



## Faculty Publications

---

2018-6

# Universal Airfoil Parametrization Using B-Splines

Dev Rajnarayan  
*Aerion Technologies*

Andrew Ning  
*Brigham Young University, aning@byu.edu*

Judd Mehr  
*Brigham Young University, juddmehr@gmail.com*

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the [Mechanical Engineering Commons](#)

### Original Publication Citation

Rajnarayan, D., Ning, A., and Mehr, J., "Universal Airfoil Parametrization Using B-Splines," AIAA Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, Jun. 2018. doi:10.2514/6.2018-3949

---

### BYU ScholarsArchive Citation

Rajnarayan, Dev; Ning, Andrew; and Mehr, Judd, "Universal Airfoil Parametrization Using B-Splines" (2018). *Faculty Publications*. 2118.  
<https://scholarsarchive.byu.edu/facpub/2118>

This Conference Paper is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

# Universal Airfoil Parametrization Using B-Splines

Dev Rajnarayan \*

*Aerion Technologies Corporation, Palo Alto, CA, 94303, USA*

Andrew Ning † and Judd Mehr ‡

*Brigham Young University, Provo, UT, 84602, USA*

**In this paper, we apply well-known techniques for parametric curves to the definition and deformation of airfoil sections. While it has already been shown that many versions of Kulfan’s Class Shape Transformation (CST) are exactly equivalent to Bézier curves, we show here that all NACA 4-digit thickness distributions and the PARSEC parametrization of Sobieczky are also higher-order Bézier curves. As with CST, Béziers and B-Splines provide direct control over aerodynamically meaningful features such as nose radius and boat-tail angle, but also provide more a intuitive parametrization of the rest of the airfoil surface. We show the efficacy of B-Spline-based parametrizations for approximating airfoil coordinates and for clean-sheet airfoil design. Finally, we show that this parametrization is ideally suited for progressive design.**

## I. Introduction

Airfoil parametrization is fundamental to aerodynamic shape optimization, and a range of approaches are used by the aerospace community. Some common approaches include the various NACA series, the Class-Shape-Transformation (CST) method [1], Hicks-Henne bump functions [2], radial-basis-functions [3], Bézier surfaces, B-Splines, a singular value decomposition approach [4], the PARSEC method [5], and freeform deformation [6].

Several studies have compared airfoil parametrization methods. Masters et al. explored the number of design variables required to match airfoil geometries [4] finding that for the six methods explored about 25 design variables were necessary for accurate shape recovery. Castonguay and Nadarajah performed an inverse design problem to match a target pressure distribution using a Reynolds Averaged Navier Stokes solver and gradients computed using an adjoint method [7]. They found that the PARSEC method did not have sufficient flexibility to match the target distribution, and that the Hicks-Henne bump functions could match the distribution but with a lower accuracy than B-splines and mesh points. Mesh points were tractable in this case because gradients were computed with an adjoint method, but for forward methods (like finite differencing) would not be feasible because of the excessively large number of design variables. Mousavi and the previous two authors also explored using mesh points, B-Splines, and the CST method for three-dimensional inverse and direct design problems using the Euler equations [8]. They found that mesh points provided the highest accuracy, B-splines achieved a similar accuracy, and that the CST method yielded a lower level of accuracy. Higher order Bernstein polynomials were explored, but they did not improve accuracy as they led to high-frequency oscillation. Wang et al. used a stochastic gradient estimation technique with the Euler equations and found that the NACA parametrization and the Hicks-Henne bump functions found lower drag solutions than B-splines and Wagner shape functions [9].

Taken together these studies indicate a tradeoff in flexibility versus robustness. At one of the spectrum, mesh points provide ultimate flexibility, but are susceptible to producing poor airfoils during an optimization that can cause numerical difficulties in the analysis or in the optimization. If an analysis is robust, smooth, and can provide exact gradients, then more flexible methods are desirable. On the other end, methods like the NACA 4-series provide much less flexibility, but if used with reasonable bounds consistently produce shapes that work well with less robust analysis method or with less accurate gradients. There are also important tradeoffs in using absolute versus perturbed geometries, and global versus local control [10].

Since so much has been written about airfoil parametrizations, one must skeptically examine the need for *yet another* method. This paper is *not* about proposing a new airfoil parametrization. We do not introduce any additional terms, factors, functions, or formulae to modify existing formulations. Rather, we show that many supposedly different airfoil

---

\*AIAA Member.

†Assistant Professor, Mechanical Engineering Department, AIAA Senior Member.

‡PhD Candidate, Mechanical Engineering Department, AIAA Member.

parametrizations are in fact special cases of B-Splines and Béziers, and we show a methodology to effectively use B-Splines for airfoil shape optimization.

There is no single ideal parametrization for airfoils, and the choice of a good parametrization is highly application dependent. We submit, however, that what we are proposing is a universal approach using stable, well-understood, and widely-used techniques, and that this approach admits a wide variety of parametrizations under a single umbrella. We demonstrate that the parametrization for fitting and the parametrization for shape design are independent decisions. Using B-splines one can fit any desired airfoil shape to whatever degree of accuracy desired, and B-splines can then be used to define perturbations to this existing shape. B-splines allow for highly localized shape changes in contrast to many existing methods where changing one design variable changes the shape everywhere along the chord. Finally, B-splines allow for both degree elevation as well as knot insertion. Use of the latter technique is ideally suited for a progressive design approach as discussed in this paper.

The rest of the paper is organized as follows. First, we show that the equivalence of several well known parametrization methods with B-Splines. Then, we discuss some of the advantages of using B-Splines directly for airfoil parametrization. Next, we show the efficacy of representing given airfoil coordinates using B-Splines as compared to CST. Finally, we present a few optimization cases looking at both clean-sheet airfoil design, as well as an approach for progressive design by perturbations. Although all of this work is just as applicable to airfoils with sharp leading edges, the focus of this paper will be on round-nosed airfoils, which are more challenging.

## II. B-Spline Equivalence

One of the underlying themes for this study is the realization that some of the different airfoil parametrization methods are actually subsets of B-splines. In this section, we show that three well-known airfoil parametrizations: NACA 4-series, CST, and the PARSEC method are *exactly* equivalent to Bézier curves. In other words, using Bézier curves directly allows one to generate every airfoil shape encompassed by these parametrizations, though the opposite is not true (i.e., NACA/CST/PARSEC cannot match every shape that a Bézier curve can). Because a B-Spline is a generalization of a Bézier curve, then NACA 4-series, CST, and PARSEC can all be represented exactly as B-splines.

### 1. NACA 4-Series

Every NACA 4-series thickness distribution is a Bézier curve, and every NACA 4-series airfoil is a B-Spline. The thickness distribution for a symmetric NACA 4-digit airfoil with maximum thickness-to-chord ratio  $\tau$  is defined as follows for unit chord.

$$T(x) = 5\tau \left( 0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4 \right), \quad x \in [0, 1]. \quad (1)$$

We substitute  $u = \sqrt{x}$ , and project the resulting polynomials shown below onto the Bernstein basis  $\mathcal{B}_8$  of order eight.

$$y(u) = 5\tau \left( 0.2969u - 0.1260u^2 - 0.3516u^4 + 0.2843u^6 - 0.1015u^8 \right), \quad (2)$$

$$x(u) = u^2. \quad (3)$$

The Bernstein coefficients give us the  $y$  and  $x$  coordinates, respectively, of the control points of an eighth-order Bézier curve. Let these control points be  $(X_0, Y_0), (X_1, Y_1), \dots, (X_8, Y_8)$ .

As an example, let us choose  $\tau = 0.12$  for the ubiquitous NACA 0012. The associated Bézier curve, its control points, and evaluations of Eq. 1, are shown in Fig. 1. Note that there is no approximation of the NACA formula here, the Bézier and the NACA formula are mathematically identical.

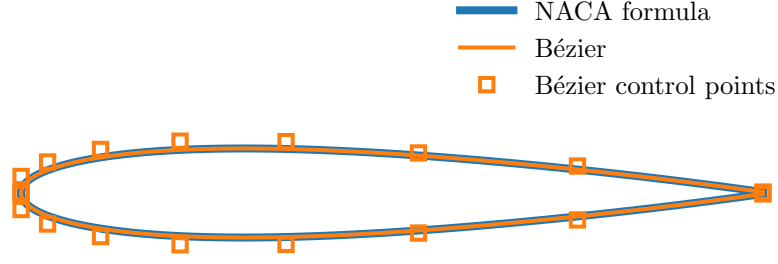
The camberline for a NACA 4-series airfoil is defined in a piecewise manner, leading to a  $C^1$  continuous curve.

$$\bar{y}(x) = \begin{cases} \frac{\epsilon}{p^2} (2px - x^2) & 0 \leq x \leq p \\ \frac{\epsilon}{(1-p)^2} (1 - 2p + 2px - x^2) & p \leq x \leq 1 \end{cases} \quad (4)$$

The discontinuity in the second derivative prevents representation using a Bézier curve, but can still be exactly represented with a B-Spline\*.

---

\*This assumes that the NACA 4-series thickness is defined normal to the chord line, rather than normal to the camber line as prescribed in the original definition. For shape optimization, the normal to the camber line approach can lead to poorly defined airfoils. For example a high amount of camber near the leading edge changes the chord both in length and angle. This makes the formula no longer explicit as the thickness and camber are dependent on the chord location. For shape optimization the distinction is of little importance as the shape will be optimized anyway, and so it is generally advantageous to use thickness normal to the chord.



**Fig. 1** NACA 0012 represented as a Bézier curve.

If we keep the  $x$ -coordinates of the control points fixed, then  $Y_1$  controls the leading-edge radius, much like the first parameter in the CST method, and  $Y_7$  controls the boat-tail angle, much like the last parameter in CST. The role of  $Y_2, \dots, Y_6$ , though, is much clearer here because they affect the geometric curve directly rather than an intermediate shape function.

## 2. CST Round-Nosed Airfoil

The Class-Shape-Transformation (CST) method [1], introduced by Kulfan over a decade ago, has found popularity in the aerodynamic design community for its optimization-friendly parameters that yield shapes that maintain desirable geometric characteristics over a wide range of parameter variations. The treatment of round-nosed airfoils is particularly appealing because Kulfan's formulation retains direct control over leading-edge radius and avoids the generation of irregular airfoil-nose shapes.

In this section we show that every CST round-nosed airfoil is a Bézier curve. This equivalence has already been pointed out by Marshall [11], although we will take a closer look at the relationship between CST and B-Splines. Consider a round-nosed airfoil parametrized using the CST method. The approach multiplies a class function that yields a round nose and sharp trailing edge:

$$C(x) = \sqrt{x}(1-x), \quad (5)$$

and *modulates* a shape function, which is often parametrized as a polynomial. Though this polynomial is usually represented in the Bernstein basis, we use the power basis here for simplicity of the following derivation.

$$S(x) = \sum_{i=0}^n a_i x^i. \quad (6)$$

An additional term,  $x\Delta z_{TE}$ , is added to allow for a thick trailing edge. Putting these terms together describes one side (upper or lower) of a round-nosed airfoil:

$$y = \sqrt{x}(1-x) \sum_{i=0}^n a_i x^i + x\Delta y_{TE} \quad (7)$$

In the above representation  $x$ ,  $y$  and  $\Delta y_{TE}$  are all dimensionless and normalized, i.e., they are interpreted as fractions of the airfoil chord. The leading-edge radius  $r_{LE}$  and boat-tail angle  $\beta_{TE}$  can be related to the first and last coefficients by:

$$\begin{aligned} S(0) &= \sqrt{2r_{LE}}, \\ S(1) &= \tan \beta_{TE} + \Delta y_{TE}. \end{aligned} \quad (8)$$

To show that this formulation is equivalent to a single higher-order Bézier curve we make the variable substitution:

$$x = u^2 \quad (9)$$

Substituting this into Eq. 7 yields:

$$\begin{aligned}
y &= u(1 - u^2) \sum_{i=0}^n a_i u^{2i} + \Delta y_{TE} u^2, \\
&= \sum_{i=0}^n a_i (u^{2i+1} - u^{2i+3}) + \Delta y_{TE} u^2, \\
&= \sum_{i=0}^{n+1} (a_i - a_{i-1}) u^{2i+1} + \Delta y_{TE} u^2, \quad \text{where } a_{-1} = 0, a_{n+1} = 0, \\
&= \sum_{i=1}^{2n+3} C_i u^i + \Delta y_{TE} u^2.
\end{aligned} \tag{10}$$

where

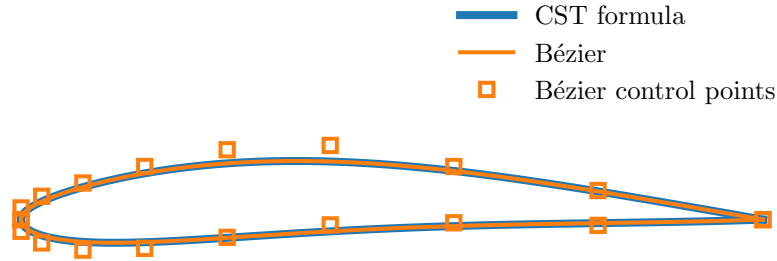
$$C_i = \begin{cases} a_{(i-1)/2} - a_{(i-3)/2}, & \text{if } i \text{ is odd,} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Eq. 10 describes a polynomial of order  $2n + 3$  represented using the monomial basis. The Bernstein basis of order  $2n + 3$  spans all polynomials of order  $2n + 3$ , which implies that both Eq. 9 and Eq. 10 can be represented in that Bernstein basis *exactly*. This yields the parametric curve C:

$$\mathbf{C}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{2n+3} X_i B_i(t) \\ \sum_{i=0}^{2n+3} Y_i B_i(t) \end{pmatrix}, \tag{12}$$

where  $B_i$  is the  $i^{\text{th}}$  Bernstein basis polynomial. Eq. 12 is the general representation of a Bézier curve of order  $2n + 3$ , showing that the Kulfan parametrization for round-nosed airfoils is mathematically equivalent to a single higher-order Bézier curve. In the monomial basis representation, the coefficients of all even powers of  $t$  are zero, but when transformed to the Bernstein basis, all coefficients are generally nonzero, except  $X_0$  and  $Y_0$  in all cases, and  $Y_{2n+3}$  if the trailing-edge is sharp.

As an example, we use  $r_{LE} = 0.01$ ,  $\beta_{TE_u} = 14^\circ$ ,  $\beta_{TE_l} = -2^\circ$ , and intermediate Bernstein coefficients  $a_u = [0.25, 0.20]$ ,  $a_l = [0.02, 0.01]$ . The equivalent Bézier curve, its control points, and direct evaluations of the CST formula are shown in Fig. 2. As shown, if the shape function is a polynomial of degree  $n$ , then the degree of the resulting Bézier is  $2n + 3$ . Since the shape function in this case is a cubic polynomial, the airfoil Bézier has degree 9, and the figure shows ten control points each for upper and lower surfaces.



**Fig. 2 CST airfoil represented as a Bézier curve.**

As with the NACA airfoil, there is no approximation involved here. The CST airfoil is mathematically identical to the Bézier representation. We have the same sort of direct control of nose radius and trailing-edge angles, and at the same time, the intermediate parameters are geometrically and visually meaningful and intuitive.

This example and the preceding one also show that working with a broader class of parametrizations has advantages. Every CST round-nosed airfoil has a Bézier representation, but not every Bézier can be represented as a CST airfoil.

For instance, no Bézier of even orders can be represented exactly in CST, even though some of those Bézier (viz., *all* NACA 4-digit thickness distributions) make pretty good airfoils!

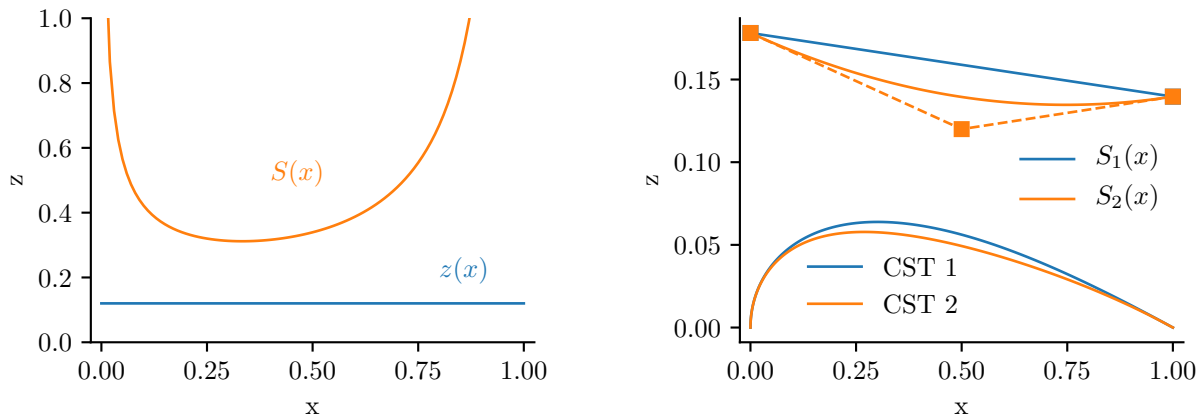
Another advantage of working with B-Splines directly is that the parameter changes are more intuitive. For example, the CST shape function  $S(x)$  is related to the geometric curve  $y(x)$  (the airfoil surface) via the following nonlinear transformation:

$$S(x) = \frac{y(x)}{\sqrt{x(1-x)}}. \quad (13)$$

Physically, it represents a (square-root of the) radius of curvature at the leading edge, a slope (boat-tail angle) at the trailing edge, and positions in between. This unusual mapping makes it challenging to intuitively understand the effect of the shape function at various chord fractions. For instance, a “horizontal”(constant thickness) straight-line in geometric space has a shape function that goes to infinity at the leading edge *and* the trailing edge.

The shape function in CST is often represented as a Bernstein polynomial of order  $n$ , which as shown can trivially be represented as a 2-D Bézier curve with  $n + 1$  control points whose ordinates (vertical coordinates) are the Bernstein coefficients of the shape function. The control-point abscissae (horizontal coordinates) are  $0, 1/n, 2/n, \dots, 1$ . The CST Bernstein coefficients (Bézier control points) affect the shape function in an intuitive manner, but the shape function is related to the geometric curve in a counter-intuitive manner, posing challenges for designers to set, for instance, design-variable bounds on the Bernstein coefficients in CST.

An example of this issue is shown in Fig. 3. The panel on the left shows a “constant  $t/c$ ” straight line and its associated shape function, which rises unbounded at both the leading edge and trailing edge. The panel on the right shows two CST airfoils. The first CST airfoil (in blue) specifies only leading-edge radius and boat-tail angle, so the shape function is a straight line. The second CST airfoil (in orange), includes one intermediate Bernstein coefficient. The value of that intermediate Bernstein coefficient is 0.12, which happens to be a substantially smaller number than the value of the original shape function at  $x = 0.5$ . As a consequence, the modified airfoil has smaller thickness over the entire chord, but preserves the original airfoil’s leading-edge radius and boat-tail angle. Note that the value of the original shape function at  $x = 0.5$  is exactly halfway between the leading-edge radius and boat-tail angle; even though its effect is on position, the value of the shape function is related to a curvature and a slope. As the picture shows, this intermediate Bernstein coefficient and its trivially-related Bézier control point affect the shape function in a very intuitive way, but affect the airfoil shape in a counter-intuitive way.



(a) The shape function,  $S(x)$  for the simple geometry  $y(x) = 0.12$ .

(b) The shape functions for two different CST airfoils.

**Fig. 3** Shape functions for a “constant  $t/c$ ” line and two airfoils.

### 3. PARSEC

Another popular airfoil parametrization method is the PARSEC method proposed by Sobieczky [5]. This method is inspired by the NASA 4-digit airfoil parametrization, but instead uses a 6th-order polynomial (although to be precise it

is not actually a polynomial in  $x$  as the powers are non-integer):

$$y = \sum_{i=1}^6 a_i x^{i-1/2} \quad (14)$$

If we make the familiar substitution of ( $x = u^2$ ) then the resulting curve is equivalent to an 11th order polynomial in  $u$ :

$$y = \sum_{i=1}^{11} c_i u^i \quad (15)$$

where

$$c_i = \begin{cases} a_{(i+1)/2} & \text{if } i \text{ is odd} \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Thus, the PARSEC parametrization is *exactly* equivalent to a Bézier curve of order 11.

$$C(u) = \begin{pmatrix} x(u) \\ y(u) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{11} X_i B_{11,i}(u) \\ \sum_{i=0}^{11} Y_i B_{11,i}(u) \end{pmatrix} \quad (17)$$

As an example, we use the PARSEC parameters  $r_{LE} = 0.015$ ,  $x_u = 0.3$ ,  $y_u = 0.06$ ,  $y''_u = 0.45$ ,  $x_l = 0.3$ ,  $y_l = 0.06$ ,  $y''_l = 0.45$ ,  $y_{TE} = 0$ ,  $\delta y_{TE} = 0$ ,  $\alpha_{TE} = 0.02^\circ$ ,  $\beta_{TE} = 19.5^\circ$ . The equivalent Bézier curve, its control points, and direct evaluations of the PARSEC formula are shown in Fig. 4. Again, the equivalence is exact.

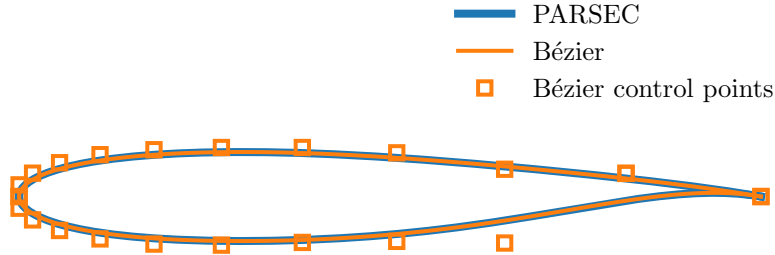


Fig. 4 PARSEC airfoil represented as a Bézier curve.

### III. B-Splines for Airfoil parametrization

In the preceding section, we showed that three existing airfoil parametrizations are mathematically identical to Béziars. B-Splines are a generalization of Béziars, and can be thought of as a chain of Béziars connected in a curvature-continuous manner. Every Bézier is a B-Spline, but not every B-Spline is a Bézier. We do not delve into any further detail on the topic of B-Splines: they are widely used, and the details can be found in any good textbook [12] on the topic. In this section we discuss some benefits of B-Splines and a first-principles parametrization that includes relevant parameters to an aerodynamicist.

#### A. B-Spline Benefits

In this paper, we attempt to show that using the B-Spline or Bézier formulation yields more than just mathematical elegance (or CAD compatibility). It unlocks a wealth of widely-known and well-tested techniques from the field of parametric curves and 3-D modeling. Some of these features were alluded to in the previous section.

- 1) All three of the discussed airfoil parametrization methods use design variables that change the airfoil shape globally across the chord. The realization that B-splines can be used instead of Béziars immediately admits

decoupled local shape variations as opposed to global shape variations: each parameter affects the airfoil shape only over a small region. For example, this allows design changes to be focused on a particular region of the airfoil where finer shape control is desired while keeping the rest of the airfoil fixed.

- 2) Because B-Splines can be used as deformations about a nominal geometry, the parametrization of the airfoil fit and the parametrization of the design variables can be independent. In other words, fitting airfoil coordinates is no longer a challenge: the interpolation error can be exactly zero if desired. If the level of error in the given airfoil coordinates is known, users can specify it, or if unknown, they can reasonably trade off interpolation error with, say, integrated curvature.
- 3) In comparison to the previous methods, B-Splines have more flexibility and more control over numerical behavior. As shown previously, as the number of Bernstein coefficients is increased the CST method performs *degree elevation* with every other degree (NACA4 and PASRSEC have no means to increase design variables). In contrast, B-Splines can perform *degree elevation* and/or *knot insertion*. For example, 5-7 Bernstein coefficients is fairly typical for aerodynamic design and in CST this leads to high-order polynomial curves that often exhibit some degree of oscillation and undulation. B-Splines, in contrast, are often cubic polynomials, regardless of the number of control points they have, and are numerically stable.
- 4) The “intermediate” Bernstein coefficients of CST yield nonintuitive changes in the airfoil geometry through the shape function. In contrast, B-Splines give way to extremely intuitive control-point positions that can be represented pictorially alongside the airfoil shape. The effect of moving these control points is also clear and direct. This feature also makes it easier to provide appropriate design variable bounds for optimization.
- 5) The ability to perform knot insertion allows for a progressive design approach using an incremental parametrization. For example, one can start with a nominal airfoil (possibly from an airfoil database), make gross changes to the shape using a few variables, and add design variables where appropriate *without changing the perturbed shape while adding variables*.
- 6) Though not explored in the scope of this paper, a tensor product of B-Splines allows for a straightforward extension to three-dimensional geometries.

## B. Airfoil Shapes from First Principles

In this section, we consider the derivation of B-Spline-based airfoil shapes from first principles. Given a nose radius  $r_{LE}$ , trailing-edge (half) thickness  $\Delta y_{TE}$ , and boat-tail angle  $\beta_{TE}$ , the simplest parametric curve we can construct that satisfies these five boundary conditions (position, tangent, and curvature at the leading edge, and position and tangent at the trailing edge) is a conic section. Sederberg points out in compiled notes [13] that the end-point curvature at the start of a rational Bézier curve of degree  $d$  is given by

$$\kappa_0 = \frac{w_0 w_2}{w_1^2} \frac{d-1}{d} \frac{h}{a^2}, \quad (18)$$

where  $w_0, w_1, w_2$  are the weights of the first three control points,  $a$  is the distance between the first and second control points, and  $h$  is the perpendicular distance between the third control point and the straight line through the first two. The terminal tangents of any clamped Bézier are given by the terminal segments of the control polygon. This suggests that, as long as  $\beta_{TE} < 90^\circ$ , the three control points of a NURBS curve of degree 2 satisfy terminal position and tangency constraints, as shown in Fig.5. All that remains to be done is to compute the weight  $w_2$  of the second control point  $P_1$ , and this is easily accomplished using Eq. 18, since we know that  $h = 1$ ,  $a = \Delta y_{TE} + \tan \beta_{TE}$ , and  $w_0 = w_2 = 1$ .

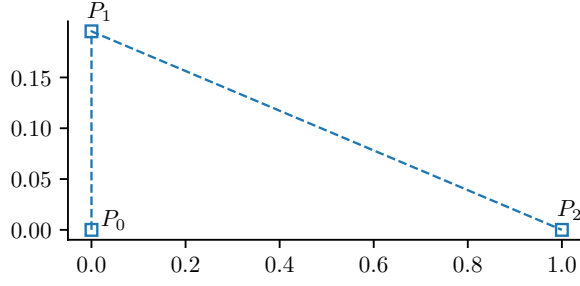
$$w_1 = \frac{1}{\Delta y_{TE} + \tan \beta_{TE}} \sqrt{\frac{r_{LE}}{2}}. \quad (19)$$

It is interesting to note that the boundary conditions for the CST shape function involve the same terms rearranged, as shown in Eq. 8. The second-order rational Bézier with control points  $(P_0, P_1, P_2)$ , knot vector  $(0, 0, 0, 1, 1, 1)$ , and weights  $(1, w_1, 1)$  with  $w_1$  given by Eq. 19 is shown in Fig. 6a, and is compared with the ‘first-principles’ CST airfoil given the same information, which is a non-rational<sup>†</sup> quintic (order 5) Bézier.

We can do something similar for a classic cubic Bézier. In this case, we can satisfy all the terminal constraints with a non-rational Bézier, but now we have a non-unique solution in the  $x$ -location of the second control point. If, however, we prescribe the parametrization  $x = u^2$ , one that we have seen several times before, then there is no ambiguity. We

<sup>†</sup>All control points have unit weight.





**Fig. 5 Control-point positions for Bézier airfoil given nose radius, boat-tail angle, and trailing-edge thickness.**

represent the polynomial  $u^2$  using the Bernstein basis for cubic polynomials, and this yields control-point  $x$ -locations of  $(0, 0, 1/3, 1)$ . We use Eq. 18 with  $h = 1/3$  and all weights set to unity, to compute  $a$ , the  $y$ -location of the second control point. This gives

$$a = \frac{1}{3}\sqrt{2r_{LE}}. \quad (20)$$

The four control points of the cubic Bézier that satisfy the terminal constraints on airfoil shape are given by

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 0, & 0 \\ 0, & \frac{1}{3}\sqrt{2r_{LE}} \\ \frac{1}{3}, & \Delta y_{TE} + \frac{2}{3}\tan\beta_{TE} \\ 1, & \Delta y_{TE} \end{bmatrix}. \quad (21)$$

In Figs. 6a and 6b, we compare the ‘fundamental’ shapes generated from the simplest boundary conditions of nose radius, boat-tail angle, and trailing-edge thickness, using a conic, a cubic Bézier, and the CST formulation, which is a fifth-order Bézier.

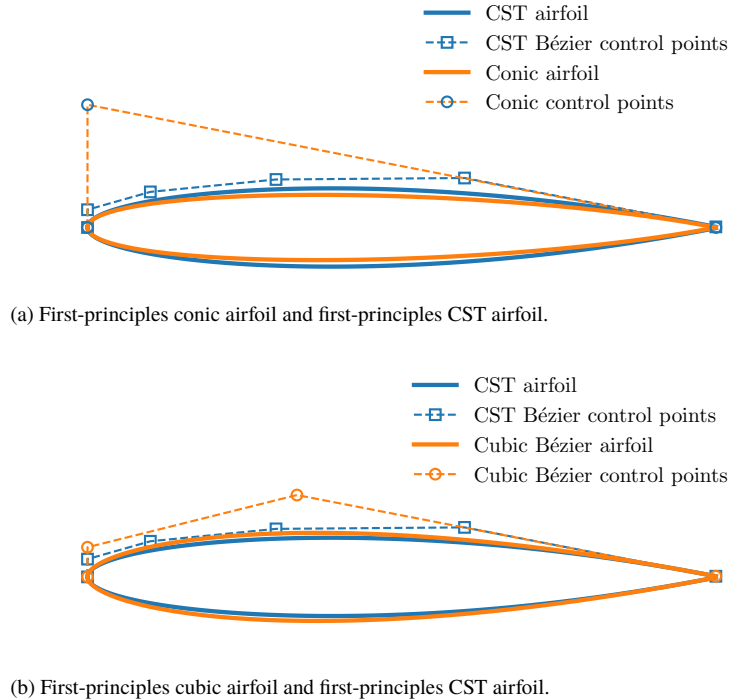
## IV. Fitting Airfoils

Parametric curves can be used both for interpolation and approximation. Algorithms for constructing smoothing (approximating) splines using any given tolerance [14] have been known and used for decades. In this section, we examine the efficacy of fitting airfoil coordinates using B-Splines. We do not prescribe a fit tolerance, but use another standard algorithm that takes a prescribed knot vector.

We downloaded coordinates for approximately 1,500 airfoils from the UIUC airfoil database (in particular, [this](#) archive), and examined the efficacy of least-squares fitting using CST and B-Splines. As it turns out, the fitting procedure for both formulations reduces to linear least-squares, so there should be no convergence issues. Moreover, the B-Spline fitting algorithm is provided as part of Python’s `scipy.interpolate` package, further simplifying matters. Finally, since the canonical CST shape function is just a Bézier, the same `scipy` spline-fitting algorithm can be used for approximating it.

### A. CST Fitting

For a given set of airfoil coordinates, we compute shape function values for upper and lower surfaces separately. The number of Bernstein coefficients defines the degree  $d$  of the shape function Bézier, and this fixes the knot vector:  $d + 1$  repeated knots at 0 and  $d + 1$  repeated knots at 1. Once the best-fit shape-function Bézier is computed, we multiply by the class function, add trailing-edge thickness, and convert the entire formula for each surface to a single Bézier (which is simple because the formula is just a polynomial in  $u = \sqrt{x}$ ). This procedure gives us the control points of the shape function as well as the control points of the airfoil shape. At the leading edge, we permit a curvature discontinuity but no tangent discontinuity. In other words, the first Bernstein coefficient is not constrained to be the same for the upper and lower surfaces. We will revisit this matter in a subsequent section.



**Fig. 6 A comparison between some simple first-principles airfoil shapes.**

## B. B-Spline Fitting

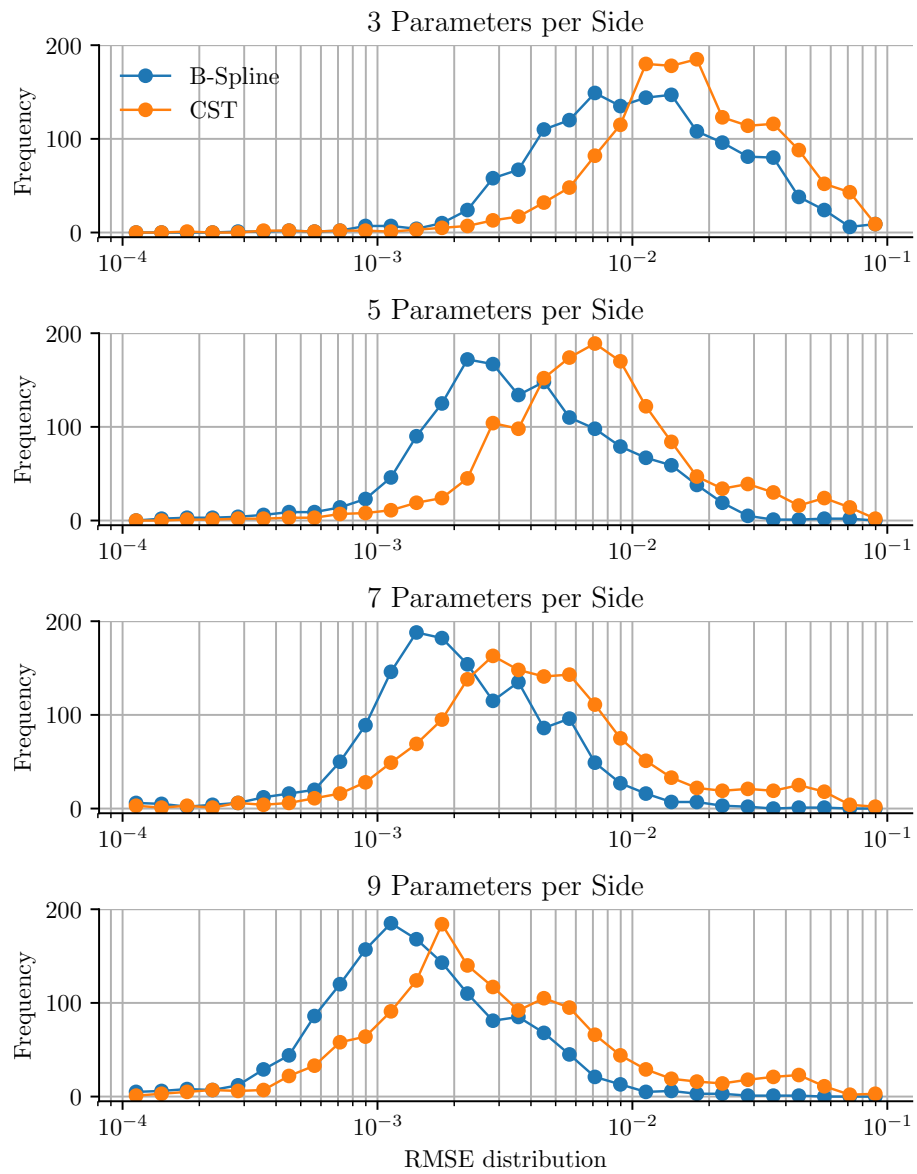
This process is only slightly more involved. For each airfoil surface, we use a similar approach as we did above for the shape function, except for three differences:

- 1) We directly fit the airfoil  $y$  coordinates, not the shape function.
- 2) The degree of the parametric curve is fixed at 3, and the knot vector is uniform. In other words, if there are to be three “design variables” per surface, then each surface must have five control points (the first and last are fixed at the LE and TE). The second and penultimate control points are already present in the cubic Bézier with no internal knots, so that leaves only one more control point. A review of the knot-insertion process confirms that this requires a single knot to be inserted at  $u = 0.5$ . Similarly, if we need  $k$  design variables on an airfoil surface represented by a clamped B-Spline, then we must insert  $k - 2$  knots uniformly in  $(0, 1)$ .
- 3) The  $x$ -locations of the control points are dictated by the parametrization  $x = u^2$ . We represent the polynomial  $u^2$  as a cubic Bézier, and then replicate the knot-insertion process we just described. The resulting *scalar* control points give us the  $x$ -locations of the control points of the 2-D curve. This also fixes the association between parameter values  $u$  and the given airfoil coordinate  $y$ , and this association is required by the B-Spline fitting algorithm.

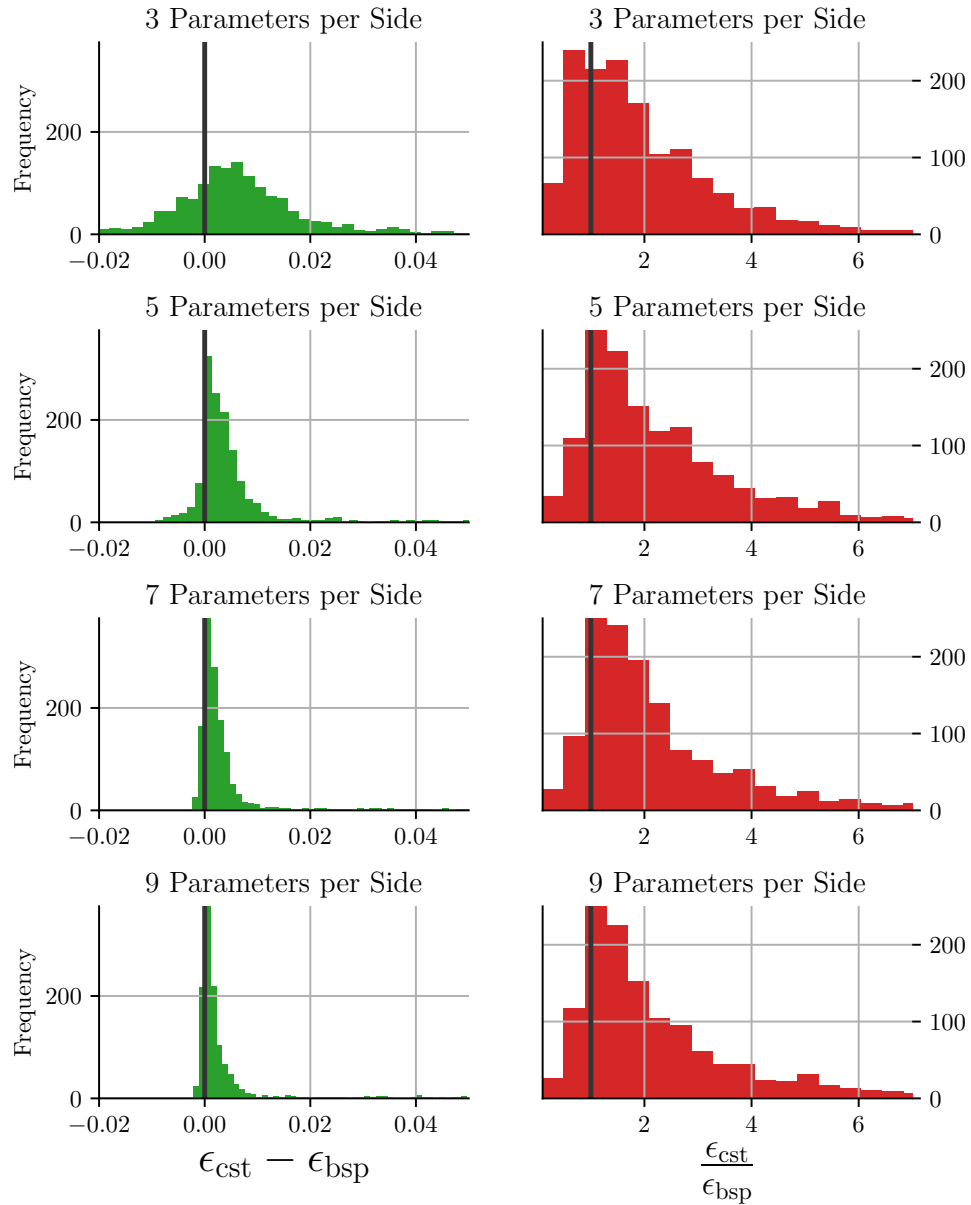
We fit each airfoil using both approaches, and we repeat the process using 3, 5, 7, and 9 parameters per side. In the case of CST, the parameters are used to describe the shape function, and for the B-Splines, they are control point locations that describe the airfoil shape directly. The quality of the fits are quantified by an error metric, the root-mean-square error (RMSE) between the fit and the airfoil coordinates, where the mean (average) is over all the given coordinates for an airfoil. The results are summarized in Tables 1 and 2, and Figs. 7 and 8.

Table 1 shows aggregate error metrics for each fitting approach. In conjunction with the plots in Figs. 7 and 8, it shows that the overall error metrics for the B-Spline fits are better. Table 2 aggregates error metric comparisons on a per-airfoil basis. For every airfoil, we track the difference in RMSE and the ratio of RMSE between the two approaches, and aggregate those comparisons across all airfoils. We see that for a given number of parameters, regardless of that number, the median<sup>‡</sup> RMSE using B-Splines is smaller than that achieved by CST by almost a factor of two. The worst-case RMSE of CST is 50-100 times as large as that using B-Splines. In the cases where the CST RMSE was lower than that achieved by the B-Spline, the B-Spline RMSE is already low. In other words, we found no cases where the

<sup>‡</sup>The mean in RMSE is over all points in a given airfoil. The median is over all airfoils; this is to reduce the impact of outliers.



**Fig. 7** Aggregate error metrics from fitting a database of 1,500 airfoils using CST and cubic B-Splines. These plots show histograms on a semi-log ( $x$ ) scale.



**Fig. 8** Comparative error metrics from fitting a database of 1,500 airfoils using CST and cubic B-Splines. The left-hand column of plots shows histograms of excess error that the CST fit has over cubic B-Splines. The right-hand column of plots presents the same information as error ratios rather than a difference of errors. The thick black vertical lines represent zero RMSE difference (unit RMSE ratio).

Number of parameters	CST RMSE			B-Spline RMSE		
	Min.	Max.	Median	Min.	Max.	Median
3	$5.66 \times 10^{-5}$	$9.69 \times 10^{-1}$	$1.63 \times 10^{-2}$	$1.96 \times 10^{-5}$	$3.77 \times 10^{-1}$	$1.03 \times 10^{-2}$
5	$6.89 \times 10^{-5}$	$6.05 \times 10^{-1}$	$6.82 \times 10^{-3}$	$1.94 \times 10^{-5}$	$2.67 \times 10^{-1}$	$3.47 \times 10^{-3}$
7	$3.89 \times 10^{-5}$	$4.10 \times 10^{-1}$	$3.83 \times 10^{-3}$	$1.77 \times 10^{-5}$	$1.84 \times 10^{-1}$	$1.96 \times 10^{-3}$
9	$3.69 \times 10^{-5}$	$3.71 \times 10^{-1}$	$2.37 \times 10^{-3}$	$8.67 \times 10^{-6}$	$1.23 \times 10^{-1}$	$1.33 \times 10^{-3}$

**Table 1 Overall error statistics for CST and B-Splines.**

Number of parameters	RMSE difference $\epsilon_{\text{cst}} - \epsilon_{\text{bsp}}$			RMSE ratio $\frac{\epsilon_{\text{cst}}}{\epsilon_{\text{bsp}}}$		
	Min.	Max.	Median	Min.	Max.	Median
3	-0.086362	0.778568	0.005803	0.052	46.867	1.611
5	-0.052093	0.440616	0.002694	0.134	96.864	1.835
7	-0.118071	0.389212	0.001559	0.145	94.061	1.826
9	-0.041182	0.351600	0.000861	0.197	94.735	1.732

**Table 2 Comparative error statistics for CST and B-Splines.**

B-Spline fit is terrible and CST is very good. In a few cases CST is at best slightly better than the B-Spline, and only in cases where the spline is pretty good to begin with.

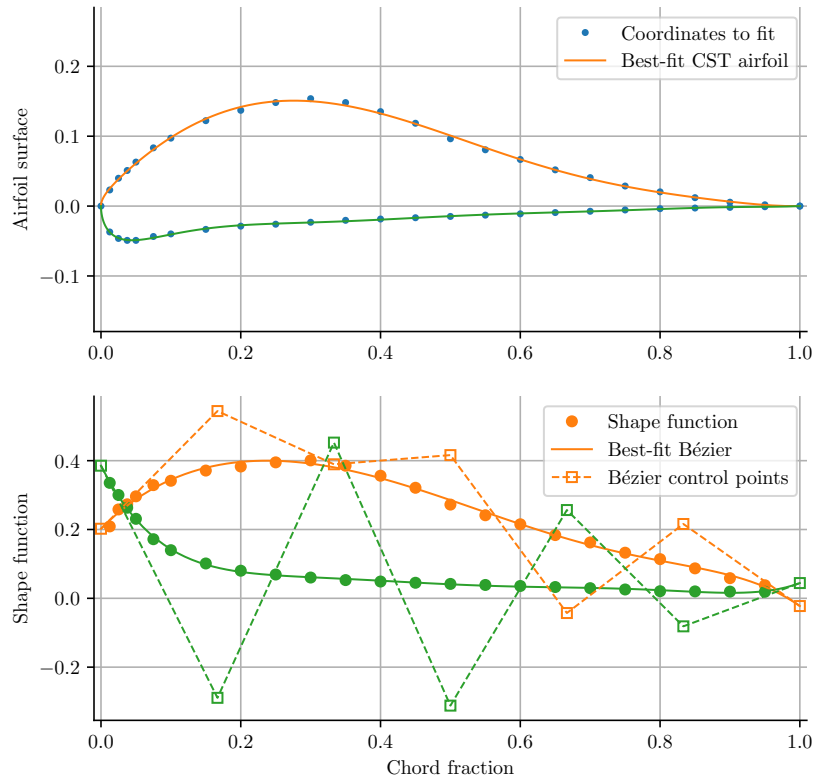
### C. Fitting the Shape Function vs. Fitting the Airfoil Surface

We take a closer look at the nonlinear transformation defined by the shape function, and the process of fitting it using a high-order Bernstein polynomial. We use L1003, one of the Liebeck high-lift airfoil sections for this example. The airfoil and its shape function are shown in Fig. 9. We compute a least-squares fit to the shape function values using a sixth-order Bernstein polynomial. The lower sub-plot shows that, although the shape function is a well-behaved and smooth curve, the control points (square symbols) of the associated Bézier offer little insight into the shape function. Naturally, it is even harder to see any relationship between these control points and the airfoil shape being approximated. This also brings out another feature of higher-order Bézier curves: it is difficult to make localized changes. In order to make changes that are even slightly restricted in spatial extent, there need to be dramatic movements of control points, exactly as shown in the plot. This means that, as far as shape flexibility is concerned, using degree elevation alone will soon lead to rapidly diminishing returns. Increasing the polynomial degree beyond 7 or 9 is unlikely to be very useful for shape control.

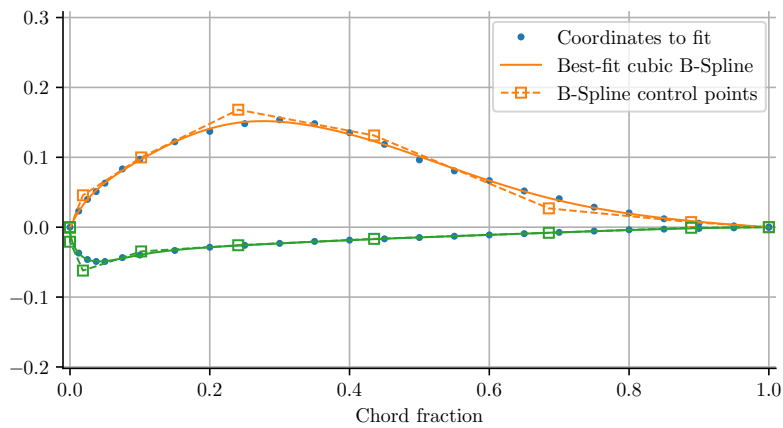
In contrast, Fig. 10 shows a best-fit cubic B-Spline using five internal knots (seven design variables, nine control points) per side. The control points reflect the shape of the curve. As the number of knots increases, the control points move closer to the curve and their effect becomes more local. In stark contrast with the shape-function control points, these B-Spline control points have an intuitive representation and favorable numerical behavior.

## V. Optimization

One of the primary motivations for airfoil parametrization is design optimization. While a given parametrization may provide good fits to existing airfoil shapes, that does not necessarily imply that the parametrization is effective for optimization. We emphasize again that there is no one best approach to airfoil parametrization. Different applications call for different needs. For example, in the presence of analytic gradients for aerodynamic objectives, such as may be obtained via the use of adjoint solvers, one could use ‘all’ surface mesh points as design variables. In contrast, with a noisy solver and less accurate gradients, one is forced to use a restrictive parametrization that uses a small number of design variables and has a strong affinity for a small family of smooth shapes. The promise of scalable parametrizations such as the ones in this paper is that they can bridge the gap between these two extremes. Insight into how best to bridge this gap, and how to detect circumstances that would favor one approach over the other, would be valuable. This section provides an initial exploration into this topic.



**Fig. 9** L1003 airfoil, the associated CST shape function, the best-fit Bernstein polynomial (Bézier curve), and the control points of that Bézier.



**Fig. 10** L1003 airfoil, the best-fit cubic B-Spline with 7 design parameters per surface, and associated control points.

## A. Formulation

For this paper, we use a lower-fidelity airfoil analysis and inexact gradients. The problem at hand is subsonic airfoil design using XFOIL, a panel method with a coupled integral boundary-layer formulation [15]. In order to avoid a point design, the optimization problem is a weighted multi-point drag minimization with constraints on the minimum airfoil thickness and minimum lift coefficient. Additional constraints on the nose radius and trailing-edge wedge angle are used.

Two separate approaches were used in this section, one for the clean-sheet design with B-Splines and another for the clean-sheet design with Béziers and for the progressive design. The reason for this was partly because independent work was done in parallel, but also to check the validity of the results by cross-comparing solutions. The clean-sheet with B-Splines used SNOPT, a large-scale sequential quadratic programming method [16], through the PyOptSparse package. Gradients were estimated using central differencing with a relative step size of  $10^{-4}$ . The objective, constraints, and design variables were also scaled to be of order one throughout the majority of the feasible design space. The convergence criteria was an absolute feasibility tolerance of  $1 \times 10^{-6}$  and an optimality tolerance of:  $(1 + \|g_0\|_\infty) \times 10^{-4}$  where  $g_0$  is the gradient at the starting point. This criterion mandates that the initial gradient be reduced by four orders of magnitude in order to be considered converged.

To help account for convergence issues within XFOIL, we established a routine to improve the robustness of the gradients. Frequently, a successful XFOIL run will be accompanied by a failure in one or two elements of the gradient vector. When this happened, we doubled the step size and re-ran XFOIL. We repeat this until we obtain a converged solution from XFOIL, or until a maximum of 10 iterations was exceeded. If the initial XFOIL analysis did not converge, then backtracking from a failure region using the methodology available in SNOPT was utilized.

The clean-sheet design with Béziers and the progressive design problems used the nonlinear SQP optimizer NLPQLP, which has been shown to be reasonably effective for problems with inexact gradients. We used the open-source PyOpt package, and estimated gradients using a forward difference scheme with a fixed absolute step size. All input variables were scaled using the formula  $10[\log_{10}(|v|)]$ , where  $v$  is the initial value of that variable. If the initial value is 0, then we use a scaling factor of 1.0. For output variables, we specified explicit scales for each of the objective and constraints to keep the values on the order of unity. The scaling factor for the objective is shown in the problem definition above, but the others are omitted for clarity. Owing to the relatively poor quality of estimated gradients, the KKT convergence tolerance was a relatively loose value of  $10^{-3}$ .

## B. Clean-Sheet Design Using Bézier Curves

We have already seen the equivalence between the CST parametrization and Béziers. In this first investigation, we compare the CST parametrization with only two parameters per surface with a ‘direct’ Bézier parametrization that also uses only two parameters. Both these parametrizations are ‘minimal’, and each design variable directly controls an aerodynamically salient feature of the geometry. The difference, however, is that one of them (the direct Bézier parametrization) lives in the space of cubic curves, and the other (CST) lives in a special subset of quintic curves. We ask two questions:

- 1) Is there any benefit to searching this special subspace of quintic curves, compared to just searching the space of cubic curves?
- 2) Is there any benefit to opening up the space of quintic curves and searching all of it?

We expect both answers to be in the affirmative. For this clean-sheet design investigation, we use 100 random combinations of nose radius and trailing-edge angles. For the generation of initial points, we reformulate the two trailing-edge angles in terms of a wedge angle  $\omega_{TE}$  and camber angle  $\gamma_{TE}$ , given by the sum and difference, respectively, of the individual boat-tail angles. From each starting point, we compare optimizations using three parametrization:

**CST2** Classic CST using Bézier shape functions and two parameters (nose radius and boat-tail angle) for each surface. There are four design variables in all, including angle of attack (because the nose radius design variable is the same for both surfaces).

**BEZ3** A cubic Bézier to directly describe airfoil shape using the same two parameters. This approach also has four design variables in all, including angle of attack.

**BEZ5** A quintic Bézier which uses these two parameters and two other intermediate control points. There are eight design variables including angle of attack.

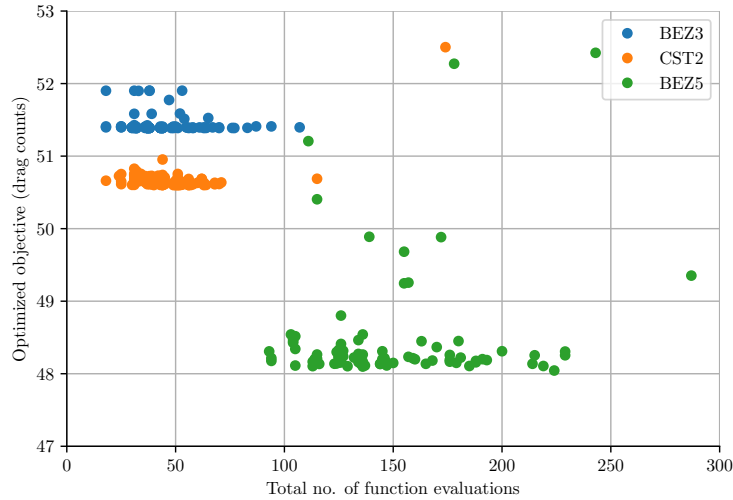
For consistency, we initialize each optimization using the quintic formulation by elevating the degree of the associated cubic Bézier. This way, optimizations using both the cubic and quintic parametrizations start from the exact same initial

shape. The formulation is shown in Eq. 22.

$$\begin{aligned}
 &\text{minimize: } \frac{2c_d(\alpha) + c_d(\alpha + 2^\circ)}{3} \times 10^3, \\
 &\text{with respect to: } r_{LE_u}, r_{LE_l}, \omega_{TE}, \gamma_{TE}, \alpha, \\
 &\text{subject to: } c_\ell \geq 0.4, \\
 &\quad t/c|_{x/c=0.2} \geq 0.1, \\
 &\quad t/c|_{x/c=0.7} \geq 0.05, \\
 &\quad r_{LE} \geq 0.007, \\
 &\quad \omega_{TE} \geq 14^\circ, \\
 &\quad \gamma_{TE} \leq 24^\circ.
 \end{aligned} \tag{22}$$

The results of the 100 optimizations for each approach are summarized in Fig. 11. The plot depicts each run via its optimized objective and the number of function evaluations (including those required for finite-difference gradients), so left and down is better. The answers to both questions raised in the previous paragraph are as expected:

- 1) Going from the cubic Bézier to the special CST subspace of quintic curves improves performance by about one drag count, as shown by the blue and orange circles.
- 2) Expanding from the subspace to the space of all quintic curves results in further drag reduction of about 2.5 counts, as shown by the orange and green circles. But this comes at a cost, both in terms of function evaluations, and in terms of increased variability. It is still unclear whether the variability is a result of finite-difference gradients in a larger design space, or the generation of shapes that are more liable to cause the solver to behave unexpectedly.



**Fig. 11 Optimization results using Bézier with 2 parameters per side, CST with 2 parameters per side, and quintic Bézier (the same polynomial degree as the 2-parameter CST formulation).**

If the number of design variables is fixed, the CST formulation parametrizes a family of curves of higher-order than a direct Bézier with the same number of design variables ( $y$ -locations of control points). Therefore, at least for a small number of design variables, we should always expect that the nonlinear formulation in CST allows the designer to access a subset of more flexible curves than a raw Bézier formulation with the *same* number of parameters. Going from searching the special CST subspace to the searching the entire space of Bézier curves of a given degree requires *twice* the number of parameters as CST. Either way, we are using high-order Bézier to parametrize either the shape function or the airfoil shape.

Both of these approaches should show rapidly diminishing returns as the degree of the curve grows large. Such high-order Bézier curves are relative inflexible, in that large control-point movements are required to “bend” them, and it is challenging to restrict the spatial extent of modifications. Owing to this “stiffness” of higher-order Bézier curves,



there should be a point where knot insertion is favored over degree elevation, and is in fact the reason B-Splines were invented. The choice between using a high-order Bézier and a cubic or quintic B-Spline is completely independent of using the CST transformation. One can use a B-Spline in at least two ways: one is to parametrize the shape function and use it in the CST context, or to directly describe the airfoil curve. In this paper, we do not compare all four combinations, but focus on comparing the use of classic CST using Bézier shape functions, with direct modeling of the airfoil shape using B-Splines. This is the topic of the next two sections.

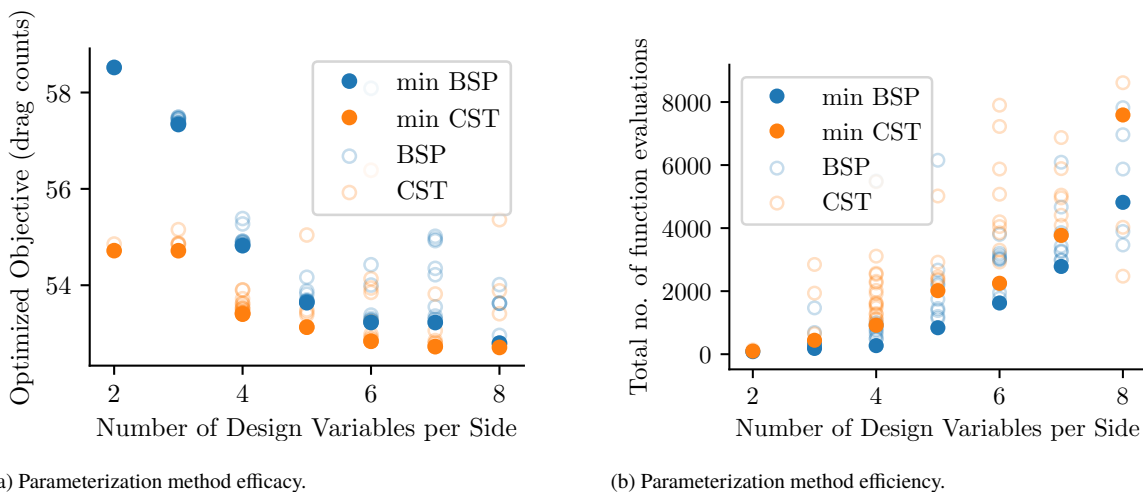
### C. Clean Sheet Design Using B-Splines

We performed another optimization set, this time using CST vs B-Spline parameterization methods over a larger range of design variables. We, again, used the XFOIL analysis tool, this time to obtain inviscid drag data. We employed another weighted drag minimization problem of the form:

$$\begin{aligned} \text{minimize: } & \frac{2c_d(\alpha) + c_d(\alpha + 2^\circ) + c_d(\alpha - 2^\circ)}{4} \times 10^3, \\ \text{with respect to: } & p, \alpha, \\ \text{subject to: } & c_\ell \geq 0.5, \\ & t/c \geq 14\%. \end{aligned} \tag{23}$$

where  $p$  are the airfoil shape parameters,  $\alpha$  is the angle of attack,  $t/c$  is the maximum airfoil thickness, and  $c_d$  and  $c_\ell$  are the drag and lift coefficients respectively. The objective was multiplied by  $10^3$  for scaling purposes.

In order to keep the comparisons fair, we used an identical set of starting points for each parameterization method and each number of parameters. We created this set from a latin hypercube sample which resulted in 20 unique starting points including variables for: leading edge radius, trailing edge boat-tail angle, trailing edge camber angle, and angle of attack. These same starting shapes were maintained as we increased the number of design variables.



**Fig. 12 Comparison of parameterization method efficacy (minimization of drag coefficient) and efficiency (number of total function calls) in a panel method aerodynamic shape optimization employing similarly parameterized CST and B-Spline (BSP) curves.**

For this problem, we compared the efficiency of each method by comparing the number of function calls to obtain the optimum. We compared the efficacy of each method by comparing the optima obtained. These results with respect to number of design parameters are shown in Fig. 12. As expected, the general trend is for the optimum to decrease and then level out as the number of parameters — and therefore control over the airfoil shape — increases. Additionally, the number of function calls grows with the number of design variables. There are two underlying reasons: one is that we are counting the number of evaluations required for finite-difference gradients, and that grows linearly in the number of design variables; the other is that convergence in a larger design space takes more iterations.

When the number of design parameters is small, CST performs better, as expected, and as we saw previously. As

the number of design variables is increased to a typical value used for design, the spread in the number of function evaluations also grows and the differences in the best solutions is within the accuracy of the analysis. We interpret these results to mean that the performance of CST and B-Splines for design is similar. Performance variations are liable to be highly problem-dependent. In fact, for a similar problem where just the thickness constraint was changed, the B-Spline approach had lower minimum drag values than did the CST approach. Though again, differences are small.

The equivalence between the approaches means that we can easily switch back and forth between approaches. For instance, one could start with CST, optimize, then add design variables by degree elevation, optimize again, and then switch to the direct B-Spline representation from then on. In general, one might expect that with accurate gradients and a smooth analysis that B-Splines would yield some advantage as seen in previous work [7, 8] and as shown with the airfoil fitting problems, whereas with a noisy analysis and inaccurate gradients a less flexible method like CST may be advantageous as shown here. However, we emphasize that these approaches are closely related, and it is probably not productive to try to pick a ‘winner’ *a priori*.

#### D. Progressive Design Using B-Splines

In Sec. V. B, the difference between the two direct Bézier representations was the degree of the curve. We obtained starting points for the higher-dimensional optimization problem using degree elevation. In this section, we will use knot insertion to add design variables to the B-Spline, and degree elevation to add design variables to the CST formulation. Moreover, since knot insertion and degree elevation do not change the shape of the curve, we implement progressive design. We further complicate matters by starting optimizations from approximations to existing airfoil coordinates. In other words, our procedure for B-Spline-based progressive design is as follows:

- Construct a best-fit B-Spline to a set of given airfoil coordinates, using an appropriate knot vector, as we did in Sec. IV. B.
- Optimize the airfoil from that best-fit starting design using a small number of design variables,
- Increment the number of design variables, use knot insertion to convert the current optimal shape to the new representation, and start the next optimization.

This process will seem familiar to those in the CFD community conversant with multigrid solvers. We compare the performance of this progressive design approach with CST, also using progressive design. We repeat the process for 3, 5, 7, and 9 design variables per side, resulting in optimization problems with 6, 10, 14, and 18 design variables. The optimization problem is the same as in Eq. 22, except with different design variables.

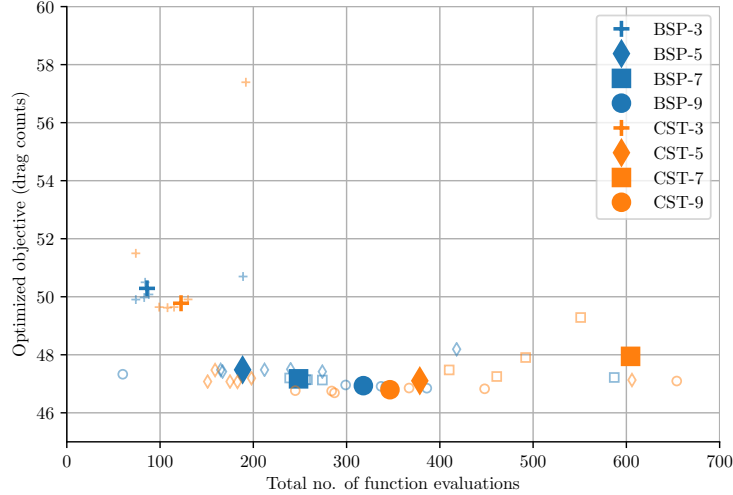
For initial designs, we start optimizations from six existing airfoils: NACA 2412, NACA 1408, NACA 63A210, tip sections L188e and C5e from the Lockheed Electra and the C-5 Galaxy respectively, and the Whitcomb supercritical section. All of these airfoils were obtained as sets of discrete coordinates from the UIUC airfoil database, and they were fit using the approaches described in Sec. IV.

These results are shown in Fig. 13. As before, points to the left of and lower than other points indicate better optimization performance. The large filled symbols denote median performance of each parametrization, where the median is taken across runs starting from the six different initial airfoil sections. The values are sorted by objective; the number of function evaluations for the associated run plays no role in the computation of the median. In other words, the bold symbols correspond to runs that had median objective, without regard to how many function evaluations that run consumed.

- As in the preceding section, with a very small number of design variables, the benefits of degree elevation afforded by CST yield an improvement in optimization performance, as shown by the ‘+’ symbols.
- Following knot insertion(s), the optimizations using five design variables per side show improved performance, as shown by the ‘diamond’ symbols.
- Going to 7 and 9 design variables yields hardly any further improvement, although the optimizations tend to stall more because we are using finite-difference gradients in higher-dimensional spaces.

## VI. Conclusions and Future Work

The goal of this paper is not to investigate whether any one parametrization is ‘better’ than another; nor is it to compare performance of these design variables on a specific problem with the goal of showing which one is better. Rather, the goal is to show that several existing parametrizations are equivalent, and one can easily transform between them. Designers seek insight and understanding of why a certain parametrization works, or works better than another. Why should one use a particular parametrization? Under what circumstances is it appropriate? How does that parametrization relate to the plethora of others? These are some of the issues we attempted to address in this paper.



**Fig. 13 Progressive design using B-Splines and CST. Progressive design starts with 3 design variables per side, then refines to 5, 7, and 9 variables per side. CST starts each new level afresh. The dark symbols represent median performance across the different starting airfoils for a given parametrization.**

We show that NACA 4-series, CST, and PARSEC parameterizations can be exactly represented by Bézier curves and thus by B-Splines. However, B-Splines have additional properties that these former methods do not such as the ability to restrict changes locally, to increase the number of design variables without increasing the degree of the curves, and to represent absolute or deformed geometry. We show how to easily incorporate geometric features like nose radius and boat-tail angle in the B-Spline definition. Fundamentally, a B-Spline is more flexible than the NACA/CST/PARSEC methods, and thus can more accurately fit a desired geometry, but that is not always beneficial in all optimization applications. Particularly if the analysis is noisy and gradients are inaccurate, a less flexible method may be desirable. However, B-Splines also naturally accommodate progressive design through knot insertion. This feature is effective for noisy analyses/gradients and for high fidelity analyses because one can start with a small number of design variables and progressively add design variables that allow for small changes about a nominal geometry.

In the context of airfoil definition and modification within the B-Spline airfoil framework, there are three main building blocks: knot insertion and degree elevation for adding design variables, and the CST nonlinear transformation that provides a ‘sneak peek’ into a higher-dimensional design (sub)space. These can be mixed and matched as the situation demands. For instance, one could use CST with a Bézier shape function of degree 5, and resort to knot-insertion for adding further design variables, thereby combining all three ‘building blocks’. Our own subsequent investigations will focus on mixing and matching these building blocks to glean insights into when and how to use them.

Our design experiments for this paper needed careful monitoring of scaling and tolerances, and sometimes necessitated customized gradient estimation techniques. If the design variables are all control point locations, whether of the shape function or the shape, then it is straightforward to show that for any finite tessellation, the surface grid positions are *linear* functions of the design variables. This holds for both CST and direct B-Spline modeling. The proliferation of adjoint solvers makes a convincing case for using these design variables, because that ensures that many or most functional gradients are not only highly accurate, but also inexpensive to compute. In fact, they can be computed ahead of time if the surface mesh topology is fixed. We will be exploring the application of these techniques in conjunction with adjoint CFD solvers for problems with more design variables.

An approach that we attempted briefly was to use CST with a B-Spline shape function. This is another straightforward extension that we hope to explore in the future; it promises a good balance of flexible shape control with the special nonlinear transformation of CST. One of the tradeoffs with CST seems to be that one must give up some amount of intuition to gain the benefits of a highly customized family of shapes for aerodynamic design that are still completely CAD-compatible. The change from Bézier to B-Spline shape functions may prove to be powerful. We will continue to explore these connections to bridge the gaps between geometry for CAD, for aerodynamics, and for shape optimization.

## Acknowledgements

The BYU authors gratefully acknowledge support from the The Connectivity Lab at Facebook.

## References

- [1] Kulfan, B., and Bussoletti, J., ““Fundamental” Parametric Geometry Representations for Aircraft Component Shapes,” *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2006. doi:10.2514/6.2006-6948, URL <https://doi.org/10.2514%2F6.2006-6948>.
- [2] 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, URL <https://doi.org/10.2514%2F3.58379>.
- [3] Rendall, T. C. S., and Allen, C. B., “Unified fluid–structure interpolation and mesh motion using radial basis functions,” *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559. doi:10.1002/nme.2219, URL <https://doi.org/10.1002%2Fnme.2219>.
- [4] Masters, D. A., Taylor, N. J., Rendall, T., Allen, C. B., and Poole, D. J., “Review of Aerofoil Parameterisation Methods for Aerodynamic Shape Optimisation,” *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2015. doi:10.2514/6.2015-0761, URL <https://doi.org/10.2514%2F6.2015-0761>.
- [5] Sobieczky, H., “Parametric Airfoils and Wings,” *Notes on Numerical Fluid Mechanics (NNFM)*, Vieweg Verlag, 1999, pp. 71–87. doi:10.1007/978-3-322-89952-1\_4, URL [https://doi.org/10.1007%2F978-3-322-89952-1\\_4](https://doi.org/10.1007%2F978-3-322-89952-1_4).
- [6] 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, URL <https://doi.org/10.1145%2F15886.15903>.
- [7] Castonguay, P., and Nadarajah, S., “Effect of Shape Parameterization on Aerodynamic Shape Optimization,” *45th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 2007. doi:10.2514/6.2007-59, URL <https://doi.org/10.2514%2F6.2007-59>.
- [8] Nadarajah, S., Castonguay, P., and Mousavi, A., “Survey of Shape Parameterization Techniques and its Effect on Three-Dimensional Aerodynamic Shape Optimization,” *18th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2007. doi:10.2514/6.2007-3837, URL <https://doi.org/10.2514%2F6.2007-3837>.
- [9] Wang, Z., Yu, S., and Liu, T., “Effect of Shape Parameterization on Aerodynamic Shape Optimization with SPSA Algorithm,” *Communications in Computer and Information Science*, Springer Berlin Heidelberg, 2014, pp. 393–402. doi:10.1007/978-3-642-53962-6\_35, URL [https://doi.org/10.1007%2F978-3-642-53962-6\\_35](https://doi.org/10.1007%2F978-3-642-53962-6_35).
- [10] Vassberg, J. C., and Jameson, A., “Influence of Shape Parameterization on Aerodynamic Shape Optimization,” *Influence of Shape Parameterization on Aerodynamic Shape Optimization*, Von Karman Institute, Brussels, Belgium, 2014.
- [11] Marshall, D. D., “Creating Exact Bezier Representations of CST Shapes,” *Proceedings of the 21<sup>st</sup> AIAA Computational Fluid Dynamics Conference*, 2013.
- [12] Piegl, L., and Tiller, W., *The NURBS Book (2nd Ed.)*, Springer-Verlag, Berlin, Heidelberg, 1997.
- [13] Sederberg, T. W., “Computer Aided Geometric Design,” , October 2014. Lecture Notes.
- [14] Reinsch, C. H., “Smoothing by Spline Functions,” *Numer. Math.*, Vol. 10, No. 3, 1967, pp. 177–183. doi:10.1007/BF02162161, URL <http://dx.doi.org/10.1007/BF02162161>.
- [15] Drela, M., “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” *Conference on Low Reynolds Number Airfoil Aerodynamics*, University of Notre Dame, 1989. URL [http://web.mit.edu/drela/Public/papers/xfoil{\\_.}sv.pdf](http://web.mit.edu/drela/Public/papers/xfoil{_.}sv.pdf).
- [16] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:10.1137/S1052623499350013, URL <http://epubs.siam.org/doi/10.1137/S1052623499350013>.