb{R}^4$ described by the tensor-product definition of a NURBS volume [18],

$$\boldsymbol{V}^w(\boldsymbol{\xi}) = \sum_{i=0}^{N_i}\sum_{j=0}^{N_j}\sum_{k=0}^{N_k} \mathcal{N}_i^{(p)}(\xi)\,\mathcal{N}_j^{(q)}(\eta)\,\mathcal{N}_k^{(r)}(\zeta)\,\boldsymbol{P}_{ijk}^w, \tag{3}$$

is guaranteed to match the eight corner control points of the lattice $\{\boldsymbol{P}^{w}_{ijk}\}$. The above definition implies that the number of control points in the ξ direction, $N_i + 1$, must equal $m - p$, i.e. the index of the last knot minus the degree of the spline associated with that direction; similar remarks hold for the η and ζ directions.

Equation 3 offers a convenient nonrational formulation of a NURBS volume that is however only valid in four-dimensional (homogeneous) space. To retrieve the desirable, three-dimensional object $\boldsymbol{V}(\xi)$, it is necessary to apply a projective mapping H to each control point $\boldsymbol{P}^{w} = (wx, wy, wz, w) = (X, Y, Z, W)$, which consists of dividing the first three coordinates by the last one (referred to as the weight w of the control point), i.e.

$$\begin{aligned}\boldsymbol{P} &= H\left\{\boldsymbol{P}^{w}\right\} = H\left\{(X, Y, Z, W)\right\}\\ &= \begin{cases}\left(\dfrac{X}{W}, \dfrac{Y}{W}, \dfrac{Z}{W}\right) & \text{if } W \neq 0\\ \text{direction } (X, Y, Z) & \text{if } W = 0\end{cases}\end{aligned} \tag{4}$$

The special case $W = 0$ corresponds to a point at infinity and is seldom used in practice. The other special case $W = 1$ gives no particular consideration to a point, and it can be shown that if all weights equal unity then Eq. (3) reduces to the traditional (nonrational) B-spline volume definition [18].

B. Two-Level Free-Form Deformation

A scheme based on two separate sets, or levels, of FFD volumes is introduced in order to respectively account for the geometry changes and resulting volume mesh movement that occur during the course of a shape optimization problem [28].

Conceptually, FFD [23] is best visualized as embedding flexible, rubbery objects into a transparent material having the same constitutive properties. As the larger block deforms, so will the embedded objects. The deformation is completely dissociated from the objects' underlying representation: clouds of points can coexist alongside structured data without any special consideration. In practice, any parametric representation can be utilized as the embedding material; here, only trivariate NURBS volumes as expressed by Eq. (3) are employed. For this reason, the terms “NURBS volume” and “FFD volume” will now be used interchangeably.

Formally, FFD can be formulated as a mapping $\mathcal{F}\colon \mathbb{R}^3 \to \mathbb{R}^3 \to \mathbb{R}^3$, from physical space $\boldsymbol{t}$, to parametric space $\boldsymbol{\xi}$, to deformed physical space $\tilde{\boldsymbol{t}}$. This is achieved by two functions. The first one is the embedding function $F^{-1}(\boldsymbol{t}) = \boldsymbol{\xi}$, which associates a parametric value $\boldsymbol{\xi}$ to each vertex $\boldsymbol{t}$ of an object. This is normally carried out by a Newton search procedure and needs only to be performed once. The second function is the deformation function $\tilde{F}(\boldsymbol{\xi}) = \tilde{\boldsymbol{t}}$, which translates to simply re-evaluating Eq. (3) for each embedded vertex once an FFD lattice $\{\boldsymbol{P}^{w}_{ijk}\}$ has deformed. The composition of F^{-1} and $\tilde{F}$ is what constitutes FFD:

$$\mathcal{F}(\boldsymbol{t},\tilde{\boldsymbol{t}}) = \tilde{F}(F^{-1}(\boldsymbol{t})) - \tilde{\boldsymbol{t}} = \tilde{F}(\boldsymbol{\xi}) - \tilde{\boldsymbol{t}} = \boldsymbol{0}. \tag{5}$$

In the context of ASO, the primary advantage of NURBS-based FFD is to allow simple sets of geometric design variables to manipulate complex geometries intuitively. 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 volume control point and twisting effects can be achieved by rotating entire cross-sections of control points. Indeed, an entire wing can be twisted using only 8 FFD volume control points (albeit linearly), a difficult task for a B-spline surface parameterization alone where the coupling of surface control points can be involved, especially when boundary requirements such as geometric continuity along the leading edge must 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.

Now let $\boldsymbol{A}$ denote an xyz-coordinate block-column vector corresponding to all the unique NURBS surface control points that define a watertight geometry. Then the first level of the two-level scheme involves embedding some or all of those control points inside one or more FFD volumes. As just described, the geometry can then be modified by perturbing the control points, hereafter represented by the vector $\boldsymbol{B}$, pertaining to the lattices of those FFD volumes. Note that by embedding control points of the underlying NURBS surfaces, rather than the usual grid points that lie on them, the exact geometry representation is never lost. This means that a geometry can be exported to CAD (and back from it) at any point during the optimization process without having to be reverse-engineered. Depending on the desired level of local control, a large number of surface control points may be necessary. This is because the local support of the embedded surface control points extends beyond the local support of the FFD volume control points. Nevertheless, the ratio of surface grid points to embedded surface control points remains high, particularly so for denser grids such as the ones destined for turbulence modeling.

At the second level, each block in the three-dimensional structured domain surrounding the geometry is assigned a NURBS volume[1]. Once again, these volumes can be regarded as FFD volumes where this time the full set of computational grid points is embedded inside them. The key observation is that the global lattice formed by the union of all the FFD volume control points acts as a coarse grid $\boldsymbol{b} \supset \boldsymbol{A}$. In words, there is a one-to-one correspondence between the control points in $\boldsymbol{b}$ embracing the geometry with those in $\boldsymbol{A}$. Mesh movement is thus tightly integrated with geometry parameterization. While this approach greatly simplifies implementation (and more importantly speeds up mesh movement, as discussed in the next paragraph), it does restrain the topology of the surface to that of the computational grid. In practice, however, this restriction is inconsequential, since 1) we assume a fixed topology and 2) we use a third-party grid generator to build the initial computational grid and subsequently fit it to $\boldsymbol{b}$ at the beginning of the optimization, starting from the edges, followed by the faces, and finally the interior of each block. As detailed in [38], high-quality fits can be achieved with curved knot lines [43] as well as parameter correction [44].

In response to boundary alterations $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{B})$ resulting from the first level of FFD volumes, the control points $\boldsymbol{b}$ of the second FFD level are updated according to the equations of linear elasticity [37]:

$$\mathcal{M}(\boldsymbol{A}(\boldsymbol{B}), \boldsymbol{b}) = \boldsymbol{0}. \tag{6}$$

After solving Eq. (6), the entire computational grid is regenerated algebraically based on the definition of the NURBS volumes. This last operation is almost instantaneous, and since there are typically one to three orders of magnitude fewer control points $\boldsymbol{b}$ than grid points, 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 [38].

The elasticity-based model is only valid under a small-strain assumption. If shape changes are sufficiently large, the mesh quality is maintained by breaking up the displacements of the surface control points into multiple increments, implying as many mesh solves [37]. We assume only one increment in this paper for the sake of simplicity.

[1] Unstructured grid users may use high-order tetrahedra, or in fact any other finite element commonly found in computational solid mechanics.

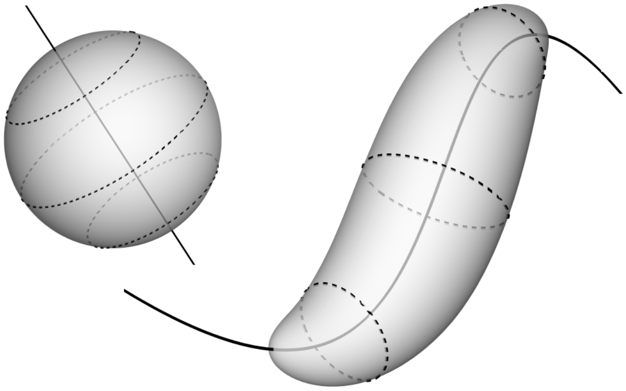

Fig. 1 Axial deformation: a curve deforms an object in 3D space.

C. Axial Deformation Tailored to Wing Design

If one is not careful, the attractive properties of FFD — smoothness, local control, rapid deformation, mathematical background — can be quickly overshadowed by the overwhelming number of control points that an FFD lattice may require. This is a simple consequence of its tensor product definition. Ideally, a wing designer would be able to rely on FFD's excellent intrinsic properties, while being spared from the cumbersome task of constraining every individual control point to prevent unfeasible designs. The most obvious way of achieving this is to group control points by proximity, and let a simpler, higher level object control the overall movement of the lattice. In this work, we investigate the use of axial curves for such purposes in the context of wing design. The result is a highly general and intuitive deformation technique that confines the design space to what makes sense aerodynamically, but without overly constraining possibly optimal wing shapes.

The term "axial deformation (AxDf)" was coined by Lazarus *et al.* [39] in 1994 and has since enjoyed some popularity, especially in the computer graphics community [22]. Like FFD, it is a volumetric deformation technique that, as its name suggests, operates from a single curve, i.e. the axial curve. Figure 1 shows the conceptual idea behind its functioning. First, an axial curve is positioned anywhere inside an object of interest. Once every point of that object has been associated (mapped) to a point on the axial curve, the axial curve is deformed, after which the initial points are re-evaluated to their new physical space coordinates based on their new local coordinate frame. Although not shown in Fig. 1, other effects such as twisting and scaling can be achieved by means of transformational functions.

The application to wing deformation follows directly. Refer to Fig. 2 where the chordwise, spanwise,

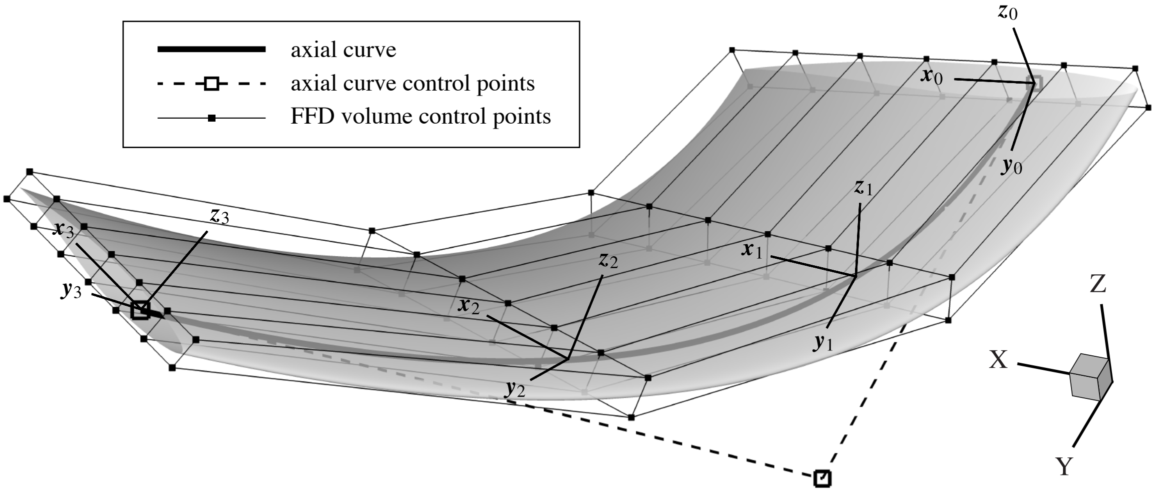

Fig. 2 Axial-curve-driven FFD applied to a generic wing.

and vertical directions of the wing are assumed to be oriented along the positive x, y, and z directions, respectively.

Let the axial curve be a p-degree NURBS curve defined by $N + 1$ weighted control points $\boldsymbol{P}_i^w$ and as many B-splines $\mathcal{N}_i^{(p)}(\xi)$ mapping $D = \{\xi \in \mathbb{R} \mid \xi \in [0,1]\}$ to $P \subset \mathbb{R}^4$ [18]:

$$\boldsymbol{C}^w(\xi) = \sum_{i=0}^{N} \mathcal{N}_i^{(p)}(\xi)\boldsymbol{P}_i^w. \tag{7}$$

Recall from Eqs. (1–2) that a nonperiodic knot vector must be used if the points $\boldsymbol{C}^w(0)$ and $\boldsymbol{C}^w(1)$ are to coincide exactly with the end control points $\boldsymbol{P}_0^w$ and $\boldsymbol{P}_N^w$, respectively. Also recall from Eq. (4) that, analogously to a NURBS volume, $\boldsymbol{C}^w(\xi)$ must be projected onto the hyperplane $W = 1$ to retrieve its three-dimensional counterpart $\boldsymbol{C}(\xi)$.

The first step is to position one such axial curve $\boldsymbol{C}(\xi)$ relative to a wing. Ideally, the choice of the axial curve should reflect the design intent. For example, the axial curve in Fig. 2 lies exactly on the wing's quarter-chord curve; in fact, it is the same NURBS curve used to generate the wing geometry in the first place [45]. This is arguably the best possible scenario, since then the generation of the axial curve (and FFD volume for that matter) can be fully automated. If the knowledge of the wing geometry generation process is unavailable, as is typically the case with "legacy" geometries, then some form of approximation is inevitable. In our experience, it suffices to position the axial curve either on the leading edge, trailing edge, or quarter-chord curve of a wing (the consequences of which are discussed below); thus, in the first two cases, an axial curve can easily be generated by fitting (approximating) the surface mesh points along the relevant edge with a B-spline curve; in the latter case, an axial curve approximating the quarter-chord curve can be derived with the following simple expression: $0.75\boldsymbol{C}_{\mathrm{LE}}(\xi) + 0.25\boldsymbol{C}_{\mathrm{TE}}(\xi)$, where $\boldsymbol{C}_{\mathrm{LE}}(\xi)$ and $\boldsymbol{C}_{\mathrm{TE}}(\xi)$ are previously fitted leading and trailing edge B-spline curves, respectively. Note that this expression is easily evaluated by performing the scaling and addition operations directly on the control points of $\boldsymbol{C}_{\mathrm{LE}}(\xi)$ and $\boldsymbol{C}_{\mathrm{TE}}(\xi)$ [18].

Next, relative to the global origin O, a local orthonormal coordinate system $\{\boldsymbol{o}(\xi), \boldsymbol{x}(\xi), \boldsymbol{y}(\xi), \boldsymbol{z}(\xi)\}$ that moves along $\boldsymbol{C}(\xi)$ is introduced. For any $\xi \in [0, 1]$, let

$$\boldsymbol{o}(\xi) = \boldsymbol{C}(\xi), \qquad \boldsymbol{y}(\xi) = \frac{\boldsymbol{C}'(\xi)_\perp}{\|\boldsymbol{C}'(\xi)_\perp\|}, \tag{8}$$

where the prime denotes the first derivative with respect to ξ, and the $\perp$ symbol refers to the projected vector onto the yz plane (the reason for this projection will be clarified below). We define wing twist to be about the axial curve, and compute $\boldsymbol{z}(\xi)$ directly from its spanwise distribution. Indeed, assuming $\boldsymbol{H}(\xi)$ to be a vector-valued function satisfying $\boldsymbol{H}(\xi) \cdot \boldsymbol{y}(\xi) = 0$ for all ξ, we have

$$\boldsymbol{z}(\xi) = \frac{\boldsymbol{H}(\xi)}{\|\boldsymbol{H}(\xi)\|}. \tag{9}$$

Finally, $\boldsymbol{x}(\xi)$ is simply defined as

$$\boldsymbol{x}(\xi) = \boldsymbol{y}(\xi) \times \boldsymbol{z}(\xi). \tag{10}$$

For the special case of an untwisted wing, then $\boldsymbol{x}(\xi) = [1 \;\; 0 \;\; 0]^{\mathrm{T}}$ for all ξ (hence $\boldsymbol{y}(\xi)$ and $\boldsymbol{z}(\xi)$ appear perpendicular when looking in the x direction).

The next step is to enclose the wing inside a sufficiently large FFD lattice whose spanwise cross-sections of control points are oriented according to the local coordinate functions described by Eqs. (8–10). Specifically, let $\bar{\xi}_0, \ldots, \bar{\xi}_m$ be the parameters associated with $m + 1$ such cross-sections; then the local coordinate system associated with the first one is $\{\boldsymbol{o}(\bar{\xi}_0), \boldsymbol{x}(\bar{\xi}_0), \boldsymbol{y}(\bar{\xi}_0), \boldsymbol{z}(\bar{\xi}_0)\}$, and so forth. Also notice in Fig. 2 (where $m = 3$) how control points pertaining to the same FFD volume cross-section reside in the same plane perpendicular to the local $\boldsymbol{y}$ axis. This requirement, together with the projection of $\boldsymbol{y}(\xi)$ in Eq. (8), is necessary in order to keep wing cross-sections facing the flow field no matter what the orientation of $\boldsymbol{C}(\xi)$ becomes. As for the degree selection of the FFD splines, we typically use cubic NURBS both in the chordwise and spanwise directions, whereas in the vertical direction it suffices to use linear splines if only wing outer mold lines are considered. This choice may change in the context of aerostructural optimization.

Once the FFD volume is set up, and each one of its spanwise lattice cross-sections "attached" to a point on the axial curve, the wing's surfaces are mapped to the FFD volume (as opposed to the axial curve, as originally proposed by Lazarus *et al.* [39]). As already explained in Section II B, we embed the surfaces' control points rather than their grid points, which ensures that a valid analytical geometry is maintained throughout and allows geometry parameterization to be tightly integrated with volume mesh movement for increased computational efficiency.

At this point the axial curve can be deformed by a set of user-defined geometric design variables $\boldsymbol{v}$, followed by the FFD lattice $\boldsymbol{B}$, and finally the embedded wing $\boldsymbol{A}$, i.e. $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{B}(\boldsymbol{v}))$ in functional form. Manipulating the axial curve control points enables variations in span, sweep, and dihedral. To vary twist, chord, and sectional shape, a sequence of transformation matrices is applied separately to each FFD volume cross-section.

Specifically, let $\mathsf{T}(\boldsymbol{w})$, $\mathsf{S}(\boldsymbol{Q}, c, \boldsymbol{u})$, and $\mathsf{R}(\boldsymbol{Q}, \theta, \boldsymbol{u})$ be 4×4 translation, scaling, and rotation matrices, respectively, where $\boldsymbol{w}$ is the translation vector, $\boldsymbol{Q}$ the scaling or rotation origin, c the scaling factor, $\boldsymbol{u}$ a unit vector parallel to the scaling direction (optional) or to the axis line of rotation, and θ the angle of rotation [46]. Then, for each FFD volume cross-section $i = 0, \ldots, m$, the following three transformations are performed:

1. $\mathsf{S}(\boldsymbol{o}(\bar{\xi}_i), c)$: uniform scaling applied to the entire cross-section along the local x and z axes. The effect is to modify taper while keeping a constant thickness-to-chord ratio. The design variable is c.

2. $\mathsf{S}(\boldsymbol{o}(\bar{\xi}_i), c, \boldsymbol{z}(\bar{\xi}_i))$: nonuniform scaling applied to individual control points along the local z axis. The effect is to modify sectional shape. The design variable is c.

3. $\mathsf{R}(\boldsymbol{o}(\bar{\xi}_i), \theta, \boldsymbol{y}(\bar{\xi}_i))$: rotation applied to the entire cross-section about the local y axis. The effect is to modify twist. The design variable is θ.

It is then necessary to reorient the FFD volume cross-sections according to the orientation of the new axial curve, hereafter denoted as $\boldsymbol{C}^*(\xi)$. The design variables are the homogeneous coordinates of the axial curve control points, i.e. the $\{\boldsymbol{P}_i^w\}_{i=0}^N$ in Eq. 7. The effect is to modify span, sweep, and/or dihedral. The final two transformations, again performed on each FFD volume cross-section $i = 0, \ldots, m$, are:

4. $\mathsf{T}(\boldsymbol{o}^*(\bar{\xi}_i) - \boldsymbol{o}(\bar{\xi}_i))$: translation applied to the entire cross-section. The effect is to reposition the local coordinate system to $\boldsymbol{o}^*(\bar{\xi}_i) = \boldsymbol{C}^*(\bar{\xi}_i)$.

5. $\mathsf{R}(\boldsymbol{o}^*(\bar{\xi}_i), \angle\, \boldsymbol{y}(\bar{\xi}_i)\boldsymbol{y}^*(\bar{\xi}_i), \boldsymbol{u})$: rotation applied to the entire cross-section about the unit vector $\boldsymbol{u} = [1 \quad 0 \quad 0]^{\mathrm{T}}$, where the $\angle$ symbol refers to the signed angle between $\boldsymbol{y}(\bar{\xi}_i)$ and $\boldsymbol{y}^*(\bar{\xi}_i) = \boldsymbol{C}^{*\prime}(\bar{\xi}_i)_\perp / \|\boldsymbol{C}^{*\prime}(\bar{\xi}_i)_\perp\|$. The effect is to reorient the local coordinate system so that its xz plane remains perpendicular to the axial curve.

The FFD volume control point weights, should they be chosen as design variables, do not affect any of the above transformations, whereas the axial curve control point weights affect transformations 4 and 5 only. Also, since transformations 1–3 are taken about the local origins, $\{\boldsymbol{o}(\bar{\xi}_i)\}_{i=0}^{m}$, it should be clear that the position of the axial curve relative to the wing matters; if, say, twist about the trailing edge is desired, then the axial curve should be positioned accordingly.

By judiciously choosing the number and placement of control points of both the axial curve and attached FFD volume, as well as their degree and knot vector, it is possible to achieve any combination of linear or nonlinear variations in span, sweep, dihedral, twist, chord, and sectional shape. We also emphasize the fact that the sole purpose of the axial curve is to “drive” the movement of the FFD volume. Therefore both entities are completely decoupled as far as their mathematical definitions are concerned. This is necessary to decouple the featured-based, high-level deformations provided by the axial curve from the detailed, low-level deformations provided by the FFD volume. For example, consider a straight wing subject to sweep, span, and dihedral changes, as well as sectional shape changes. A linear axial curve defined with only two control points is sufficient to handle all planform changes, regardless of the number of FFD volume cross-sections used to control sectional shape. Of course, the same effect could be achieved without the axial curve, but at the cost of many constraints. Hence, not only is axial-curve-based wing deformation intuitive, it also greatly simplifies problem setup.

It is possible to connect multiple axial curves one after another to accommodate cranked wings or to achieve custom deformations on a single continuous wing segment as will be demonstrated in Section IV C. However, if the relative orientation of any two adjoining axial curves changes at their connection point, then, as a result of the transformation process described above, the two FFD volume cross-sections meeting there will no longer overlap, possibly opening up cracks in the geometry. In such cases it is necessary to “clamp” back the two FFD volume cross-sections onto each other. This last operation should be accomplished through shearing, otherwise the quality of the wing surfaces at the connection may be compromised.

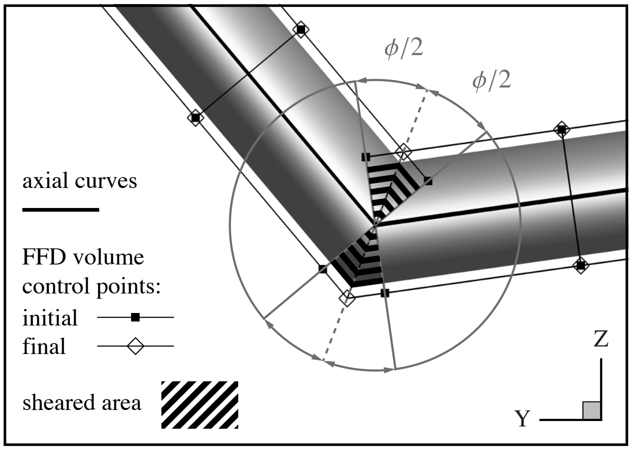

Fig. 3 Reconnecting two wings through shearing.

Specifically, let $\mathsf{H}(\boldsymbol{u}, \boldsymbol{Q}, \boldsymbol{w}, \phi)$ be a 4×4 shearing matrix, where $\boldsymbol{u}$ and $\boldsymbol{Q}$ are respectively a unit vector and a point defining the shearing plane, $\boldsymbol{w}$ is a unit vector parallel to the shearing plane and perpendicular to $\boldsymbol{u}$, and ϕ is the shearing angle [47]. Then, referring to Fig. 3, a shearing plane for the left (L) wing is defined with the vector $\boldsymbol{z}_{\mathrm{L}}(0)_{\perp}$ (here the $\perp$ symbol refers to the untwisted vector) and point $\boldsymbol{o}_{\mathrm{L}}(0)$, and a shearing direction with the vector $\boldsymbol{y}_{\mathrm{L}}(0)$. Similarly, a shearing plane for the right (R) wing is defined with the vector $\boldsymbol{z}_{\mathrm{R}}(1)_{\perp}$ and point $\boldsymbol{o}_{\mathrm{R}}(1)$, and a shearing direction by $\boldsymbol{y}_{\mathrm{R}}(1)$. For a signed angle ϕ between $\boldsymbol{y}_{\mathrm{L}}(0)$ and $\boldsymbol{y}_{\mathrm{R}}(1)$, a shear of $\phi/2$ is applied to the entire first (last) cross-section of the left (right) FFD volume. Specifically, the shearing operator for the left FFD volume is $\mathsf{H}(\boldsymbol{z}_{\mathrm{L}}(0)_{\perp}, \boldsymbol{o}_{\mathrm{L}}(0), \boldsymbol{y}_{\mathrm{L}}(0), \phi/2)$ and $\mathsf{H}(\boldsymbol{z}_{\mathrm{R}}(1)_{\perp}, \boldsymbol{o}_{\mathrm{R}}(1), \boldsymbol{y}_{\mathrm{R}}(1), -\phi/2)$ for the right FFD volume. Due to the local support property of NURBS, the wing surfaces will shear even far outside the area shown in Fig. 3. That is not to say that the surfaces there will necessarily change *geometrically* (as opposed to parameterically). A simple fix to limit the extent of the sheared area is to add more FFD volume cross-sections immediately inboard and outboard of the connection.

Finally, axial curves can also be connected in numerous different ways depending on the topology of the wing system under consideration. In such cases, the shearing operation just described may not apply; however, depending on the desired effects, care must still be taken at junctions when choosing the right FFD volume in which to embed the surface control points. Refer to [40] where a strut-braced wing is optimized following this concept. The reader is warned that problem setup for arbitrary wing systems is highly case dependent, which is inevitable given that FFD is by itself (i.e. without appropriate constraints or external mechanisms) ill-suited to handle the general deformation of surface-surface intersections such as those found at wing-wing and wing-fuselage junctions.

III. Adjoint-Based Optimization Methodology

The two-level FFD and axial deformation schemes described in Section II are part of a broader ASO methodology (known as Jetstream) thoroughly explained and tested in [38]. An important part of this methodology is the gradient-based optimization package SNOPT [48].

SNOPT is based on the sequential quadratic programming paradigm, where the Hessian of the Lagrangian is approximated using the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno [49]. It can handle large numbers of design variables and constraints, provided all gradient entries are supplied by the user. Strictly speaking, this requirement is optional since SNOPT can finite-difference them; however this would not only be prohibitive but would also yield values subject to round-off and truncation errors.

For constraints that do not depend on the flow, such as projected area and volume, the gradients are mostly hand-differentiated. Otherwise, they are calculated analytically by using discrete-adjoint variables. The next two subsections briefly review how this is accomplished, starting with the evaluation of the aerodynamic functionals.

A. Function Evaluation

Once a geometry has changed shape as a result of an optimization iteration, and the surrounding volume mesh has adapted to it, a function evaluation is performed. The two aerodynamic functionals considered in this work are lift L and drag D, both divided by the freestream dynamic pressure q_∞. Note that all reported values, as well as reference areas S, are based on half geometries. Each functional is computed by integrating the pressure field bounding the aerodynamic surfaces, thus we may write $\mathcal{J} = \mathcal{J}(\alpha, \boldsymbol{b}, \boldsymbol{q})$, where $\mathcal{J}$ is either L/q_∞ or D/q_∞, α is the freestream angle of attack, and $\boldsymbol{q}$ is the discrete solution of the equations governing fluid flow. The dependence of $\mathcal{J}$ on $\boldsymbol{b}$ is through the surface normal (recall from Section II B that $\boldsymbol{b}$ represents the volume control points responsible for the mapping of the grid).

To solve for the flow solution $\boldsymbol{q}$, we ignore temporal and viscous effects and consider the steady Euler equations only. The spatial derivatives are discretized with second-order accurate finite-difference summation-by-parts operators, while simultaneous approximation terms enforce boundary conditions and couple blocks, requiring only C^0 continuity between matching grid lines at interfaces [50, 51]. To stabilize the solution in the vicinity of shocks, the discretization is augmented by a pressure switch mechanism involving both second- and fourth-difference scalar dissipation [52]. The discretization produces a set of nonlinear algebraic equations represented by the vector equation

$$\mathcal{R}(\alpha, \boldsymbol{b}, \boldsymbol{q}) = \mathbf{0}. \tag{11}$$

Here, the flow residuals depend on α through the boundary terms and on the control points $\boldsymbol{b}$ through the grid metrics. With a sufficiently good initial iterate $\boldsymbol{q}^{(0)}$ obtained through a globalization technique, Eq. (11) is converged using Newton's method, where at each iteration a system of linear equations is solved inexactly by a parallel Schur-preconditioned Jacobian-free Krylov iterative method [53].

B. Gradient Evaluation

To evaluate the total gradient of an aerodynamic functional $\mathcal{J}$ with respect to the geometric design variables $\boldsymbol{v}$ and angle of attack α, it is helpful to recast the problem as the minimization of $\mathcal{J}$ such that the mesh and flow residuals (Eqs. (6) and (11), respectively) are converged to a small enough tolerance (by roughly 12 orders in practice). Typical of such a partial-differential-equation-constrained optimization problem, a Lagrangian is introduced:

$$\mathcal{L} = \mathcal{J}\Big(\alpha, \boldsymbol{b}, \boldsymbol{q}\Big) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{\mathcal{M}}\Big(\boldsymbol{A}(\boldsymbol{B}(\boldsymbol{v})), \boldsymbol{b}\Big) + \boldsymbol{\psi}^{\mathrm{T}} \mathcal{R}\Big(\alpha, \boldsymbol{b}, \boldsymbol{q}\Big), \tag{12}$$

where the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$ are the vectors of mesh- and flow-adjoint variables, respectively. As usual the first-order (necessary) optimality conditions are obtained by setting the partial derivatives of $\mathcal{L}$ to zero [49]. This leads to a sequential formulation of the gradient calculation that is reminiscent of the reverse mode of algorithmic differentiation, where the high-level variables $\boldsymbol{b}$ and $\boldsymbol{q}$ are regarded as intermediate variables [37]. The last two partials of $\mathcal{L}$, i.e. those with respect to $\boldsymbol{v}$ and α, are the desired gradients:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\boldsymbol{v}} = \boldsymbol{\lambda}^{\mathrm{T}} \left.\frac{\partial \boldsymbol{\mathcal{M}}}{\partial \boldsymbol{A}}\right|_{\boldsymbol{b}} \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{B}} \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{v}}, \tag{13}$$

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\alpha} = \left.\frac{\partial \mathcal{J}}{\partial \alpha}\right|_{\boldsymbol{b},\boldsymbol{q}} + \boldsymbol{\psi}^{\mathrm{T}} \left.\frac{\partial \mathcal{R}}{\partial \alpha}\right|_{\boldsymbol{b},\boldsymbol{q}}. \tag{14}$$

A detailed description of the specialized solution strategies employed to solve for $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$ is given in [38].

The term $\partial \boldsymbol{A}/\partial \boldsymbol{B}$ in Eq. (13) corresponds to the sensitivity of the embedded surface control points to the

FFD lattices, and is most easily derived from the rational form of Eq. (2):

$$\frac{\partial \boldsymbol{V}(\boldsymbol{\xi})}{\partial \boldsymbol{P}_{ijk}} = \frac{\mathcal{N}_i^{(p)}(\xi)\,\mathcal{N}_j^{(q)}(\eta)\,\mathcal{N}_k^{(r)}(\zeta)\,w_{ijk}}{\sum_{d=0}^{N_i}\sum_{e=0}^{N_j}\sum_{f=0}^{N_k}\mathcal{N}_d^{(p)}(\xi)\,\mathcal{N}_e^{(q)}(\eta)\,\mathcal{N}_f^{(r)}(\zeta)\,w_{def}}\,\mathsf{I}_3, \tag{15}$$

where I_3 is the rank 3 identity matrix. The partial derivatives with respect to the control point weights, $\partial \boldsymbol{V}(\boldsymbol{\xi})/\partial w_{ijk}$, are not as straightforward, but can still be easily derived by repetitive application of the quotient rule. Note that if the weights are not included as design variables then $\partial \boldsymbol{A}/\partial \boldsymbol{B}$ is constant and must be evaluated only once.

As for the term $\partial \boldsymbol{B}/\partial \boldsymbol{v}$, the partials of the FFD lattice(s) with respect to the geometric design variables, it is implemented in the present algorithm using the complex-step method [54], where each design variable, whether it be for example the coordinate of an axial curve control point or the rotation angle of an entire FFD volume cross-section, is perturbed one at a time by an imaginary step size $h = 10^{-20}$. The computational cost of this approach is negligible compared with the cost of computing the adjoint variables.

IV. Lift-Constrained Drag Minimizations

The axial-deformation and FFD framework augmenting the aerodynamic optimizer is now demonstrated through three drag minimization problems, each subject to a lift equality constraint. For a fixed span b (and in the absence of wave drag), the problem is thus equivalent to maximizing span efficiency

$$e = \frac{(L/q_\infty)^2}{\pi b^2 (D/q_\infty)}. \tag{16}$$

Since this metric is easily interpreted — the reference value $e = 1$ corresponding to an elliptically loaded planar wing — we sometimes supplement values of C_D or D/q_∞ with e. All reported values are based on flow solutions obtained on refined grids of roughly $O(10^8)$ nodes. This is made possible by refining, in parameter space, the NURBS volumes parameterizing each computational grid, an approach that also ensures that the surface nodes remain consistent with the geometries.

All three geometries were generated by a specialized in-house drawing tool that outputs outer mold lines in the form of smooth networks of NURBS surfaces [45]. The surfaces were output in IGES format and subsequently tesselated inside multi-block H-H grids using the commercial mesh generator ANSYS ICEM CFD.

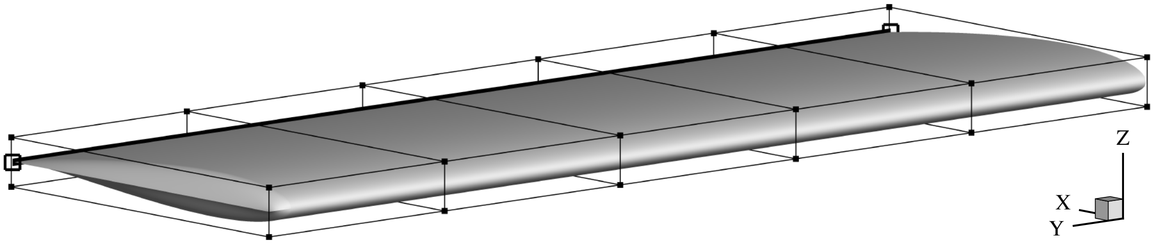

Fig. 4 Wing twist validation case setup.

A. NACA-0012 Wing Twist Validation

We start by validating our proposed axial deformation technique against a benchmark test case recognized by the ASO community. Specifically designed for that purpose is the Twist Optimization Case of the AIAA Aerodynamic Design Optimization Discussion Group. It consists of minimizing the induced drag of a rectangular planar wing by only enabling twist design variables while constraining the lift coefficient to a fixed value. A similar case is found in [55], which is the one emulated here.

The half wing geometry and axial curve setup are shown in Fig. 4. The wing is parameterized with a sharp NACA-0012 airfoil and is initially untwisted, untapered, unswept, and has zero dihedral. The semispan to root chord ratio is 3, hence the semispan projected area is initially 3 root-chord units squared. The inboard section extends 97.5% of the semispan, at which point a wingtip pinching the lower and upper surfaces covers the remaining 2.5%. As per the recommendations of the Twist Optimization Case, twist is applied about the trailing edge, which is easily achieved here by placing the axial curve directly on the wing's trailing edge (recall from Section II C that twist is attained by rotating the FFD volume cross-sections about the axial curve). The only design variables are the rotation angle of each FFD volume cross-section, of which there are 6 in Fig. 4. The FFD volume used here is a B-spline volume that is cubic in the spanwise direction and linear in the other two directions; thus, wing twist is effectively parameterized by a cubic B-spline.

The freestream Mach number is set at 0.5 and the angle of attack is set such that the lift coefficient of $C_L = 0.375$ (given as an equality constraint to SNOPT) is closely met at the start of the optimization. As described above, all FFD volume cross-sections are allowed to rotate independently from each other, including the root section, hence the angle of attack is not included as a design variable to ensure a unique set of design variables at the optimum.

The spanwise lift distributions of both the initial and twist optimized wings are shown in Fig. 5. The

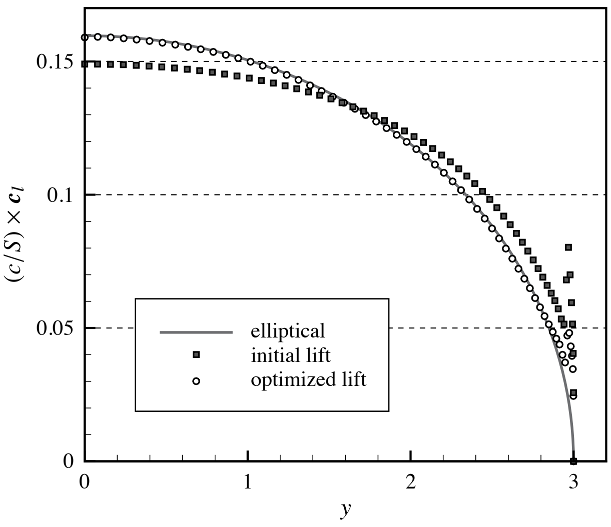

Fig. 5 Spanwise lift distribution of the initial and optimized wings.

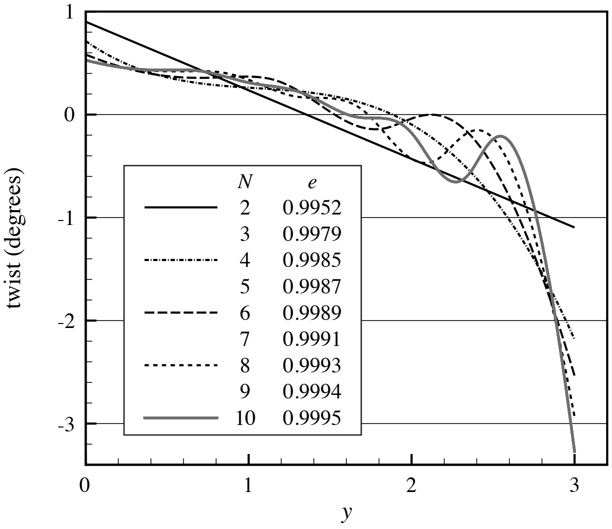

Fig. 6 Effect of the number of FFD volume cross-sections on the spanwise twist distribution.

lift distribution of the optimized wing closely approximates an elliptical distribution, except at its tip where side-edge separation occurs [55]. The final coefficient of drag is $C_D = 0.00753$, yielding a span efficiency of $e = 0.9989$, in excellent agreement with linear theory. For comparison, the coefficient of drag and span efficiency of the initial configuration are $C_D = 0.00769$ and $e = 0.9780$, respectively.

To investigate the effect of the number of spanwise FFD volume cross-sections on twist distribution and span efficiency, we repeatedly performed the same optimization with the number of cross-sections ranging from 2 to 10. Given the definition of a NURBS volume, Eq. (3), only linear and quadratic spanwise twist variations could possibly be achieved for the cases $N = 2$ and $N = 3$, respectively.

As expected, increasing the number of FFD volume cross-sections leads to higher span efficiencies. More subtle is the following general trend, visible in Fig. 6: as the number of cross-sections increases, the more wavy the twist distributions become, especially toward the wingtip. Clearly, in the limit the optimal twist distribution is not elliptical (which theoretically corresponds to $e = 1$); rather, here $e > 1$ can be expected (which we observed for $N > 13$) as the optimizer is given more and more freedom in maximizing the twist rate of change at the tip, i.e. in forming a mini winglet. As pointed out by Hicken and Zingg [55], who also observed a nonelliptical twist distribution at optimality, this behavior can be attributed to side-edge separation occurring at the wingtip. Side-edge separation is a real flow phenomenon, although the prediction of the exact location and strength of the tip vortices with an Euler code is debatable.

B. Feathered Winglet Shape Exploration

Nonplanar wingtip devices, if designed properly as an integral part of a wing, can significantly reduce induced drag [56]. These savings can be reasonably predicted by analytical methods based for example on the potential-flow equations, but only if the lifting line remains continuous along the entire span, as it is the case for simple C- and closed-wing systems. Analytical methods, however, fail to predict the induced drag generated by closely staggered wings, mainly because they cannot accurately model the highly nonlinear effects produced by those systems. This fact is exposed by the work of La Roche and La Roche [57], who showed that the span efficiency of their “Winggrid” device (inspired by the tip feathers of a bird’s wing) is $e = 1.91$ when measured experimentally, while the same device registers $e = 0.80$ when predicted with standard linear theory. It is thus not surprising that until now most research aimed at exploiting the benefits of avian flight has been primarily conducted by experimentation.

The work of Spillman [58] in 1978 revealed that, when appropriately twisted and cambered, a series of small sails spiralled at the tip of a wing can reduce induced drag by as much as 30%. Later, Smith *et al.* [59] investigated the influence of longer, multiple winglets on total drag, and found that negative incidence and twist of the winglets re-orient the lift vector forward and thus cancel part of the drag. They also concluded that a dihedral spread of the winglets localizes the tip vortices away from the wing plane, thereby reducing the effective downwash, and, consequently, drag. Similar observations are reported by Miklosovic [60],

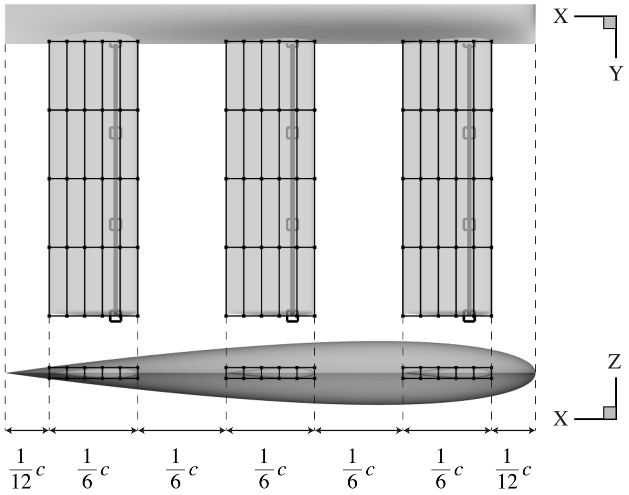

Fig. 7 Feathered winglet shape exploration case setup.

who found that three winglets separated by maximum dihedral can achieve 54% improvement in total drag at equivalent lift coefficients. Encouraged by these results, we now perform the numerical shape optimization of a wingtip extension consisting of three feathers staggered in the streamwise direction. Contrary to the above experimental investigations, we give the optimizer much greater freedom in exploring the design space.

The initial setup is based on the same rectangular wing geometry used for the twist validation case of Section IV A, where the semispan and chord are respectively, in root-chord units, 3 and 1 (see Fig. 4). The wingtip is also the same, except for the three feathers that now protrude in the spanwise direction by an equal amount of $1/2$ root-chord units. As shown in Fig. 7, the chord of each feather is $1/6$ root-chord units, which is also the distance that separates the feathers from each other. Just like the main wing, each feather is constructed from the NACA-0012 airfoil, so at the start of the optimization the entire system is perfectly symmetric in the vertical (z) direction.

A total of four axial curves are used to manipulate each lifting surface independently. The first axial curve is again the same one that is depicted in Fig. 4; it is linear, falls exactly on the main wing's trailing edge, and controls an FFD volume made up of 6 spanwise cross-sections. The other three axial curves are each assigned to one feather; each is cubic with four control points, falls exactly on its respective feather's quarter-chord line, and controls an FFD volume made up of 5 spanwise cross-sections. This last set of FFD volumes are cubic in the chordwise direction, and are made up of 6 control points in that direction.

All but the last FFD volume cross-section pertaining to the main wing are free to rotate, and all but the first two FFD volume cross-sections pertaining to the feathers are free to rotate. The rationale is to decouple the relative motion between all four wing segments by maintaining their common fairing free of twist. To compensate, we activate the angle of attack as a design variable, and choose its initial value such the lift constraint of $C_L = 0.375$ at Mach 0.5 is met almost exactly at the beginning of the optimization. The optimizer is also free to change the feathers' section shape by scaling the FFD volume control points by no less than 0.5 and no more than 2.0 relative to the feathers' symmetry plane. This is with the exception of the first (inboard) FFD volume cross-section of each feather, whose control points are completely frozen. Thus freezing the FFD volume control points is necessary to maintain smooth surfaces at the fairing.

Nonlinear dihedral is allowed to develop on each feather by enabling the z-coordinate of the last control point of each axial curve responsible for a feather to vary by as much as plus or minus $0.35c$. The wingspan is therefore fixed at $7c$, so the maximum achievable height-to-span ratio, h/b, is 0.1 (0.2 with respect to the semispan of $3.5c$). The other three inboard control points of the same axial curves are frozen; again, given the property of cubic B-splines, this is necessary in order to maintain smooth surfaces at the fairing. Summarizing, there are 14 twist design variables, 144 scaling design variables, 3 z-directional axial curve control point design variables, for a total of 162 geometric design variables, plus the angle of attack.

SNOPT took 101 major iterations to reduce optimality from 1.8×10^{-3} to 7.9×10^{-6}, reducing drag by 16% to $C_D = 0.00501$, corresponding to a span efficiency of $e = 1.193$ at the final angle of attack of approximately 3.31 degrees. In comparison, according to lifting line theory (when airfoil thickness is accounted for) a box-wing with the same height-to-span ratio of 0.1 is $e = 1.180$ [55]. The feathered wing also proves to be superior to the split-tip configuration investigated by Hicken and Zingg [55], who obtained $e = 1.167$ for the same h/b. In the future, it would be interesting to study the effect of the number of feathers and their chordwise interspacing on span efficiency.

Figure 8 provides a frontal view of the optimal shape, zoomed in around the wingtip region. The optimizer reached both the maximum and minimum allowable vertical bounds related with the dihedral of the fore and aft feathers, respectively. The tip location of the middle feather has moved down, and is therefore closer to the aft feather than the fore feather. The spanwise lift distributions of all three feathers are shown in Fig. 9; notice how the fore feather carries significantly more lift than the other two. Interestingly, the main wing does not follow an elliptical distribution. Even more interesting are the spanwise drag distributions

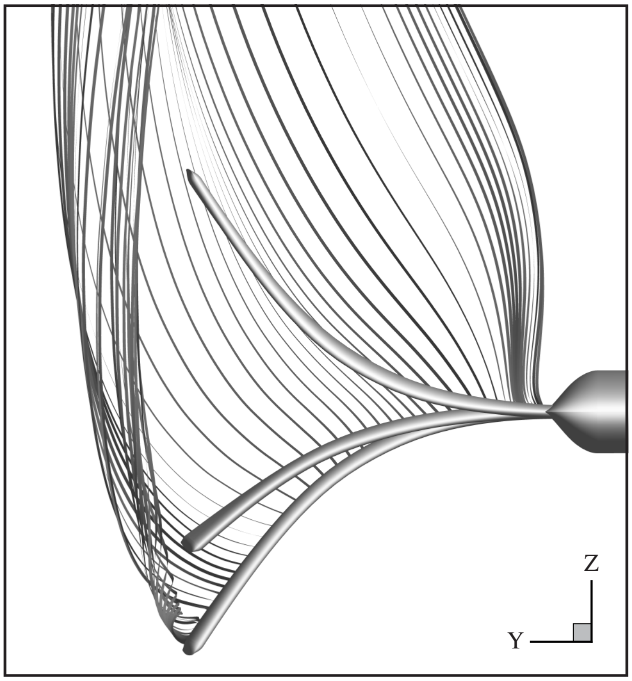

Fig. 8 Frontal view of the optimized feathered winglet with streamtraces.

also plotted on Fig. 9. The fore feather produces thrust over 80% of its span. Moreover, the middle feather produces considerably less drag than the aft feather, suggesting a cascading effect where each aft feather generates a favorable flow field about its fore feather. Also notice, in Fig. 8, how small is the tip vortex core emanating from the aft feather's tip, an indicator that roll-up is minimized and that high circulation regions are diffused [60].

C. Regional Jet Winglet Generation

We now test the algorithm in a more practical, yet even more challenging setting. The geometry of interest is a baseline wing inspired from the Bombardier CRJ700 W34 planform [61]. It has a straight leading edge swept back 30 degrees with a 2 degree dihedral and a root plug ending at 40% span. The chord lengths at the kink and tip sections are 0.59 and 0.21 root-chord units, respectively. The wing surfaces are linearly interpolated from the NASA-0614, -0612, and -0410 supercritical phase 2 airfoils located at the root, kink, and tip sections of the wing, respectively. When immersed in a Mach 0.78 inviscid freestream at zero angle of attack the wing experiences a lift coefficient of $C_L = 0.53713$ based on its projected area $S = 1.1581$ root-chord units squared.

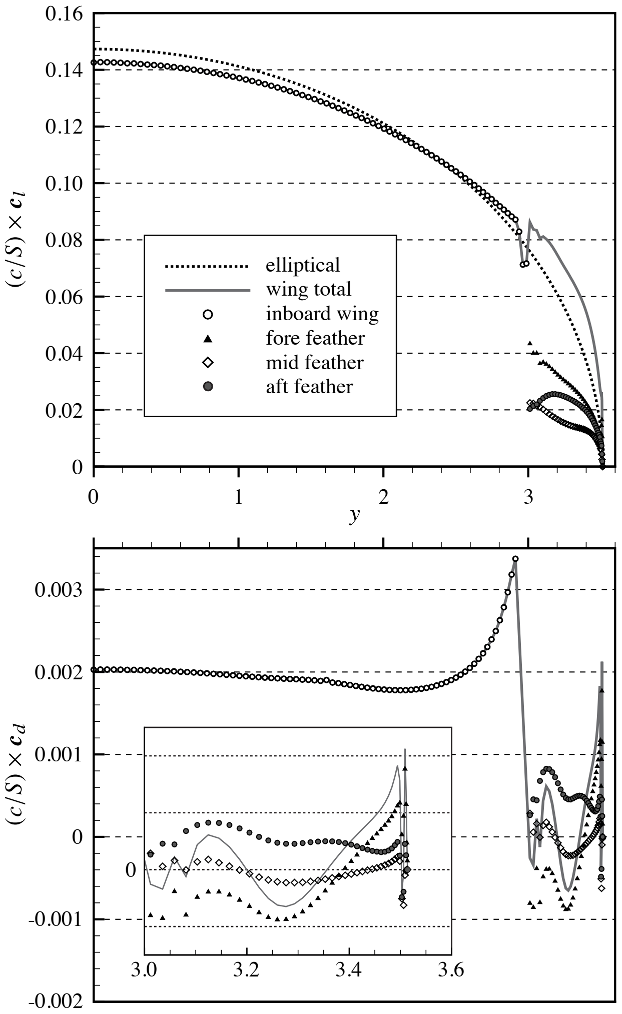

Fig. 9 Spanwise lift and drag distributions of the optimized wing and winglet.

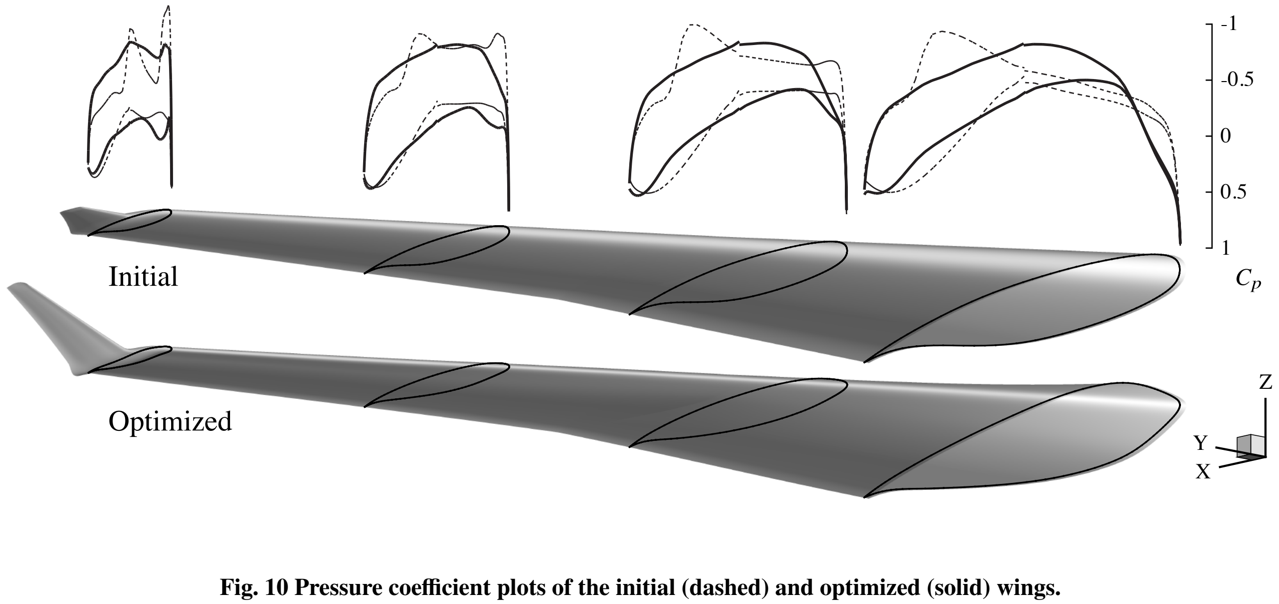

Fig. 10 Pressure coefficient plots of the initial (dashed) and optimized (solid) wings.

The motivation for this experiment is twofold. First, it is desired to completely remove the shocks initially present over much of the upper surfaces of the baseline wing. As is visible from the pressure coefficient plots on Fig. 10, this includes an acute double shock pattern located toward the wingtip. Second, it is desired to confirm whether the optimizer, if given the freedom, will generate a winglet on its own starting from a swept-forward, linearly tapered wingtip extension (Fig. 10, upper wing). Generating a nonplanar wing from a planar one is a stiff test of both the geometry control methodology and the mesh movement algorithm. Note that the taper of the wingtip extension is not a design variable; it is applied manually on the baseline wing at the beginning of the optimization.

Since the wing can be broken down into 4 logical units, i.e. the inboard, outboard, transition, and wingtip segments, we use 4 axial curves. They are all attached one after another along the leading edge of their respective segment. We choose linear axial curves, except for the one that provides the smooth outboard-wingtip transition, which is cubic and has 4 control points. The chordwise and vertical extents of the wingtip extension are controlled by only 2 design variables: the x- and z-coordinates of the outermost axial curve control point. The optimizer is free to increase any one of these two coordinates by as much as $1/6$ root-chord units. Two linear constraints are used to keep the axial curve control point connecting the transition-wingtip axial curves in-line with the two axial curve control points that sandwich it, which includes the (free) outermost axial curve control point. These constraints follow from the properties of B-splines and are necessary to maintain a smooth transition. All other axial curve control points are frozen; thus, except for the wingtip extension, it is not possible for the wing segments to vary in sweep or dihedral, nor is it possible for the wing to vary in overall span.

The number of FFD volume cross-sections assigned to each axial curve is 4, 6, 4, and 5, respectively. Accounting for overlapping cross-sections there are thus a total of 16 cross-sections; all are free to rotate (twist). Each FFD volume cross-section is respectively made up of 10 chordwise and 2 vertical control points; all are free to scale vertically in their respective local coordinate system. The angle of attack is fixed at 0, therefore there are 16 twist plus 320 section design variables, for a total of 336 geometric design variables. Each FFD volume is cubic in the chordwise and spanwise directions and linear in the vertical direction. Finally, aside from the lift constraint of $L/q_\infty = 0.618$ root-chord units squared, the evolving wing shape is forced to at least maintain its initial internal volume of $V = 0.065$ root-chord units cubed, which is

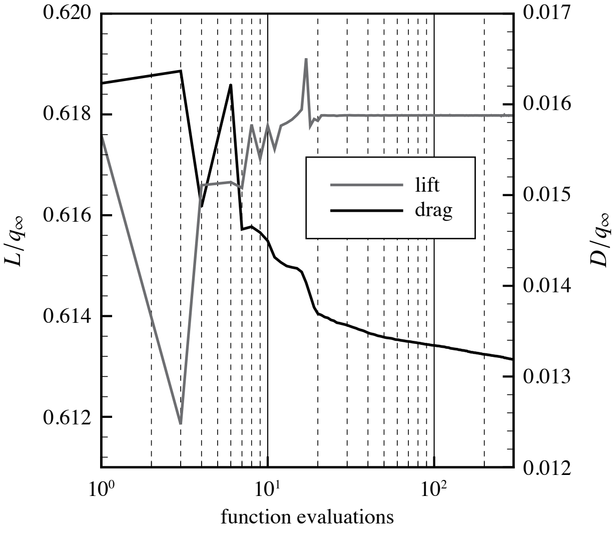

Fig. 11 Lift and drag convergence histories.

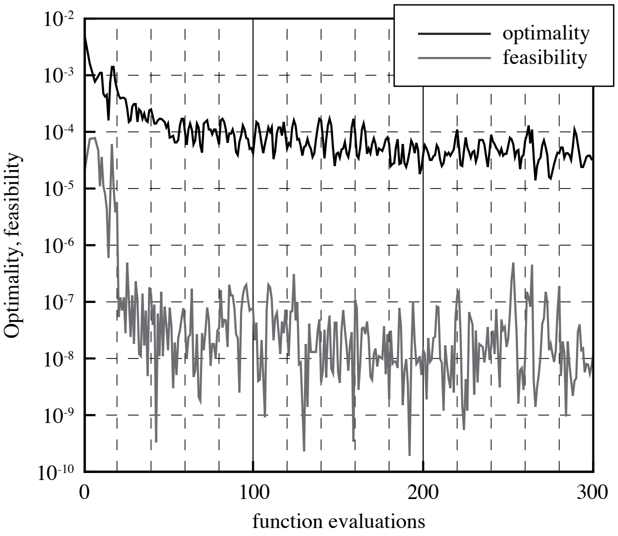

Fig. 12 Optimality and feasibility convergence histories.

also given to SNOPT in the form of a nonlinear inequality constraint.

As depicted by the bottom wing in Fig. 10, the optimizer successfully generated a winglet-up configuration. In fact, it swept back and sheared the winglet upward as much as it could. The optimizer also heavily remodeled the wing sectional shape, thus making the suction side completely shock free. The very rapid pressure recovery is commonplace of Euler-based optimizations; in viscous flow, the same wing would likely exhibit separation.

In all, SNOPT achieved an 18.8% drag reduction in 300 function calls, at which point we terminated the optimization for practical reasons. The final drag coefficient is $C_D = 0.01161$, which corresponds to a span efficiency of $e = 1.0581$. A further decrease in drag can be achieved by continuing the optimization further, but the improvement is likely to be small. It is worth pointing out that SNOPT achieved a 25 drag-count reduction in the first 20 design iterations, compared to the 5 additional counts in the remaining 280 iterations; see Fig. 11. Also see Fig. 12, where optimality (a measure of the Lagrangian gradient) and feasibility (a measure of the constraint violation) are shown to decrease more rapidly at the start of the optimization.

V. Conclusions

This paper described an intuitive and versatile technique that parameterizes wing shape changes. It uses the principles of FFD combined with those of axial deformation to respectively achieve local and global control.

Unlike previous work, here it is the surface control points that are embedded inside the FFD volume(s). The primary advantage is to maintain an analytic surface representation throughout an optimization. Another advantage, this one intrinsic to FFD, is to decouple the surface topology from the parameterization used in an optimization problem, thus offering a consistent parameterization when comparing two different geometries, or even two identical geometries but with a different surface topology.

Any form of span, sweep, dihedral, taper, twist, and sectional shape changes can be easily achieved by means of simple transformation matrices, an approach that promotes an easy and thorough exploration of the design space. The technique enables very large shape changes and is directly extensible to multi-segmented wings of arbitrary topology. Together, these characteristics are advantageous for the systematic study of both conventional and unconventional lifting systems.

The details of integrating the proposed axial deformation scheme with an existing gradient-based aerodynamic optimizer were also provided. Assuming the exact sensitivities of the aerodynamic functionals with respect to the surface grid are available, then, by virtue of the chain rule, exact *total* sensitivities of those functionals with respect to the design variables can be easily computed given the simplicity with which axial deformation can be differentiated. This fact remains true for any other shape optimizer, including those involving fully coupled aerostructural analysis.

Table 1 Aerodynamic performance of the optimized configurations

Geometry	C_L	C_D	Λ	e	Grid nodes
Twist	0.37674	0.00753	6.0011	0.9989	101,291,680
Feather	0.37630	0.00501	7.5271	1.1930	100,298,156
CRJ700	0.53992	0.01161	7.5506	1.0581	99,476,256

Three cases demonstrating the applicability of axial deformation to ASO were presented; refer to Table 1 where they are summarized. The first case validated the accuracy and efficiency of the framework by recovering an elliptical lift distribution on a rectangular wing subject to twist. With only 2 design variables, the span efficiency converged from $e = 0.9780$ to $e = 0.9952$ in 7 function calls. The second case underlined the intuitiveness of axial deformation by applying it to the optimization of a three-pronged feathered winglet. The parameterization proved to be flexible enough to recapture important experimental results, and it did so automatically without human intervention beyond the initial problem setup. Finally, the third case showcased axial deformation in an industrial context by optimizing a realistic wing in transonic flow while letting the optimizer develop a winglet on its own.

Future work will apply the flexibility and efficiency of axial-curve-driven FFD to the high-fidelity aerodynamic design of unconventional aircraft for minimum fuel consumption. Special consideration will also be given to wing-fuselage junctions and the like, expanding the axial-deformation framework toward a full-fledged geometry control system.

Acknowledgments

The authors are grateful for the financial support provided by the Ontario Graduate Scholarship. Computations were performed on the General Purpose Cluster supercomputer at the SciNet High Performance Computing Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund – Research Excellence; and the University of Toronto.

References

[1] Lee, D. S., Pitari, G., Grewe, V., Gierens, K., Penner, J. E., Petzold, A., Prather, M. J., Schumann, U., Bais, A., Berntsen, T., Iachetti, D., Lim, L. L., and Sausen, R., “Transport impacts on atmosphere and climate: Aviation,” *Atmospheric Environment*, Vol. 44, No. 37, 2010, pp. 4678–4734,
doi:10.1016/j.atmosenv.2009.06.005.

[2] Jameson, A., “Aerodynamic design via control theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260,
doi:10.1007/BF01061285.

[3] 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.

[4] Haimes, R. and Follen, G. J., “Computational Analysis PRogramming Interface,” in “6th Numerical Grid Generation in Computational Field Simulations,” University of Greenwich, United Kingdom, 1998.

[5] Fudge, D., Zingg, D. W., and Haimes, R., “A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimization,” in “43rd AIAA Aerospace Sciences Meeting and Exhibit,” AIAA Paper 2005-0451, Reno, Nevada, 2005,
doi:10.2514/6.2005-451.

[6] Requicha, A. G. and Voelcker, H. B., “Boolean operations in solid modeling: Boundary evaluation and merging algorithms,” *Proceedings of the IEEE*, Vol. 73, No. 1, 1985, pp. 30–44,
doi:10.1109/PROC.1985.13108.

[7] Price, M., Raghunathan, S., and Curran, R., “An integrated systems engineering approach to aircraft design,” *Progress in Aerospace Sciences*, Vol. 42, No. 4, 2006, pp. 331–376,
doi:10.1016/j.paerosci.2006.11.002.

[8] Samareh, J. A., “Status and Future of Geometry Modeling and Grid Generation for Design and Optimization,” *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 97–104,
doi:10.2514/2.2417.

[9] Samareh, J. A., “Geometry and Grid/Mesh Generation Issues for CFD and CSM Shape Optimization,” *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 21–32,
doi:10.1023/B:OPTE.0000048535.08259.a8.

[10] Nemec, M. and Aftosmis, M. J., “Aerodynamic Shape Optimization Using a Cartesian Adjoint Method and CAD Geometry,” in “24th AIAA Applied Aerodynamics Conference,” AIAA Paper 2006-3456, San Francisco, California, 2006,

doi:10.2514/6.2006-3456.

[11] Jones, W. T., Lazzara, D., and Haimes, R., “Evolution of Geometric Sensitivity Derivatives from Computer Aided Design Models,” in “13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference,” AIAA Paper 2010-9128, Fort Worth, Texas, 2010,
doi:10.2514/6.2010-9128.

[12] 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.

[13] Robinson, G. M. and Keane, A. J., “Concise Orthogonal Representation of Supercritical Airfoils,” *Journal of Aircraft*, Vol. 38, No. 3, 2001, pp. 580–583,
doi:10.2514/2.2803.

[14] Sobieczky, H., “Parametric Airfoils and Wings,” in Fujii, K. and Dulikravich, G. S., eds., “Recent Development of Aerodynamic Design Methodologies,” Vieweg+Teubner Verlag, Vol. 65 of *Notes on Numerical Fluid Mechanics (NNFM)*, pp. 71–87, 1999,
doi:10.1007/978-3-322-89952-1_4.

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

[16] Bloor, M. I. G. and Wilson, M. J., “Efficient Parametrization of Generic Aircraft Geometry,” *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1269–1275,
doi:10.2514/3.46874.

[17] de Boor, C., *A Practical Guide to Splines*, Vol. 27 of *Applied Mathematical Sciences*, Springer, revised ed., 2001.

[18] Piegl, L. and Tiller, W., *The NURBS Book*, Monographs in Visual Communication, Springer, 2nd ed., 1997.

[19] Braibant, V. and Fleury, C., “Shape optimal design using B-splines,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3, 1984, pp. 247–267,
doi:10.1016/0045-7825(84)90132-4.

[20] Cosentino, G. B. and Holst, T. L., “Numerical optimization design of advanced transonic wing configurations,” *Journal of Aircraft*, Vol. 23, No. 3, 1986, pp. 192–199,
doi:10.2514/3.45288.

[21] Anderson, W. K. and Venkatakrishnan, V., “Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation,” in “35th Aerospace Sciences Meeting and Exhibit,” AIAA Paper 97-0643, Reno, Nevada, 1997,

doi:10.2514/6.1997-643.

[22] Gain, J. and Bechmann, D., “A Survey of Spatial Deformation from a User-Centered Perspective,” *ACM Transactions of Graphics*, Vol. 27, No. 4, 2008, pp. 107:1–107:21,
doi:10.1145/1409625.1409629.

[23] Sederberg, T. W. and Parry, S. R., “Free-form deformation of solid geometric models,” *ACM SIGGRAPH Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160,
doi:10.1145/15886.15903.

[24] Samareh, J. A., “Novel Multidisciplinary Shape Parameterization Approach,” *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 1015–1024,
doi:10.2514/2.2888.

[25] Ronzheimer, A., “Shape Parametrization Using Freeform Deformation,” in Kroll, N. and Fassbender, J. K., eds., “MEGAFLOW - Numerical Flow Simulation for Aircraft Design,” Springer Berlin Heidelberg, Vol. 89 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM)*, pp. 211–222, 2005,
doi:10.1007/3-540-32382-1_15.

[26] Duvigneau, R., “Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization,” Tech. Rep. RR-5949, Institut National de Recherche en Informatique et en Automatique (INRIA), 2006.

[27] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” in “13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference,” AIAA Paper 2010-9231, Fort Worth, Texas, 2010,
doi:10.2514/6.2010-9231.

[28] Gagnon, H. and Zingg, D. W., “Two-Level Free-Form Deformation for High-Fidelity Aerodynamic Shape Optimization,” in “14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference,” AIAA Paper 2012-5447, Indianapolis, Indiana, 2012,
doi:10.2514/6.2012-5447.

[29] 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.

[30] Dubé, J.-F., Guibault, F., Vallet, M.-G., and Trépanier, J.-Y., “Turbine Blade Reconstruction and Optimization Using Subdivision Surfaces,” in “44th AIAA Aerospace Sciences Meeting and Exhibit,” AIAA Paper 2006-1327, Reno, Nevada, 2006,
doi:10.2514/6.2006-1327.

[31] Hsu, W. M., Hughes, J. F., and Kaufman, H., “Direct manipulation of free-form deformations,” *ACM SIGGRAPH*

Computer Graphics, Vol. 26, No. 2, 1992, pp. 177–184, doi:10.1145/142920.134036.

[32] Yamazaki, W., Mouton, S., and Carrier, G., "Geometry Parameterization and Computational Mesh Deformation by Physics-Based Direct Manipulation Approaches," *AIAA Journal*, Vol. 48, No. 8, 2010, pp. 1817–1832, doi:10.2514/1.J050255.

[33] Jakobsson, S. and Amoignon, O., "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization," *Computers & Fluids*, Vol. 36, No. 6, 2007, pp. 1119–1136, doi:10.1016/j.compfluid.2006.11.002.

[34] Morris, 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, 2010, pp. 297–312, doi:10.1002/fld.2067.

[35] Lamousin, H. J. and Waggenspack Jr., W. N., "NURBS-based free-form deformations," *Computer Graphics and Applications, IEEE*, Vol. 14, No. 6, 1994, pp. 59–65, doi:10.1109/38.329096.

[36] Blake, 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. RP-1338, NASA, Ames Research Center, Moffett Field, California, 1994.

[37] Truong, 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, 2008, pp. 1695–1704, doi:10.2514/1.33836.

[38] Hicken, J. E. and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 400–413, doi:10.2514/1.44033.

[39] Lazarus, F., Coquillart, S., and Jancène, P., "Axial deformations: an intuitive deformation technique," *Computer-Aided Design*, Vol. 26, No. 8, 1994, pp. 607–613, doi:10.1016/0010-4485(94)90103-1.

[40] Gagnon, H. and Zingg, D. W., "High-fidelity Aerodynamic Shape Optimization of Unconventional Aircraft through Axial Deformation," in "52nd Aerospace Sciences Meeting," AIAA Paper 2014-0908, National Harbor, Maryland, 2014, doi:10.2514/6.2014-0908.

[41] Désidéri, J.-A., Majd, B. A., and Janka, A., "Nested and self-adaptive Bézier parameterizations for shape optimiza-

tion," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 117–131,
doi:10.1016/j.jcp.2006.12.016.

[42] Han, X. and Zingg, D. W., "An adaptive geometry parametrization for aerodynamic shape optimization," *Optimization and Engineering*, Vol. 15, No. 1, 2014, pp. 69–91,
doi:10.1007/s11081-013-9213-y.

[43] Hayes, J. G., "Curved knot lines and surfaces with ruled segments," in Watson, G. A., ed., "Numerical Analysis," Springer Berlin Heidelberg, Vol. 912 of *Lecture Notes in Mathematics*, pp. 140–156, 1982,
doi:10.1007/BFb0093154.

[44] Hoschek, J., "Intrinsic parametrization for approximation," *Computer Aided Geometric Design*, Vol. 5, No. 1, 1988, pp. 27–31,
doi:10.1016/0167-8396(88)90017-9.

[45] Gagnon, H. and Zingg, D. W., "Geometry Generation of Complex Unconventional Aircraft with Application to High-Fidelity Aerodynamic Shape Optimization," in "21st AIAA Computational Fluid Dynamics Conference," AIAA Paper 2013-2850, San Diego, California, 2013,
doi:10.2514/6.2013-2850.

[46] Goldman, R., "Matrices and Transformations," in Glassner, A., ed., "Graphics Gems I," Academic Press, pp. 472–475, 1990.

[47] Goldman, R., "More Matrices and Transformations: Shear and Pseudo-Perspective," in Arvo, J., ed., "Graphics Gems II," Academic Press, pp. 338–341, 1991.

[48] Gill, 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,
doi:10.1137/S0036144504446096.

[49] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, 2nd ed., 2006.

[50] Del Rey Fernández, D. C., Hicken, J. E., and Zingg, D. W., "Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations," *Computers & Fluids*, Vol. 95, No. 22, 2014, pp. 171–196,
doi:10.1016/j.compfluid.2014.02.016.

[51] Svärd, M. and Nordström, J., "Review of summation-by-parts schemes for initial-boundary-value problems," *Journal of Computational Physics*, Vol. 268, No. 1, 2014, pp. 17–38,
doi:10.1016/j.jcp.2014.02.031.

[52] Lomax, H., Pulliam, T. H., and Zingg, D. W., *Fundamentals of Computational Fluid Dynamics*, Scientific Compu-

tation, Springer, 2001.

[53] Hicken, 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, 2008, pp. 2273–2786, doi:10.2514/1.34810.

[54] Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Review*, Vol. 40, No. 1, 1998, pp. 110–112, doi:10.1137/S003614459631241X.

[55] Hicken, J. E. and Zingg, D. W., "Induced-Drag Minimization of Nonplanar Geometries Based on the Euler Equations," *AIAA Journal*, Vol. 48, No. 11, 2010, pp. 2564–2575, doi:10.2514/1.J050379.

[56] Kroo, I. M., "DRAG DUE TO LIFT: Concepts for Prediction and Reduction," *Annual Review of Fluid Mechanics*, Vol. 33, No. 1, 2001, pp. 587–617, doi:10.1146/annurev.fluid.33.1.587.

[57] La Roche, U. and La Roche, H. L., "Induced Drag Reduction Using Multiple Winglets, Looking Beyond the Prandtl-Munk Linear Model," in "2nd AIAA Flow Control Conference," AIAA Paper 2004-2120, Portland, Oregon, 2004, doi:10.2514/6.2004-2120.

[58] Spillman, J. J., "Use of Wing Tip Sails to Reduce Vortex Drag," *Aeronautical Journal*, Vol. 82, No. 813, 1978, pp. 387–395.

[59] Smith, M. J., Komerath, N., Ames, R., and Wong, O., "Performance analysis of a wing with multiple winglets," in "19th AIAA Applied Aerodynamics Conference," AIAA Paper 2001-2407, Anaheim, California, 2001, doi:10.2514/6.2001-2407.

[60] Miklosovic, D. S., "Analytic and Experimental Investigation of Dihedral Configurations of Three-Winglet Planforms," *Journal of Fluids Engineering*, Vol. 130, No. 7, 2008, pp. 071103:1–071103:10, doi:10.1115/1.2948372.

[61] Kafyeke, F., Pépin, F., and Kho, C., "Development of High-Lift Systems for the Bombardier CRJ-700," in "23rd Congress of the International Council of the Aeronautical Sciences," Toronto, Ontario, 2002.