Hybrid B-Spline-Targets Airfoil Parametrization with a Direct Link to CAD-Based Aircraft Geometry

Časlav Ilić† and Patrick Wegener

German Aerospace Center (DLR)
Institute of Aerodynamics and Flow Technology, Braunschweig, Germany
†Corresponding author, caslav.ilic@dlr.de

Abstract. This work presents an airfoil parameterization method, called hybrid b-spline–targets (HBT), that enables a human designer or an optimization algorithm to separately control airfoil overall thickness, nose radius, camber distribution, and thickness distribution, while at the same time preserving a standard b-spline representation of the airfoil lower and upper surface in a CAD system, by means of an implicit parameter transformation external to the CAD system. The parametrization methodology, direct and inverse transformation between HBT and b-spline parameters, and an optimization demonstration with both types of parameters on a transonic wing-body transport aircraft configuration with RANS flow modeling are presented.

Keywords: CFD, aerodynamics, optimization, CAD, CATIA, parametrization, b-spline, implicit method

1 Introduction

In the field of aircraft aerodynamic and multi-disciplinary design there is a neverending quest for the most problem-suitable parametrized description of airfoil geometry. CAD systems usually provide higher-order polynomial-basis curves, such as b-splines or NURBS, with which airfoils can be modelled. For rubber-like deformation of given baseline 2D or 3D shape, including airfoil shapes, a popular parametrization method is FFD [7]. More tuned to the airfoil design-significant requirements are PARSEC [8] and CST [5].

Aerodynamically and structurally design-significant airfoil quantities include overall thickness, camber and thickness distribution, nose radius, trailing edge thickness, curvature distribution, boat-tail angle, height (or depth) at spar locations, and so on. Some of the mentioned parametrization methods do not, by themselves, represent any of those quantities (b-splines, NURBS), while other make some effort to do so (PARSEC, CST). All these methods are *explict* methods, where, for the given set of inputs, the output shape is computed by a direct application of a set of formulas.

In this work instead an *implicit* airfoil parametrization method is proposed, called *hybrid b-spline–targets* (HBT).

Within a CAD system that supports standard b-splines, in this case Dassault CATIA, airfoil lower and upper surface are represented by a single b-spline each. Both b-splines are clamped at their endpoints (leading and trailing edge) and may be of any degree up to the maximum degree for the used number of control points. At the trailing edge, the lower and upper b-spline clamping points do not coincide, to model a finite-thickness edge. The design parameters within the CAD system are thus the coordinates of the b-spline control points, specifically those normal to the chord line. Coordinates parallel to the chord line are kept constant, since they have a much lower effect on the aerodynamic properties.

Externally to the CAD system, a system of equations is solved that performs an invertible (bijective) transformation between the CAD b-spline control-point coordinates on the one side, and the design-significant set of airfoil geometric quantities on the other side. Observed are overall thickness, nose radius, camber distribution, and thickness distribution. Thus, a human designer or an optimization algorithm can use any subset of the design-significant quantities directly as design parameters, while losslessly maintaining the quality of the geometry representation provided by the CAD system.

2 Methodology

The starting point is a CAD model, in which there is an airfoil defined using one b-spline for the lower surface and another b-spline for the upper surface. Airfoil chord line always runs along the airfoil-local coordinate system x-axis, from 0 to 1 in whatever length units are used. The chord normal is along the z-axis. B-spline control points for the lower surface are designated $B_{L,i} = (x_{L,i}, z_{L,i})$, with $i_L = 0..n_L$, and same for the upper surface with index U.

Firstly, the control point coordinates $x_{L,i}$ and $x_{U,i}$ are taken as constant input for the CAD-external part of the parametrization. The number of control points n_L and n_U can be different, and no correspondence between $x_{L,i}$ and $x_{U,i}$ is necessary. The designer must distribute internal (away from leading and trailing edge) lower and upper control point indices into *camber sets*. Each camber set must contain at least one lower and one upper index, but can contain more of each. In case $n_L = n_U$ are equal, the natural choice is for each camber set to contain exactly one lower and one upper index, which are equal. Otherwise, indices can be grouped according to visual distances, such as on Fig. 1(a).

For each camber set, a *camber reference x-position* $x_{C,i}$, with $i_C = 1..n_C$, is computed as the average of all x-coordinates in the set. This is shown on Fig. 1(b). Note that it must hold $n_C < n_L - 2$ and $n_C < n_U - 2$. B-spline control points that will define the airfoil camber line are introduced, as $C_i = (x_{C,i}, z_{C,i})$. The chord-normal camber control point coordinates $z_{C,i}$ are the first part of HBT design parameters. The designer may use them to directly control the shape of only the airfoil camber line, as depicted on Fig. 1(c).

Next, the camber point (not control point) chord-normal coordinates $v_{C,i}$ are at $x_{C,i}$ are evaluated from the constructed camber b-spline. The b-spline may be of the degree as desired. Except for one selected camber set, i.e. one selected index

$i_{C,t}$, at each camber point a *thickness offset* $t_{C,i}$ is introduced. This thickness offset is subtracted from $z_{C,i}$ from the first lower-surface point index in the camber set, and added to $z_{C,i}$ for the first upper-surface point index in the camber set. The offsets $t_{C,i}$ form the second part of HBT parameters. The designer may use them to alter the airfoil thickness distribution without altering either the camber line or the overall airfoil thickness. At the previously excluded camber set index $i_{C,t}$, instead the overall airfoil thickness t is subtracted from/added to the lower/upper indices in that camber set. The designer may modify the overall thickness without altering either the camber line or the thickness distribution. The addition of offsets is shown on Fig. 1(d).

The z-coordinates obtained by subtracting and adding thickness offsets and overall thickness to camber points form the *hybrid lower/upper b-spline* control points, $H_{L,i} = (x_{L,i}, z_{H,L,i})$ and $H_{U,i} = (x_{U,i}, z_{H,U,i})$. If b-splines would be constructed from these control points, the resulting airfoil would have neither the previously defined camber nor the overall thickness. Thus, an extra *bump function* is assigned to each camber set. This is not entirely unlike a basis function in a finite-element method. When the basis function is multiplied with an unknown *bump factor* $b_{C,i}$ and added to the lower/upper hybrid b-spline control point z-coordinates, the final lower/upper airfoil-shape defining b-spline control points are obtained. Bump functions used in this work were simple hat functions, with minimal local support (function at $x_{C,i}$ has support between $x_{C,i-1}$ and $x_{C,i+1}$). The bump factor for the camber set carrying the overall airfoil thickness is denoted b_t. Fig. 1(e) depicts this setup. Bump factors have to be solved for (yielding an implicit method) such as that the constructed camber points and overall thickness are obtained by the final lower/upper b-splines. Bump factors are internal parameters of the method, they are seen neither by the designer nor by the CAD system.

In the camber sets which contain more than one lower-surface index, an additional $t_{L,i,k}$ offset is introduced, with $k_{L,i} = 1..n_{L,i}$ and $n_{L,i}$ being the number of lower-surface indices in the camber set. Analogously, $t_{U,i,k}$ offsets are introduced for any non-first upper-surface indices. These offsets are however subtracted from/added to the basic offsets of the first lower/upper index in the set. No additional bump functions or bump factors are assigned; the non-first hybrid control points in the set get contributions from previously introduced bump functions, according to their x-position.

Airfoil nose requires special treatment. The lower/upper final b-spline control points at the nose must remain fixed at $(0, 0)$. The second lower/upper final b-spline control points from the nose must instead have a zero x-coordinate. The lower/upper hybrid b-spline control point is constructed by subtracting/removing the desired nose radius r_n from the point $(0, 0)$. Here a separate bump function is introduced for lower and upper hybrid control point, yielding two bump factors. The bump function could be a half-hat (extending between $x = 0$ and $X_{C,1}$), but this results in a much too local influence of the nose radius on the airfoil shape. Instead, inverted half-ellipse bump functions are introduced, with explicitly stated extent x_n (e.g. $x_n = 0.1$ i.e. 10% chord). This is shown on Fig. 1(f). Nose radius

for the lower and upper surface must be equal, $r_n = r_{n,L} = r_{n_U}$, in order to have a C2-continuous shape at the leading edge. Thus, the nose radius r_n completes the HBT design parameter set. The two bump factors $b_{n,L}$ and $b_{n,L}$ have to be solved for to match the given nose radius, as well as in combination with the rest to match the constructed camber points and overall thickness.

The final step is to assemble the system of implicit equations for the bump factors b and solve it:

$$\begin{aligned} t(b_t, b_{C,1}, ..., b_{C,n_C}, b_{L,n}, b_{U,n}) &= t \\ v_{C,1}(b_t, b_{C,1}, ..., b_{C,n_C}, b_{n,L}, b_{n,U}) &= v_{C,1} \\ &\vdots \\ v_{C,n_C}(b_t, b_{C,1}, ..., b_{C,n_C}, b_{n,L}, b_{n,U}) &= v_{C,n_C} \\ r_{n,L}(b_t, b_{C,1}, ..., b_{C,n_C}, b_{n,L}, b_{n,U}) &= r_n \\ r_{n,U}(b_t, b_{C,1}, ..., b_{C,n_C}, b_{n,L}, b_{n,U}) &= r_n \end{aligned} \tag{1}$$

This produces the final lower/upper b-spline control points, which define the airfoil shape that is having all of the input overall thickness, camber shape, and nose radius properties.

It must hold that the sum of the number of final lower/upper b-spline control-point coordinates ($z_{L,i}$ and $z_{U,i}$) is equal to the sum of the number of HBT parameters (one overall thickness parameter t, number of camber control-point coordinates $z_{C,i}$, number of thickness offsets $t_{C,i}$, $t_{L,i,k}$, $t_{U,i,k}$, and one nose radius r_n). In this way an invertible transformation between HBT parameters and final b-spline control-point coordinates is well defined.

In practice, a CAD model will come with some sensible airfoils already defined, i.e. with the lower/upper b-spline control point z-coordinates given. In that case it is necessary to reconstruct the HBT parameter set by an inverse transformation. Only after that the designer may start using the HBT parameters themselves. The overall thickness t and nose radius r_n HBT parameters may be evaluated directly from the given lower/upper b-splines. The internal bump factors b_t, $b_{C,i}$ and $b_{n,L}$ and $b_{n,U}$ can be solved for from Eq. 1. However, additional equations are needed to solve for camber b-spline z-coordinates $z_{C,i}$ and thickness offsets $t_{C,i}$, $t_{L,i,k}$, $t_{U,i,k}$. These equations are introduced by matching on the camber point z-coordinates $v_{C,i}$ (which may be evaluated directly) to get $z_{C,i}$, and by matching on a combination of the lower/upper final b-spline control point coordinates $z_{L,i}$ and $z_{U,i}$ (which are given) to get $t_{C,i}$, $t_{L,i,k}$, $t_{U,i,k}$. The word "combination" is used for the latter, because there are more control point coordinates than the thickness offsets; in infinite precision it would be sufficient to match on any subset of the required size, but to avoid uncontrolled residuals at excluded points in finite precision, it is better to match on weighted groups including all points.

At no place does the inverse transformation apply regressive matching, e.g. by a least-squares method. Instead, in infinite precision the roundtrip inverse-direct transformation would yield exact same final b-spline control points. With the applied root-finding method from the Python SciPy package, the roundtrip produces control point coordinates equal up to 5–7 significant digits. This is considered comfortably sufficient in practice.

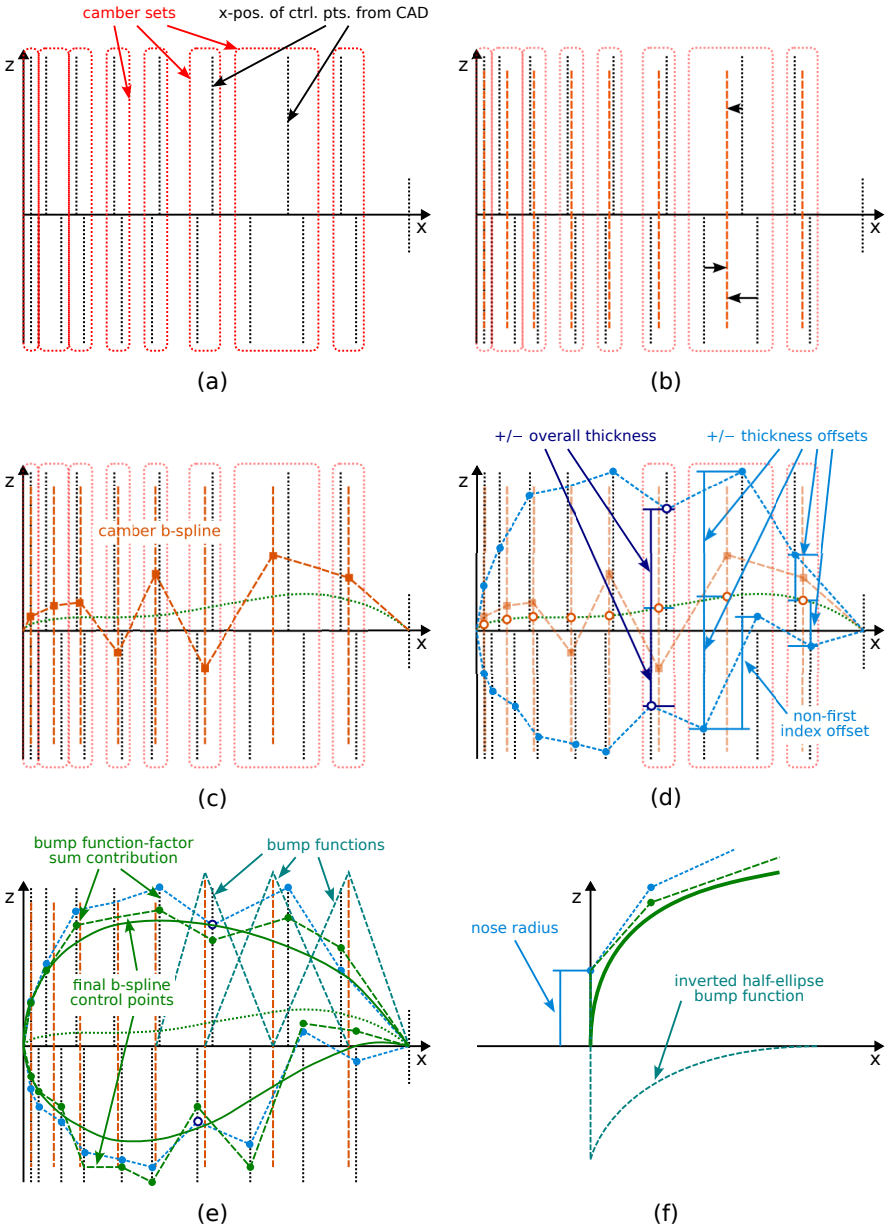

Fig. 1: Steps in construction of lower/upper b-splines from HBT parameteres.

One last note about the methodology concerns the definition of the "overall thickness" of an airfoil. This is not a uniquely defined quantity. For the HBT parametrization, a smoothly varying (i.e. not max/min-like) quantity is needed,

yet one which captures the influence of airfoil thickness on the sizing of the internal wing structure. Thus *inertial thickness* is used, defined as the thickness of a "rectangle airfoil" that would have the same second moment of inertia (resistance to bending moment) as the actual, curved airfoil shape. Furthermore, inertial thickness is evaluated not over the complete airfoil length, but only over the section between the front and the rear spar (the part which forms the structural wing box). The chord-relative x-positions of the spars are supplied by the designer and kept fixed.

3 Demonstration

As the basis for demonstration, the DLR-F25 configuration is used. DLR-F25 was originally developed within the LuFo VI-2 project VirEnfREI funded by the German Ministry for Economic Affairs and Climate Action (BMWK).[4] Only the wing-body-horizontal tailplane part of the configuration is used in this work, while the vertical tailplane, engine, and pylon are left out. The CFD mesh of this configuration, constructed with the ANSA mesh generator and having 10 million points, is shown on Fig. 2.

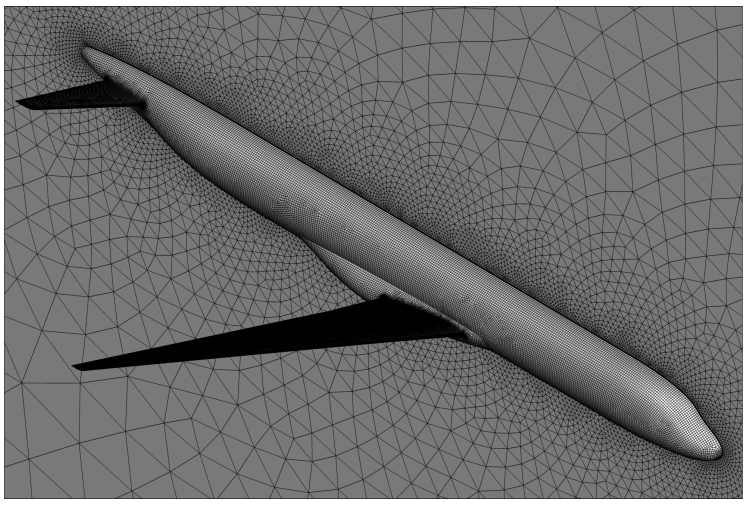

Fig. 2: Hybrid-unstructured mesh of the wing-body-HTP DLR-F25 configuration.

In several contexts, it was desired to perform a pure aerodynamic optimization of wing airfoils, but such that the baseline wing thickness, spanwise load distribution and low-speed performance are preserved. With HBT parametrization, this requirement can assured to a good extent by modifying only the thickness offset parameters ($t_{C,i}$, $t_{L,i,k}$, $t_{U,i,k}$). However, with pure b-spline parametrization it can be likewise assured, by using all b-spline control point coordinates as design

parameters ($z_{L,i}$, $z_{U,i}$) while adding explict constraints on the overall thickness, nose radius, and camber distribution (to be equal to baseline). Thus these two optimization setups are compared here.

There are a total of six airfoils along the span defined within the CAD model in CATIA: at wing root, planform kink, tip, one airfoil between root and kink, and two airfoils between kink and tip. For HBT parametrization, Fig. 3 illustrates the effect of a roundtrip inverse-direct transformation for the root airfoil (shape remains same), and the resulting set of overall thickness, nose radius, camber control points, and thickness offsets. Run time for direct and inverse transformation is on the order of a few seconds per airfoil. There are a total of 19 HBT parameters per airfoil, out of which 9 are thickness offsets, resulting in a total of 54 HBT parameters for optimization. For pure b-spline parametrization, there are 19 b-spline control point coordinates per airfoil (same as number of HBT parameters), for a total of 114 b-spline parameters in optimization; additionally there are 10 explicit constraints per airfoil (overall thickness, camber points, lower/upper nose radius), resulting in 60 constraints in total. Note that the number of b-spline parameters minus the number of explicit constraints equals the number of HBT parameters.

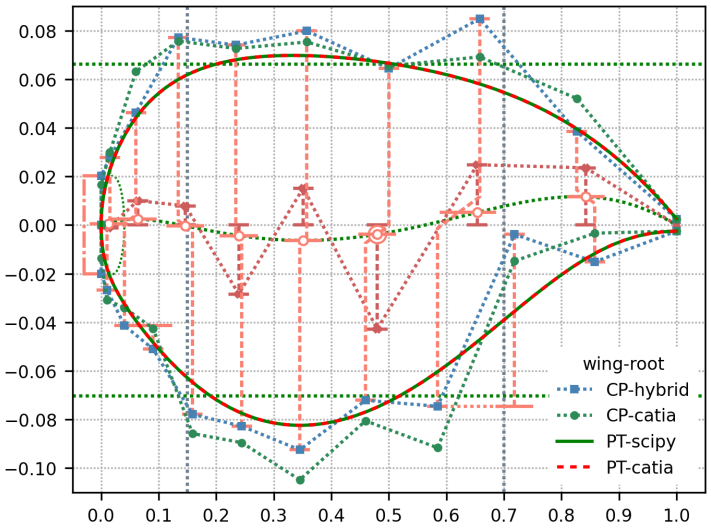

Fig. 3: HBT parameter reconstruction of the DLR-F25 root airfoil.

An adjoint-gradient based unconstrained (in case of HBT) or constrained (in case of b-splines) optimization method is used. The flow and discrete-adjoint gradients were computed by the DLR TAU flow solver [2, 1], using RANS flow with SA turbulence model. The optimization algorithm for both cases was SNOPT [3]. CATIA geometry updates were handled through a reduced-order model (ROM) of

a topologically constant "point cloud" [6], such as to avoid having to call CATIA from within the optimization loop. The updated CFD mesh was obtained by surface mesh interpolation from the ROM-evaluated point cloud and volume mesh deformation according to the linear-elasticity analogy. The optimization objective was a weighted drag coefficient C_D at constant lift coefficient C_L, at three flight points $C_L = [0.55, 0.60, 0.65]$ with weights $w = [0.2, 0.5, 0.3]$ respectively. Mach number was 0.78 for all points. No horizontal tailplane trimming was performed.

Optimization histories are given by Fig. 4. Total run time of each optimization is about 60 hours; per flight point 384 processor cores are used, for 1152 cores in total. Lift-to-drag polars of baseline and two optimized configurations are compared on Fig. 5, and spanwise load and lift coefficient distribution on Fig. 7 and Fig. 8. Airfoil shapes and chordwise pressure distribution for two selected airfoils are shown on Fig. 6. Wing thickness and nose radius were fixed by construction in the HBT case, while the optimizer preserved them accurately in the pure b-spline case. Due to the fixed camber shape, spanwise load distribution changed very little. This small change is a pure non-linear, transonic effect, which would not appear with linear aerodynamic methods (such as used e.g. for computing design loads for structural sizing).

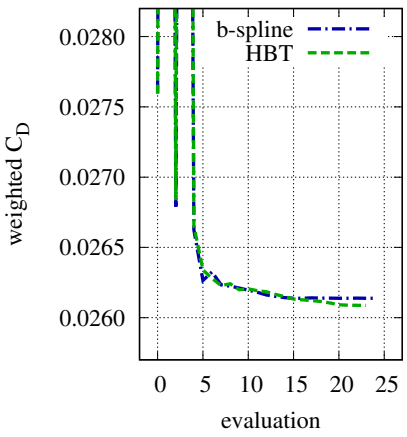

Fig. 4: Convergence histories.

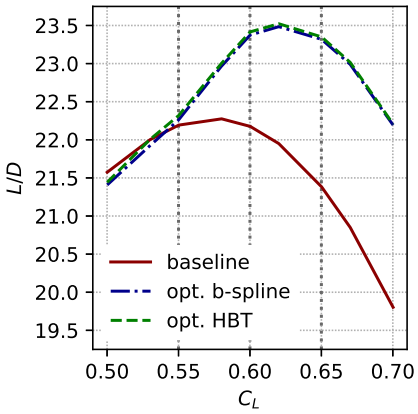

Fig. 5: Lift-to-drag polars.

In effect, there is no difference of any significance between the unconstrained optimization run with HBT parameters and the constrained run with pure b-spline parameters, either in optimization convergence or in optimal shapes and their properties.

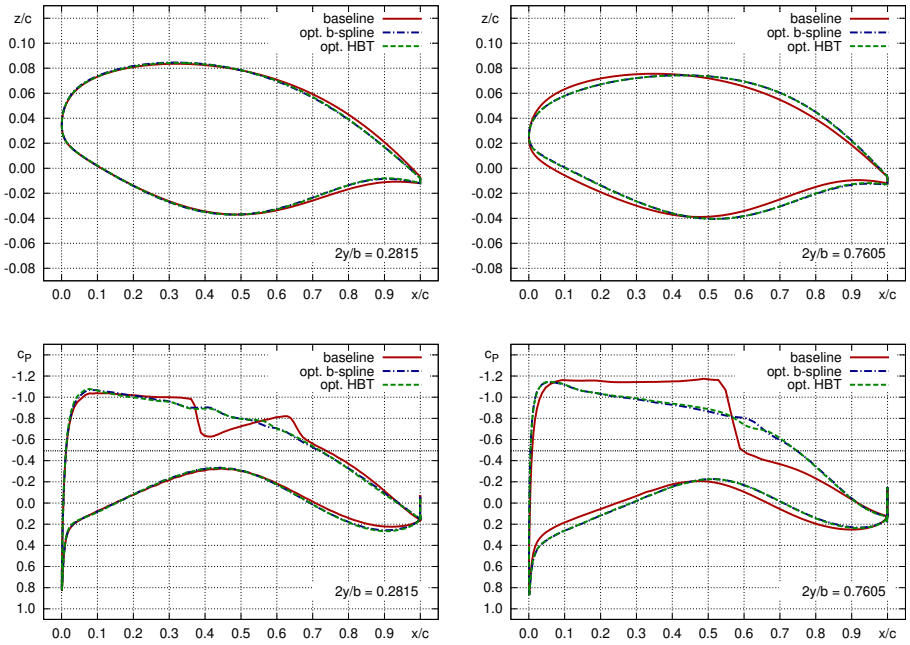

Fig. 6: Airfoil shapes and pressure distributions at two selected relative spanwise positions $2y/b$.

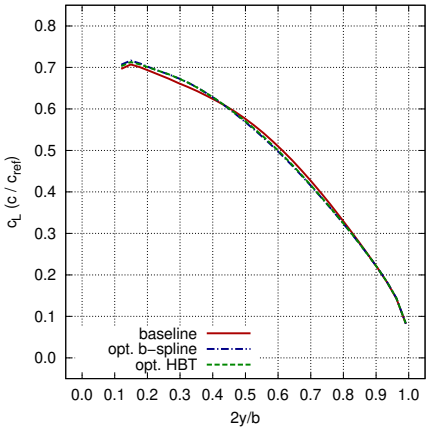

Fig. 7: Spanwise load distribution.

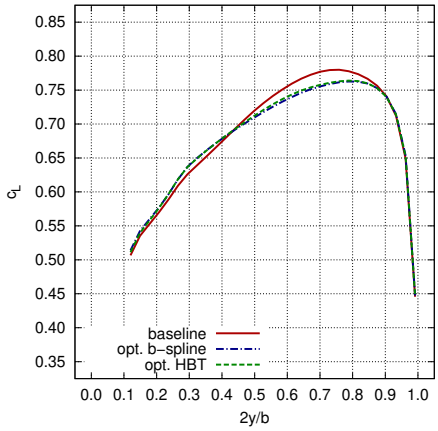

Fig. 8: Spanwise lift distribution.

4 Discussion

Based on the results in Sec. 3, one could ask what is the benefit of introducing the HBT parametrization method, with yet another layer of indirection in

the optimization process and possibly still unforseen problems, when the more straightforward approach with a well-known, b-spline parametrization and explit constraints works just as well. Indeed, for an optimization problem of the kind as presented, there does not appear to be any advantage.

However, the salient aspect of the presented optimization problem is that a full set of rather accurate derivatives was available, and that it was quick to evaluate with an adjoint approach. In more complex optimization problems, this may not be the case. With more complex configurations (such as with a powered engine or control and high-lift surface deflections), derivatives may come out with substantial noise. In multidisciplinary problems there may be intentional approximations in the cross-disciplinary couplings, leading to a systematic inaccuracy of derivatives. It is in these cases where a more orthogonal parametrization, such as HBT, could provide an advantage.

In this work itself there was an unintentional demonstration of such an advantage. It has been mentioned that a ROM-based update of the CFD mesh was performed. For constructing the ROM, a DOE set of snapshots from CATIA had to be evaluated prior to starting the optimization, one DOE per parametrization type. There it was observed that for the HBT parametrization, the number of failed snapshots was 0.2% of the total, indicating that almost all HBT parameter combinations were yielding well behaved shapes. For the pure b-spline parametrization, the failure rate was 8.5%.

In that light, the results from Sec. 3 can be viewed as an indicator that the HBT parametrization does not behave in any unexpected way, does not introduce any large non-linear effects beyond basic b-spline behavior, and so far seems to be safe to use wherever such a split of design parameters is desired.

References

1. Brezillon, J., Dwight, R.P.: Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3D configurations. CEAS Journal (2011)
2. Gerhold, T., Galle, M., Friedrichs, O., Evans, J.: Calculation of complex three-dimension al configurations employing the DLR TAU-Code. In: 35th Aerospace Sciences Meeting and Exhibit, AIAA-97-0167 (1997)
3. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Journal on Optimization **12**, 979–1006 (1997)
4. Ilic, C., Wegener, P., Ruberte Bailo, J., Himisch, J., Geisbauer, S., Wunderlich, T., Abu-Zurayk, M.: Phased high-fidelity aerodynamic design from scratch of a very high-aspect ratio narrow-body airliner. In: AIAA Aviation 2024 (2024)
5. Kulfan, B.M., Bussoletti, J.E.: "Fundamental" parametric geometry representations for aircraft component shapes. In: 11th AIAA/ISSMO Conference (2006)
6. Merle, A., Ronzheimer, A., Bekemeyer, P., Görtz, S., Keye, S., Reimer, L.: Gradient-based optimization of a flexible long-range transport aircraft using a high-dimensional CAD-ROM parameterization. In: DLRK 2018 (2018)
7. Ronzheimer, A.: Shape based on freeform deformation in aerodynamic design optimization. In: ERCOFTAC Design Optimization International Conference (2004)
8. Sobieczky, H.: Parametric airfoils and wings. Notes on Numerical Fluid Mechanics **68** (1998)