

# RBF-Based Morphing of B-Rep Models for Use in Aerodynamic Shape Optimization

Flavio Gagliardi ([flavio@central.ntua.gr](mailto:flavio@central.ntua.gr))

Kyriakos C. Giannakoglou ([kgianna@central.ntua.gr](mailto:kgianna@central.ntua.gr))

*National Technical University of Athens (NTUA), School of Mechanical Engineering,  
Parallel CFD & Optimization Unit, Athens, Greece.*

## Abstract

This paper presents a shape parameterization method based on morphing that acts directly on CAD-compatible Boundary-Representations (B-Rep), effectively integrated into aerodynamic shape optimization. The proposed technique requires the definition of a small number of "handles", which are strategically placed around or on the B-Rep shapes to be optimized. Displacement vectors associated with these handles are used as design variables in the optimization method. Using Radial Basis Functions (RBF) as an interpolation method, these displacements are transferred from the handles to the Non-Uniform Rational Basis-Splines (NURBS) control points of the B-Rep model; this approach offers the advantage that the updated surface remains in CAD format and is thus exportable to a STEP file. The proposed method combines two successive deformation steps. Each deformation is controlled by an independent set of handles, increasing the flexibility of the morphing action. A strategy for updating the CFD surface grid to the already changed B-Rep models enables the seamless inclusion of the proposed method into any optimization loop, avoiding any into-the-loop dependence on grid generation packages. The surface grid is updated by computing new nodal coordinates based on the updated NURBS parametric coordinates; these are computed according to changes in the parametric domain of trimmed surfaces by an RBF-based interpolation. The displacement of the surface nodes is then used to displace the CFD volume grid. The tool is differentiated and integrated into gradient-based optimization loops using the adjoint method to compute the gradient of the objective and constraint functions with respect to the surface nodal positions. The performance of the proposed method is assessed in four aerodynamic shape optimization problems, concerning a 2D airfoil, a duct, an aircraft model and a compressor stationary blading. An assessment of the proposed method based on parametric effectiveness, in the 2D case, is also presented.

## 1 Introduction

The goal of aerodynamic or hydrodynamic shape optimization is to find shapes with improved performance according to one or more criteria and constraints. This process typically starts with the creation of a (reference) model using a Computer-Aided Design (CAD) software, followed by grid generation and Computational Fluid Dynamic (CFD) simulations. The optimization process generates a sequence of design variations that exhibit progressively improved performance, according to evaluations on CFD tools. Shapes and grids involved in CFD-based shape optimization require parameterization; as the task that determines the optimization or design variables, parameterization affects the computational cost and the success of the optimization. In fact, it defines the "types" of reachable shapes and the search space to be explored.

The re-parameterization of geometries available in standard Boundary-Representation (B-Rep) formats, such as STEP files, in which shapes are represented by a collection of connected patches, is a key component of the proposed method. Overall, surface parameterization techniques for use in shape optimization should:

- be able to handle complex, yet smooth, regular and realistic shapes and allow the imposition of geometric constraints;
- ensure a wide range of reachable shapes with the minimum number of design variables because the complexity of shape optimization scales with this number [1];
- give rise to properly scaled design variables to facilitate the convergence of the optimization loop [2];
- be able to easily compute derivatives of surface nodal coordinates with respect to (w.r.t.) the design variables; in gradient-based optimization, this should be combined with the adjoint method that

computes the gradients of the objective and constraint functions w.r.t. the surface nodal coordinates [3];

- offer the ability to deform the existing CFD surface grid by adapting it to the new shape, which allows the displacement of the volume grid to fit the new boundary, so as to avoid re-meshing;
- retain compatibility with the CAD software for further processing in industrial workflows.

Shape parameterization has been an area of continuous and extensive research; therefore, a vast amount of relevant literature can be found. A detailed survey of shape parameterization techniques for design optimization is provided in [4]; these techniques can be classified into grid-based and CAD-based ones.

Grid-based parameterization relies upon either direct nodal deformations or space morphing techniques. The former perturb surface grid nodes and give rise to the most extensive design space but require smoothing to avoid unrealistic designs or invalid grids [5]. On the other hand, space morphing techniques deform the space in which grids are embedded. They speed-up the optimization process by avoiding re-meshing and enable the continuation of new CFD simulations from a previous one on a different domain. Space morphing/deformation functions  $d : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  assign a displacement to each point in space. Displacements imposed on a set of points ("handles") determine the deformation field. The most common space deformation techniques, such as lattice-based Free-Form Deformations (FFD) and Radial Basis Functions, are compared in [6]. RBF handles are placed arbitrarily in space (point-based), in contrast to lattice-based FFD, in which handles coincide with the vertices of polyhedra. Examples of RBF-based parameterization are presented in [7] and [8]. In [7], airfoil shapes in discrete form are controlled by RBF handles placed off the shape. In [8], the RBF model is employed to parameterize and optimize sails trim by applying rigid displacements to sets of surface nodes. Overall, grid-based parameterizations succeed in building effective design spaces, even for complicated shapes; however, any link with the CAD model is lost during the optimization. Upon completion of the optimization loop, it is desirable to export the optimized shapes in a CAD format, though this unavoidably introduces approximation error and may compromise the obtained performance gain. Handling shapes in other than CAD formats, during the optimization loop, generally makes it harder to impose geometric constraints. For instance, keeping surfaces of revolution, such as the casing in turbomachinery applications, intact while morphing the blade surfaces requires handling the intersection of the blade with the casing or keeping these intersections fixed; both operations are readily available in CAD systems.

CAD-based methods can be classified as constructive, NURBS-based and space morphing. In constructive CAD tools, shapes are defined starting from standard geometric features, such as curves, extrusions and holes. This can be done using generic CAD packages or application-specific parameterization software. For instance, in turbomachinery design, it is common to have dedicated CAD kernels, such as the one presented in [3], which use spanwise and chordwise distributions of various geometric quantities (angles and thicknesses), usually in the form of polynomial or NURBS curves. Constructive CAD tools generate geometries from scratch and, if necessary, can vary the topology of the model. They are defined by a sequence of instructions intrinsically accommodating geometric constraints but they may over-constrain the designed shapes. Moreover, constructive CAD methods are not easily differentiable [9]. Pure NURBS-based methods may overcome this problem [10]. Such methods use NURBS control point coordinates, contained in B-Rep models, to directly deform the shape, allowing for rich design spaces. It is challenging to generalize such methods to cases with many trimmed patches by keeping adjacent NURBS surfaces watertight and eventually tangent when control points are displaced. In [11], a method for respecting geometric continuities over NURBS patch interfaces, which relies on the discrete filtering of the NURBS control point displacements, is presented; this provides a rich design space and shape derivatives. However, NURBS-based methods can be impractical in some industrial cases, the CAD models of which may contain thousands of NURBS control points. The third class of methods, namely space morphing techniques applied to B-rep models, displaces the NURBS control points indirectly; the herein proposed parameterization technique belongs to the latter. These introduce just a few design variables and provide smooth shape variations, even for CAD models with thousands of NURBS control points; however, these methods are not commonly found in the literature. An approach based on FFD, proposed in [12] for structural analysis, applies the same transformation to both the CAD model and the computational grid. However, the resulting CAD model only approximates the corresponding grid surface and, in [12], the method is demonstrated only for small shape modifications. Commercial CAD packages, such as Rhino [13], offer the ability to use morphing boxes to alter single NURBS surfaces. The software employed in [8] can approximately bring the shape of an RBF-morphed grid back to CAD.

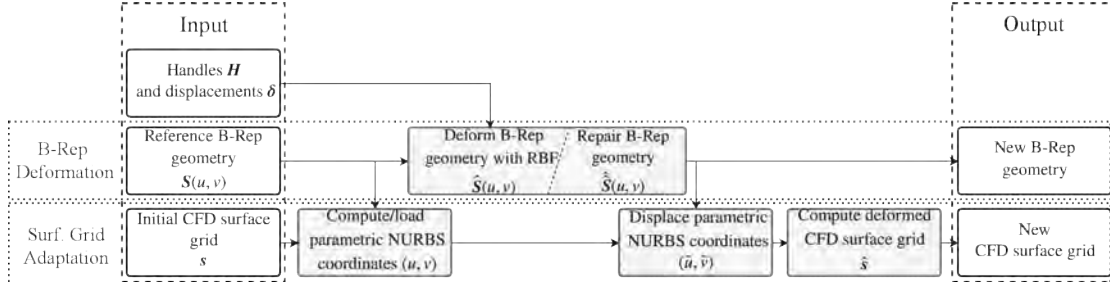


Figure 1: Main phases of the parameterization procedure based on shape morphing and CFD surface grid deformation. Within an optimization loop, the deformation of the volume CFD grid follows. The multi-step B-Rep deformation procedure is described in Section 3. The surface grid adaptation procedure is described in Section 4.

In [14], a method for editing B-Rep models via push-pull direct modeling of patches is proposed. Despite its robustness, it cannot easily be used for shape optimization.

Generally, in shape optimization, CFD grid deformation techniques are preferable, because directly changing the CAD model should be followed by re-meshing. The proposed method overcomes this shortcoming of CAD-based optimization by simultaneously morphing the CFD surface grid according to the morphed CAD. The proposed shape-morphing strategy can handle complex geometries with a user-defined number of parameters, which result in well-scaled design variables for use in shape optimization. Shape morphing is based on the RBF interpolation model and decouples the construction of the reference geometry from the parameterization to be used during the optimization. Regarding the optimization itself, gradient-based methods supported by the continuous or discrete adjoint method for computing the objective function gradient have become very attractive in CFD-based optimization with many design variables, mainly because adjoint methods compute gradients at a cost independent of their number. However, adjoint methods primarily compute sensitivities of an objective function  $\mathcal{F}$  w.r.t. the nodal coordinates on the grid surface, namely  $\partial\mathcal{F}/\partial\mathbf{x}$  [3]. To obtain the sensitivities of  $\mathcal{F}$  w.r.t. the design variables  $\mathbf{b}$ , the parameterization must be differentiated (i.e.  $\partial\mathbf{x}/\partial\mathbf{b}$  must be computed; geometric sensitivities) and linked to the sensitivities computed by the adjoint method through the chain rule

$$\frac{\partial\mathcal{F}}{\partial\mathbf{b}} = \frac{\partial\mathcal{F}}{\partial\mathbf{x}} \frac{\partial\mathbf{x}}{\partial\mathbf{b}} .$$

Various methods exist for computing geometric sensitivities, such as Finite Differences (FD), algorithmic differentiation and analytic differentiation. The proposed method makes use of FD. Thanks to the strategy implemented to displace the CFD surface grid according to shape variations, geometric sensitivities are easily computed at the surface grid nodes.

The paper is structured as follows. Section 2 introduces the proposed parameterization method based on shape morphing and describes the necessary background tools. The method is presented in detail in Section 3. Section 4 outlines the strategy for updating the CFD surface grid according to the updated CAD model. Section 5 applies the programmed software to four problems.

## 2 Shape-Morphing Strategy and Theoretical Background

Before presenting the details of the proposed shape-morphing technique in Section 3, Section 2.1 concisely summarizes the main phases of the parameterization procedure and CFD surface grid deformation. They are based on two mathematical entities, NURBS and RBF, for which the concepts and notations used in the remainder of the paper can be found in Sections 2.2 and 2.3.

### 2.1 Shape-Morphing Strategy

Figure 1 illustrates the workflow defining the shape parameterization strategy based on morphing. Firstly, the reference shape is imported in the form of a B-Rep model, practically as a STEP file. Because STEP files might contain non-NURBS entities, these are converted to NURBS and thus made compatible with what follows. Handles used to deform the shapes are then positioned on or around these NURBS patches.

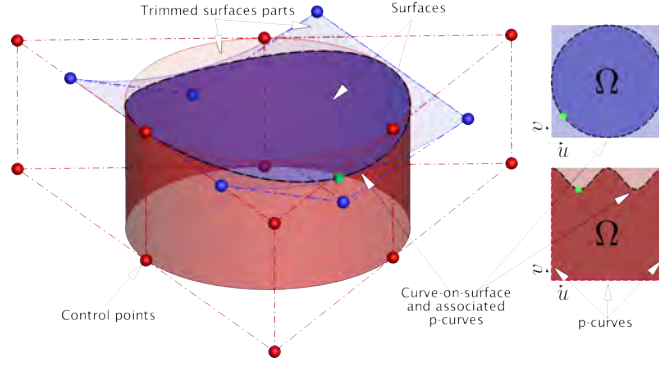


Figure 2: Left: Two intersecting/trimmed NURBS surfaces and bi-directional control nets. The shared curve (dashed) and trimmed parts of the surfaces, invisible in the CAD model, are illustrated. Right: Trimmed parametric domains of the two surfaces along with the dashed p-curves corresponding to the shared curve. The green point is one of the many test points controlling geometric continuities, see Section 3.2. It is represented on the curve-on-surface and the two p-curves.

To increase morphing flexibility, a series of successive deformation actions, each corresponding to a set of handles, is employed; according to it, shapes are incrementally deformed (multi-step deformations). The displacements of the handles are used as design variables during the optimization. Section 3 presents the details of the procedure for deforming sets of NURBS surfaces, following the displacement of handles, while satisfying geometric continuities. This procedure results in modified B-Rep models, to be exported as STEP files. Associated with the same task is the deformation of the CFD surface grid of the reference shape, by adapting it to the deformed B-Rep model. To this end, the NURBS parametric coordinates of each surface grid node are needed. If not available from the grid generation software, these should be reconstructed by projecting grid nodes onto the NURBS parametric space of the patches pertaining to the reference B-rep model. Once the shape has been morphed, the trimming-curves displacements of the NURBS surfaces in the parametric space drive the RBF-based displacement of the nodal parametric coordinates; the so-computed new nodal parametric coordinates determine the new positions of the surface grid nodes. The last phase is the displacement of the 3D CFD grid, for which the RBF-based method described in [15] is used; we refrain from presenting the details of this method.

## 2.2 Basics of NURBS and Relevant Issues

NURBS are analytical functions describing 2D or 3D curves up to complex 3D surfaces. Because of their flexibility, NURBS are frequently employed in CAD systems. In standard vendor-neutral file formats for exchanging information among CAD systems, such as the STEP file format, B-Rep primarily (though not exclusively, as noted above) relies upon NURBS patches, for the purpose of portability. Using NURBS patches to represent shapes enables the exchange of information in standard CAD format.

A NURBS curve is a parametric curve mathematically defined as

$$C(u) = \sum_{i=0}^I R_i(u) P_i, \quad (1)$$

where  $P_i$  are the control points forming a control polygon/net,  $R_i(u)$  are the rational basis functions and  $u$  is the parameter. If  $P_i \in \mathbb{R}^2$ ,  $C(u)$  is a curve on a plane else, if  $P_i \in \mathbb{R}^3$ ,  $C(u)$  is a curve in the 3D space. Similarly, a NURBS surface is a two-parameter ( $u$  and  $v$ ) function, defined as

$$S(u, v) = \sum_{i=0}^I \sum_{j=0}^J R_{i,j}(u, v) P_{i,j}. \quad (2)$$

In this case, control points  $P_{i,j} \in \mathbb{R}^3$  form a bi-directional control net [16].

NURBS surfaces have the limitation of being four-sided patches; thus, to create complex shapes, many trimmed patches must be combined. Trimming is the operation that "cuts" surfaces using curves lying on them, increasing the flexibility in shape representation. For instance, a surface with a hole

Table 1: RBF activation functions.

Name	$\phi(r)$	Used for	See Section
Inverse multiquadric	$((r/\sigma)^2 + 1)^{-1/2}$	Shape morphing	3.1
Wendland C2	$\begin{cases} (1 - r/\sigma)^4 (1 + 4r/\sigma) & \text{if } r < \sigma \\ 0 & \text{if } r \geq \sigma \end{cases}$	Shape morphing	3.1
Wendland C0	$\begin{cases} (1 - r/\sigma)^2 & \text{if } r < \sigma \\ 0 & \text{if } r \geq \sigma \end{cases}$	Continuity fixing	3.2
Thin plate spline	$r \log r^r$	Displacement of parametric coordinates	4

can be represented by a single trimmed surface. The trimmed-away portions of the surface are neither discarded nor rendered when showing the B-Rep model. Curves used for trimming surfaces are called curves-on-surface ( $C_{cs}$ ) or p-curves ( $C_{pc}$ ) if represented in the parametric space of surfaces sharing the 3D curve-on-surface. P-curves and curves-on-surface are also used to describe the natural edges of surfaces. The portion of the surface that is actively part of the B-Rep model is the parametric domain  $\Omega$ , defined by a set of p-curves forming one or more loops in the parametric space of a surface, as illustrated in Figure 2, where the shared  $C_{cs}$  corresponds to two  $C_{pc}$  in the parametric domains of the two intersecting surfaces. Trimming is a crucial operation for building complex shapes. However, it may introduce gaps between intersecting NURBS patches; continuity across these interfaces (to a specified tolerance) is ensured by appropriate control points positioning rather than by making control points on the edges of the surfaces coincide. In a B-Rep model,  $G0$  continuity does not require that adjacent NURBS surfaces share the same number or distribution of control points. Figure 2 illustrates a simple example in which the control points of a curve-on-surface do not coincide with those of the adjacent surface. Such configurations create geometric continuity issues when B-Rep models are changing during an optimization loop; a remedy to this is presented in Section 3.2.

### 2.3 Basics of RBF-Based Interpolation and Relevant Issues

RBF interpolation is extremely versatile since it can interpolate values given at scattered points by returning the exact values at those points. In this study, quantities to be interpolated are the known 2D or 3D displacements defined at distinct source nodes, which are (i) the handles of the (3D) parameterization in Section 3.1, (ii) NURBS control points (3D) in Section 3.2, or (iii) NURBS parametric coordinates (2D) and (iv) grid nodes (3D) in Section 4. Radial basis functions are real-valued functions  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  depending only on the distances of a point  $\mathbf{x} \in \mathbb{R}^Q$  from the so-called RBF interpolation sources  $\mathbf{x}_n \in \mathbb{R}^Q, n \in [1, N]$ . The RBF deformation function  $\mathbf{d} : \mathbb{R}^Q \rightarrow \mathbb{R}^Q$  takes the form

$$\mathbf{d}(\mathbf{x}) = \sum_{n=1}^N \mathbf{c}_n \phi(\|\mathbf{x} - \mathbf{x}_n\|), \quad (3)$$

where  $\|\cdot\|$  is the Euclidean distance, and the coefficients  $\mathbf{c}_n \in \mathbb{R}^Q$  are computed so as to correctly reproduce the imposed displacements  $\mathbf{d}(\mathbf{x}_n) = \boldsymbol{\delta}_n \in \mathbb{R}^Q, \forall n \in [1, N]$  at the source nodes; this requires the numerical solution of an  $N \times N$  linear system. Since the proposed method requires the solution of small linear systems, computational cost is not an issue even in multi-step deformations.

The behavior of the interpolation is profoundly influenced by the chosen RBF activation function  $\phi$  [17], with either local or global support. Activation functions adopted in this paper are reported in Table 1; some of them depend on  $\sigma$ , which also affects the interpolation [17]. In locally supported RBF activation functions,  $\sigma$  determines the region of influence of the RBF around each source node; this implies that  $\phi(r) \neq 0$  if and only if  $r < \sigma$ . The displacement of the handles dictated by the optimization is interpolated at the NURBS control points using the RBF method, with the Wendland C2 function if the user wishes to apply local deformations or the inverse multiquadric function for global ones. These functions are bounded ( $|\phi(r)| \leq 1, \forall r \geq 0$ ) and strictly monotonically decreasing ( $\phi(r_1) > \phi(r_2), \forall r_1 < r_2, r_1 \geq 0$ ); that is, the displacements interpolated at the NURBS control points fade away far from the handles, providing smooth and intuitive deformations (Section 3.1). Nodal displacements in the 2D parametric

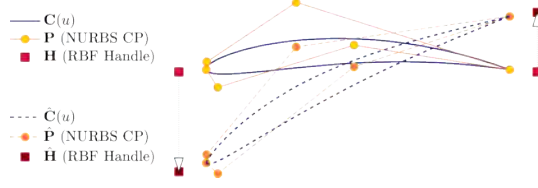


Figure 3: Deformation of two NURBS curves describing a 2D airfoil shape. The NURBS curves alter by interpolating the displacements  $\delta$  of the RBF handles  $H$  to the control points  $P$ .

space of NURBS surfaces are computed using the  $\sigma$ -free thin plate spline function, which has excellent interpolation capabilities in spaces where deformations are known along the boundaries and interpolated at the internal nodes (Section 4). Local deformations of surface grid nodes (Section 4) and NURBS control points (Section 3.2) are carried out with the Wendland C0 function. In RBF interpolations with global support, equation 3 is modified by adding polynomial terms to guarantee the exact affine displacement by respecting translation, rotation and scaling. These terms are not included in this paper; the reader may find more about them in [18].

### 3 RBF-Based CAD Shape-Morphing Framework

In the proposed method, shapes described by or transformed into NURBS patches undergo RBF-based morphing, see Section 3.1. By doing so, geometric continuities at the interfaces of NURBS patches can be jeopardized. Section 3.2 proposes an approach for preserving positional ( $G0$ ) and tangential ( $G1$ ) continuity between surfaces. Hereafter, surfaces and curves of a B-Rep model are not indexed.

#### 3.1 B-Rep Deformation Driven by Handles

During the optimization, shapes are modified by imposing displacement vectors  $\delta_n$ ,  $n \in [1, N]$  on a user-defined number of handles  $H_n$ . These displacements are interpolated at the NURBS control points  $P_i$ ,  $i \in [0, I]$  and  $P_{i,j}$ ,  $(i, j) \in [I \times J]$  of curves  $C_{cs}(u)$  and surfaces  $S(u, v)$ . Changed NURBS curves  $\hat{C}_{cs}(u)$  and surfaces  $\hat{S}(u, v)$  are created by means of the same rational basis functions  $R_i(u)$  and  $R_{i,j}(u, v)$  as the reference ones and displaced control points  $\hat{P}_i$  and  $\hat{P}_{i,j}$ . The so-morphed NURBS curves and surfaces are computed as

$$\hat{C}_{cs}(u) = \sum_{i=0}^I R_i(u) \underbrace{(P_i + d_M(P_i))}_{\hat{P}_i} \quad (4)$$

and

$$\hat{S}(u, v) = \sum_{i=0}^I \sum_{j=0}^J R_{i,j}(u, v) \underbrace{(P_{i,j} + d_M(P_{i,j}))}_{\hat{P}_{i,j}}, \quad (5)$$

where  $d_M(\cdot)$  is an RBF deformation operator (equation 3) trained on the displacements  $\delta_n$  of  $H_n$ ,  $n \in [1, N]$ . A 2D example is illustrated in Figure 3: two handles are used to deform an airfoil shape described by two NURBS curves, separately for its pressure and suction sides, which together form the B-Rep model.

In shape morphing, a frequent requirement is to keep certain parts of the overall shape fixed. Two possible ways to handle this requirement are proposed. In the first, although the entire geometry is considered deformable, fixed handles coinciding with the NURBS control points, describing the portion of the shape that should remain intact, are inserted. This option must be used if a smooth transition from deformable to fixed parts is required. For example, this approach is used in Section 5.2 to keep the inlet and outlet sections of a duct fixed while deforming its central part, with a smooth transition in between. The other option is to deform only a subset of the NURBS surfaces in the B-rep model. After the deformation action, the p-curves at the intersection of deformed and undeformed surfaces are updated by re-trimming. This allows the displacement of the intersection lines over the fixed surfaces while maintaining the validity of the CAD model. This approach is used in Section 5.3 in order to redesign

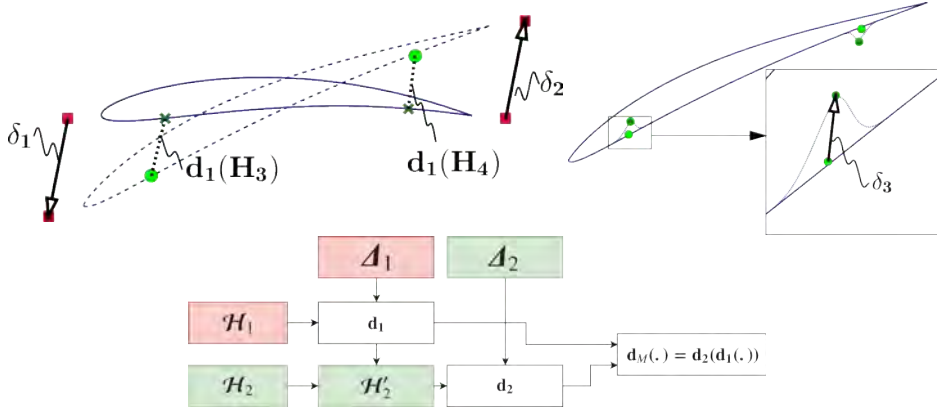


Figure 4: Two-step deformation of an airfoil shape.  $\Delta_1 = [\delta_1, \delta_2]^T$  are the displacements of the set of handles  $\mathcal{H}_1 = [\mathbf{H}_1, \mathbf{H}_2]^T$  which are active in the first step (red).  $\Delta_2 = [\delta_3, \delta_4]^T$  are the displacements of the set of handles  $\mathcal{H}_2 = [\mathbf{H}_3, \mathbf{H}_4]^T$  which are active in the second step (green). Top-left: The first step, in which active handles are marked in red and passive in green. New active handles are used to train the deformation function  $d_1 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , which is used to displace the passive handles from positions  $\mathcal{H}_2$  to  $\mathcal{H}'_2$  together with the shape. Top-right: The second step, in which green handles become active. Active handles are used to train the deformation function  $d_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  to further displace the shape deformed by  $d_1$ . Bottom: Schematics for the construction of the two-step deformation function  $d_M : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ .

an aircraft wing while keeping its fuselage fixed and, in Section 5.4, to keep the same axisymmetric hub and shroud surfaces of a stationary blade-row fixed while deforming the blade itself. Updated p-curves in the deformed B-Rep model are denoted by  $\hat{\mathcal{C}}_{pc}$ .

The RBF interpolation offers flexibility in shape modifications, allowing handles to be placed in arbitrary locations. For even higher flexibility, multi-step deformations<sup>1</sup> can be used: these are performed one after the other to incrementally deform surfaces. In each step, different sets of handles are used. Using more than one steps makes it possible to define more complex shape modifications, defining multiple sets of handles with increasing levels of control. Practically, the use of a few handles producing low-frequency deformations and many handles making high-frequency ones is recommended. In most cases, this gives rise to a two-step deformation approach. However, the proposed method and the programmed software support any number of steps. By solving the RBF problems (one for each step) sequentially, it is possible to employ various activation functions and  $\sigma$  values. It is recommended that handles responsible for low-frequency deformations be placed at a distance from the shape; just a few of them are usually enough. The activation function associated with these handles should be of global support. In contrast, handles responsible for high-frequency deformations should be placed on or next to the shape surface, in areas where higher deformation resolution is needed. The corresponding activation functions should be of local support. Depending on the application, users can choose whether handles move along Cartesian or cylindrical directions.

One of the metrics that can be used to measure the performance of the proposed method is the parametric effectiveness (PE) [19]. PE computes the degree to which the proposed method alters the search space compared to an optimization controlling NURBS control point positions directly. PE is the ratio of the maximum objective function improvement that can be achieved using the parameterization in hands to the one that could be achieved if surface nodal points were free to move independently; both are subjected to the constraint of the same root-mean-squared boundary displacement. PE can be computed at the cost of a single adjoint run [20]. Herein, we refrain from presenting the definition of PE; this can be found in [19] and [20].

Unlike single-step RBF-based deformation, in which all handles interact with each other, the use of various sets of handles allows successive modifications over the same part of the shape. Each step allows for deforming the shape with a different geometric modeling paradigm. However, to maintain the modeling paradigm imposed by the user at each step, sets of handles must be displaced hierarchically. In each step, a distinction is made between active and passive sets of handles. The former is a single set used

<sup>1</sup>“Multi-step deformations” denotes a sequence of distinct deformations and not the division of a single deformation into many steps.

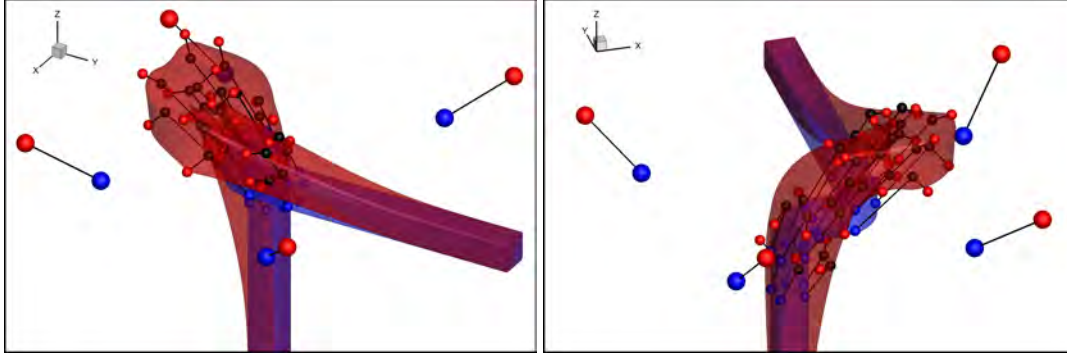


Figure 5: Double elbow duct. Reference (blue) and optimized (red) geometry targeting total pressure losses. The shape is altered by displacing the handles (spheres) using a two-step approach. The first step is based on 4 handles (large spheres), whereas the second on 30 handles (smaller spheres). Handles are shown in their initial (blue) and final (red) positions. Moreover, the intermediate (black) positions of the handles of the second step are shown. These are computed after applying the deformation action of the first step. Inlet and outlet sections are kept intact by fixed handles (not shown) coinciding with the NURBS control points.

to train the RBF deformation function of the current step. The latter includes all sets that will become active in the subsequent steps; their position is passively displaced by the RBF deformation functions of all previous steps. More precisely, to perform an  $S$ -step deformation, handles must be organized in  $S$  sets  $\mathcal{H}_s, s \in [1, S]$ :

$$\begin{aligned} \mathcal{H}_1 &= [\mathbf{H}_1, \dots, \mathbf{H}_{n_1}]^T, \quad \mathcal{H}_2 = [\mathbf{H}_{n_1+1}, \dots, \mathbf{H}_{n_2}]^T, \quad \dots, \\ \mathcal{H}_S &= [\mathbf{H}_{n_{S-1}+1}, \dots, \mathbf{H}_N]^T, \quad 1 \leq n_1 < \dots < n_{S-1} < n_S \equiv N. \end{aligned}$$

For an  $S$ -step deformation, the operator  $\mathbf{d}_M$  in equations 4 and 5 includes the deformation functions of all steps, namely  $\mathbf{d}_M(\mathbf{x}) = (\mathbf{d}_S \circ \mathbf{d}_{S-1} \circ \dots \circ \mathbf{d}_1)(\mathbf{x})$  for  $S$  steps, where the  $s^{\text{th}}$  deformation function is trained on the set

$$\mathcal{H}'_s = \begin{bmatrix} \mathbf{H}_{n_{s-1}+1} + (\mathbf{d}_{s-1} \circ \dots \circ \mathbf{d}_1 \circ \mathbf{d}_0)(\mathbf{H}_{n_{s-1}+1}) \\ \vdots \\ \mathbf{H}_{n_s-1} + (\mathbf{d}_{s-1} \circ \dots \circ \mathbf{d}_1 \circ \mathbf{d}_0)(\mathbf{H}_{n_s-1}) \\ \mathbf{H}_{n_s} + (\mathbf{d}_{s-1} \circ \dots \circ \mathbf{d}_1 \circ \mathbf{d}_0)(\mathbf{H}_{n_s}) \end{bmatrix},$$

for the set of displacements  $\mathcal{A}_s = \{\delta_n, n \in [n_{s-1} + 1, n_s]\}$ ;  $\mathbf{d}_0(\mathbf{x})$  is the null deformation function ( $\mathbf{x} = \mathbf{d}_0(\mathbf{x})$ ).

This process is better explained using a pair of examples. In Figure 4, the first deformation step makes low-frequency shape modifications and also displaces the position of the handles of the second step. The second step makes high-frequency shape modifications. In the second step, handles placed close to the surface remain close to it even after the first-step deformation, maintaining the same modeling paradigm prescribed on the reference shape. Figure 4 also illustrates the composition of a two-step deformation function. The second example, illustrated in Figure 5, is concerned with the double elbow duct further analyzed in Section 5.2. The duct shape is altered by displacing two sets of handles corresponding to two deformation steps. In the first step, the four active handles deform the duct shape and displace the passive handles of the second step (from the "blue" to the "black" position). In the second step, the second set of handles becomes active and additionally deforms the shape through its displacement.

In the 2D airfoil case of Figure 3, after the deformation,  $C0$  continuity at the leading and trailing edges is guaranteed, because the first and last control points of the two NURBS curves coincide. In fact, if two control points  $\mathbf{P}_\alpha$  and  $\mathbf{P}_\beta$  coincide before the transformation ( $\mathbf{P}_\alpha \equiv \mathbf{P}_\beta$ ), this will be the case for the deformed shape too, because  $\mathbf{d}_M(\mathbf{P}_\alpha) \equiv \mathbf{d}_M(\mathbf{P}_\beta)$ . However, in 3D CAD models, this is generally not the case, and positional and tangential ( $G0$  and  $G1$ ) continuity between surfaces must be recovered after the RBF-based deformation by repositioning some NURBS surface control points. For example, any shape-morphing action applied to a CAD model, such as the DPW6 wing-body configuration, Figure 6, would break the  $G0$  continuity of some surfaces, rendering the CAD model invalid. The DPW6 is



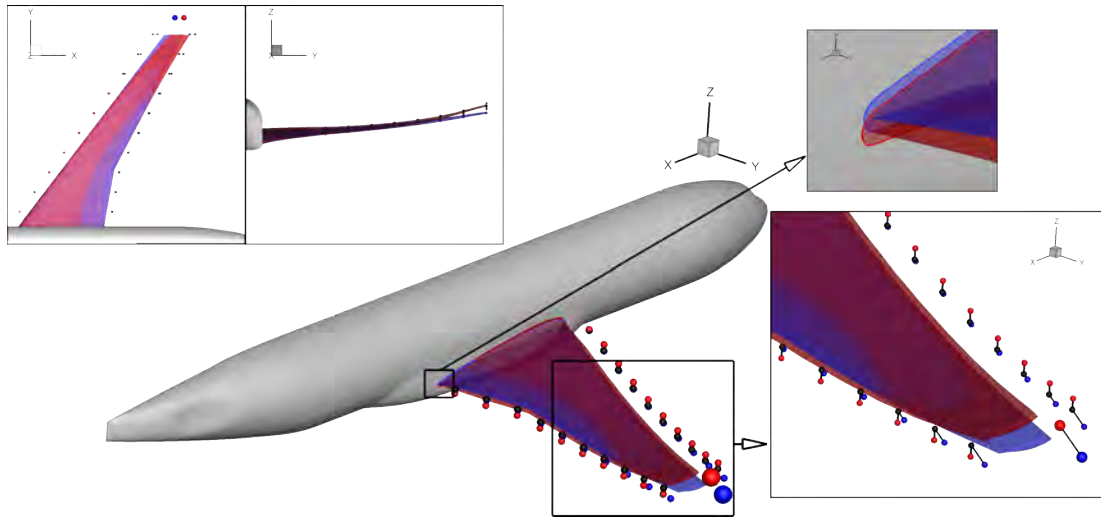


Figure 6: DPW6 aircraft. Shape deformation. The fuselage shape remains intact while the wing is deformed. The initial shape of the wing and the initial handle positions are shown in blue, the deformed wing and the final handle positions in red. The deformation action is carried out in two steps; the black spheres show the positions of the handles at the beginning of the second step.

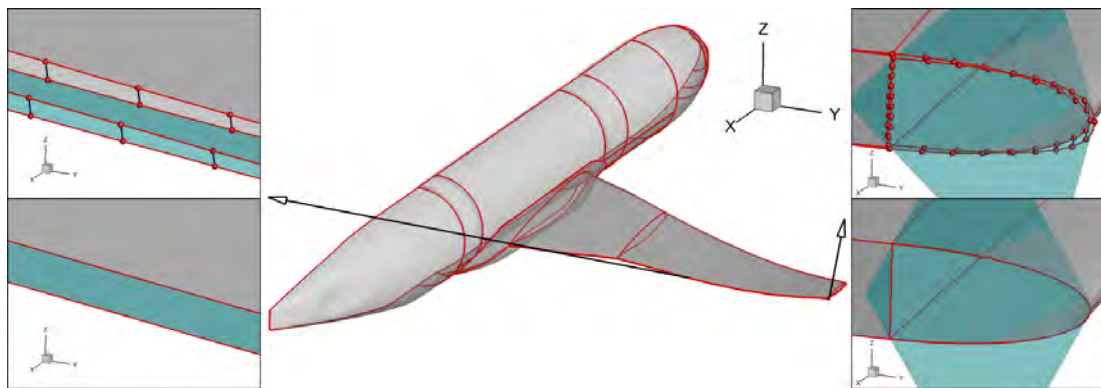


Figure 7: DPW6 aircraft.  $G_0$  continuity fixing. The RBF-based displacements of the NURBS control points create gaps at joints, violating  $G_0$  continuity. The surfaces are sewed by minimizing the squared distances between test points, here illustrated as red spheres placed along the edges of the CAD model. Top: Close-up view of a narrow surface at the wing trailing edge, top-left, and a small surface at the wing tip, top-right, prior to the fix. Bottom: Same close-up views after the fix.

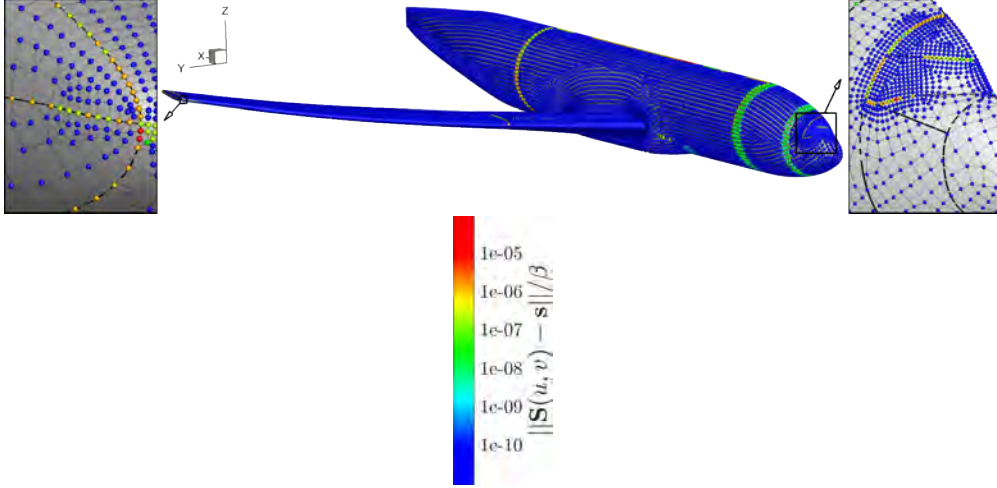


Figure 8: DPW6 aircraft. Results of the projection of a CFD surface grid onto the CAD model. Residual values  $\|S(u, v) - s\|$  are normalized w.r.t. the length of the diagonal of the surface grid bounding box  $\beta$ . The edges of the CAD model are also shown as black dashed curves.

an aerodynamic model representative of a modern transonic commercial aircraft with available CAD model [21]. The wing geometry is defined by 9 NURBS patches, containing 34,765 control points overall, whereas half of the fuselage is formed by 23 patches, containing 134,128 control points overall. Figure 6 illustrates the shape-morphing action based on a two-step deformation. In the first step, one handle, which is allowed to move in the  $xz$ -plane, controls the overall shape of the wing. In the second step, handles placed close to the leading and trailing edges control the wing locally through their displacements in the  $z$ -direction; displacements in the  $x$ -direction are not allowed in order to keep the axial chord of the wing constant. The fuselage shape is excluded from the morphing actions and remains intact. The deformation action smoothly changes the shape of the wing but creates gaps, such as those at the narrow trailing edge surface or over the small surface at the tip of the wing, which renders the deformed CAD model temporarily invalid, Figure 7. In Section 3.2, a procedure for repairing this kind of discontinuity is proposed.

### 3.2 Geometric Continuity Correction

As mentioned above, during the RBF deformation, NURBS control points are displaced in a way that generally violates continuity between neighboring NURBS patches in the B-rep model, which needs to be corrected.  $G0$  continuity is ensured by "sewing" the p-curves  $\hat{C}_{pc}(u)$  of the deformed surfaces to the corresponding deformed curves-on-surface  $\hat{C}_{cs}(u)$  by displacing the surfaces. This approach requires an evaluation of the  $G0$  continuity at test points distributed on the curves-on-surface. Figure 2 shows a single test point on a curve-on-surface and the corresponding p-curves of adjacent surfaces; Figure 7 illustrates some test points in the Cartesian space for the DPW6 B-rep model. These test points  $u_{cs,t}$ ,  $t \in [1, N_T]$  are distributed in the parametric space of the curves-on-surface  $C_{cs}$  of the reference B-Rep model, and the corresponding parameters  $u_{pc,t}$  of each p-curve  $C_{pc}$  in the corresponding surface  $S$  are computed by

$$\min_{u_{pc,t}} \left\| S \left( \underbrace{u_t, v_t}_{C_{pc}(u_{pc,t})} \right) - C_{cs}(u_{cs,t}) \right\|.$$

Multiple ways to compute (by equations 1 and 2) the same 3D point on the surface of the model, up to a tolerance, are established; in fact, for all test points  $C_{cs}(u_{cs,t}) \approx S(C_{pc}(u_{pc,t}))$ . After the B-Rep model is deformed,  $\hat{C}_{cs}(u_{cs,t})$  and  $\hat{S}(\hat{C}_{pc}(u_{pc,t}))$  no longer coincide. In order to fix it, a minimization problem is set up to reduce the distance between the points computed through the deformed curves-on-surface  $\hat{C}_{cs}(u_{cs,t})$  and p-curves  $\hat{S}(\hat{C}_{pc}(u_{pc,t}))$ . Such a minimization problem uses a subset of the control points  $\hat{P}_\Gamma := \{\forall \hat{P}_{i,j} : (i, j) \in \Gamma\}$  in each surface  $\hat{S}$  as design variables. The corresponding minimization problem

is defined for each surface as

$$\min_{\Delta \hat{\mathbf{P}}_{\Gamma}} C_{G0}(\Delta \hat{\mathbf{P}}_{\Gamma}) = \min_{\Delta \hat{\mathbf{P}}_{\Gamma}} \sum_{t=1}^{N_T} \left\| \hat{\mathbf{S}}(\hat{\mathbf{C}}_{pc}(u_{pc,t})) - \hat{\mathbf{C}}_{cs}(u_{cs,t}) + \sum_{(i,j) \in \Gamma} R_{i,j}(\hat{\mathbf{C}}_{pc}(u_{pc,t})) \Delta \hat{\mathbf{P}}_{i,j} \right\|^2, \quad (6)$$

where  $\hat{\mathbf{S}}(\hat{\mathbf{C}}_{pc}(u_{pc,t}))$  is the point computed on the deformed surface,  $\hat{\mathbf{C}}_{cs}(u_{cs,t})$  is the point computed on the deformed curve and  $\Delta \hat{\mathbf{P}}_{\Gamma} := [\forall \Delta \hat{\mathbf{P}}_{i,j} : (i, j) \in \Gamma]$  are the corrections to the surface control points identified by  $\Gamma$ .  $\Gamma$  is the set of indices defined as

$$\Gamma = \bigcup_{t=1}^{N_T} \Gamma_t, \Gamma_t = \{(i, j) \in [I \times J] : \frac{R_{i,j}(u_t, v_t)}{\sum_{l=0}^I \sum_{m=0}^J R_{l,m}(u_t, v_t)} \geq \bar{R}_t\},$$

where  $\bar{R}_t$  is a threshold defined by averaging all non-zero  $R_{i,j}(u_t, v_t)$ . It is necessary to modify only the control points identified by  $\Gamma$ , instead of the full set  $(i, j) \in [I \times J]$ , in order to well pose the minimization problem of equation 6. Due to the local support property of NURBS, many control points might play no role in minimizing the cost function  $C_{G0}(\Delta \hat{\mathbf{P}}_{\Gamma})$ . To avoid singularities in the Hessian matrix derived from  $C_{G0}$ , these control points must not be included in the minimization problem. Excluding control points that have little influence also improves robustness and convergence rate. Test points are automatically arranged on the curves-on-surface based on distance, curvature and knot-distributions criteria. Since NURBS are polynomial expressions, testing the  $G$ -continuities of two NURBS entities at a finite number of points is sufficient to guarantee continuity along the entire curve-on-surface. The proposed method based on the minimization of  $C_{G0}(\Delta \hat{\mathbf{P}}_{\Gamma})$  overcomes over-sampling, with the only extra burden being the increased computational cost for computing  $C_{G0}$ ; however, the number of degrees of freedom ( $\Delta \hat{\mathbf{P}}_{\Gamma}$ ) of the cost function is independent of the number of test points.

When computing the new surface according to the corrections  $\Delta \hat{\mathbf{P}}_{\Gamma}$ , since these corrections are applied only to the control points identified by  $\Gamma$  in each NURBS surface, the resulting shape might be irregular; that is, some rows of control points might move while others remain unaffected. An additional procedure is, therefore, applied to regularize the deformation induced by the corrections. The displacements  $\Delta \hat{\mathbf{P}}_{\Gamma} := [\forall \Delta \hat{\mathbf{P}}_{i,j} : (i, j) \in \Gamma]$  computed by minimizing  $C_{G0}(\Delta \hat{\mathbf{P}}_{\Gamma})$  are propagated to the rest of the control points by the RBF interpolation. The locally supported RBF activation function used in this phase is the Wendland C0 (Table 1) with  $\sigma$  chosen to be four times the maximum displacement norm,  $\sigma = 4 \max_{(i,j) \in \Gamma} \|\Delta \hat{\mathbf{P}}_{i,j}\|$ , so as to smoothly propagate the deformation while retaining a local effect. The deformed  $G0$ -continuous surfaces are given by

$$\hat{\mathbf{S}}(u, v) = \sum_{i=0}^I \sum_{j=0}^J R_{i,j}(u, v) (\hat{\mathbf{P}}_i + \mathbf{d}_C(\hat{\mathbf{P}}_{i,j})),$$

where  $\mathbf{d}_C$  is the interpolation operator ( $C$  stands for correction). The minimization problem of equation 6 is solved using the Newton method. Since the Hessian matrix is constant (i.e., it does not depend on the NURBS control point positions), and many of its entries are zero due to the local support property of the NURBS, the Hessian matrix for each NURBS surface is inverted once with a sparse decomposition [22]. Similarly, the computationally expensive rational basis functions that appear in the objective function and its first derivative are computed just once and re-used, since they do not change during the minimization of  $C_{G0}$ . In fact, by pre-computing these quantities, it is possible to reduce the time needed to perform a Newton iteration by as much as two orders of magnitude. Moreover, these minimization problems can be solved in parallel, being independent of each other. In some B-Rep models, the distribution of control points does not allow the optimization to drive the  $C_{G0}$  objective sufficiently low to make the CAD model valid. In fact, a curve that contains too few control points cannot be sewed to another curve effectively due to the limitation in shape flexibility. In this case, additional knots are automatically inserted into the reference B-rep model.

$G1$  continuity is corrected by solving a constrained minimization problem: on condition that  $G0$  continuity is satisfied,  $G1$  continuity requires that adjacent NURBS surfaces share the same tangent plane along the shared curve-on-surface. For two adjacent surfaces  $\mathcal{S}_1$  and  $\mathcal{S}_2$  that are  $G0$  continuous at the

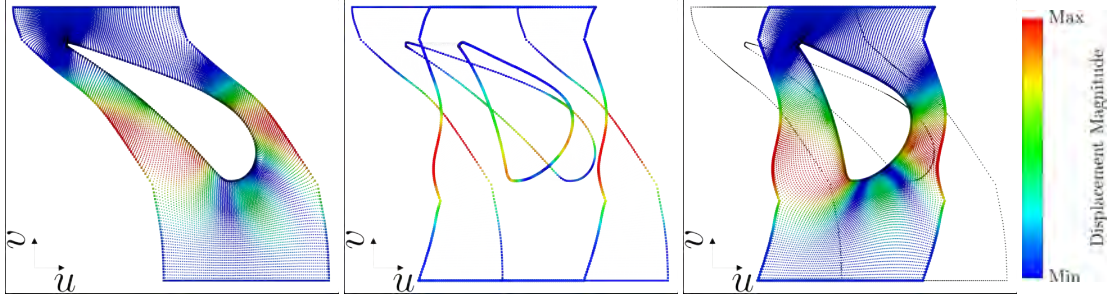


Figure 9: Displacement of the hub surface grid of a turbine stator in the NURBS surface parametric space. Left: Initial positions of the nodes in the  $u - v$  plane. Middle: Displacements plotted along the p-curves of the reference and morphed shape. Right: Nodes displaced by the RBF interpolation. Colors represent the magnitude of the nodal displacements.

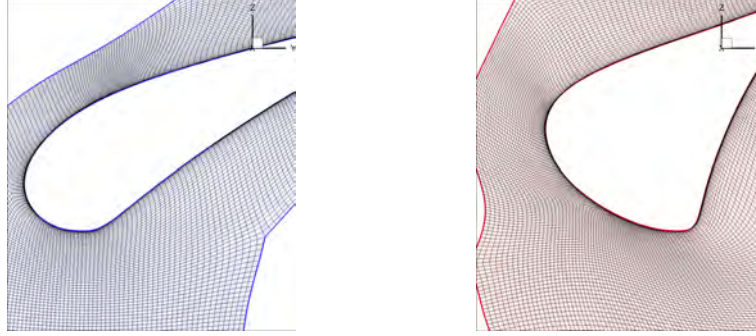


Figure 10: Reference and a morphed Cartesian grid of a turbine stator. Left: The reference surface grid projected onto the  $(u, v)$  space of the hub surface. Right: The morphed grid computed with the morphed  $(\hat{u}, \hat{v})$  parametric nodes.

points  $(u_t, v_t)_{S_1}$  on  $S_1$  and  $(u_t, v_t)_{S_2}$  on  $S_2$  ( $S_1((u_t, v_t)_{S_1}) \cong S_2((u_t, v_t)_{S_2})$ ), the  $G1$  continuity condition at such test points is expressed as

$$\det \left( \frac{\partial S_1(u, v)}{\partial u} \Big|_{(u_t, v_t)_{S_1}}, \frac{\partial S_1(u, v)}{\partial v} \Big|_{(u_t, v_t)_{S_1}}, \frac{\partial S_2(u, v)}{\partial u} \Big|_{(u_t, v_t)_{S_2}} \right) = 0, \quad (7)$$

which is a measure of the coplanarity of the tangential vectors computed at the test points on the two surfaces.

From equation 7, a differentiable cost function can easily be derived as the summation of the determinants of all test points shared by two or more surfaces and minimized by displacing the surface NURBS control points similar to the  $G0$  correction case. Whereas  $G0$  continuity correction requires solving an independent unconstrained problem for each surface,  $G1$  continuity requires solving a constrained optimization problem involving all surfaces; the cost function for the latter involves the displacement of the control points of all deformable surfaces of the B-Rep model. However, the RBF deformation function is implicitly smooth [23], and in many cases,  $G1$  continuity does not need to be fixed. The Sequential Quadratic Programming (SQP) method is used to handle the gradient-based constrained optimization with many design variables. The corrective deformation imposing  $G1$  continuity is interpolated with the RBF method at all NURBS control points.

## 4 CFD Surface Grid Displacement

Deforming an existing 3D CFD grid according to a morphed CAD model is fundamental to reduce the computational cost of aerodynamic shape optimizations. The proposed CFD surface grid displacement method exploits the fact that, in the shape-morphing strategy presented in Section 3, the CAD model

topology does not alter during the deformations; thus, the NURBS-based representation of shapes enables the mapping of the CFD grid surface nodes onto the  $(u, v)$  NURBS surface parametric spaces. Such a mapping works by determining which NURBS surface a grid node  $s \in \mathbb{R}^3$  belongs to and identifying the parameters  $(u, v)$  that yields  $s$  by taking into account the domain  $\Omega$  of each NURBS surface (part of the surface that is not trimmed). The projection of each surface grid node is checked separately on each surface. That is, for each surface, the  $(u, v)$  parameters of each node are found by solving the problem

$$\min_{(u,v) \in \Omega} \|S(u, v) - s\| . \quad (8)$$

Surface grid nodes at joints belong to more than one surfaces, and their parametric values are computed for all of them. Because the problem of equation 8 is overdetermined (three equations and two unknowns), it is solved by nonlinear least-squares. Care must be taken in case the NURBS patch is not  $C1$  continuous and in order to provide a good starting point for the least-squares iterations. More details on the point projection algorithm can be found in [16, Section 6.1]. Because the projection task is expensive, even if fully parallelized, due to a large number of CFD surface grid nodes to be projected, the  $(u, v)$  values are computed once and, then, stored. Figure 8 illustrates the residuals of the projection of a CFD surface grid on the B-rep model of the DPW6 test case. Grid nodes lying close to or along the boundary of the parametric domains  $\Omega$  are subject to higher residual values from the minimization of the problem in equation 8 due to CAD model tolerances involved in the definition of the domain  $\Omega$  itself.

When deforming NURBS surfaces, their parametric domains  $\Omega$  might change due to changes in the p-curves  $C_{pc}$ , as described in Section 3.1. Before computing the surface grid, the grid projected onto the parametric domain  $\Omega$  needs to be adapted to the new parametric domain  $\hat{\Omega}$  defined by the new p-curves  $\hat{C}_{pc}$ . The parametric domain is adapted using RBF-based interpolation. As noted, there is a one-to-one correspondence between surfaces, curves and p-curves of the reference and deformed CAD models. For each parametric domain, nodes are distributed equidistantly along the p-curves (distances are measured in the parametric space of the surface) obtained from the reference and deformed CAD model: the displacement of each node  $(u, v) \in \Omega$ , computed from the p-curves, is used to find the RBF deformation function  $d_{PAR} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ . The thin plate spline activation function (Table 1) is used. The new parametric values contained in  $\hat{\Omega}$  are then computed as  $(\hat{u}, \hat{v}) = (u, v) + d_{PAR}(u, v)$  and used to find the new CFD surface grid nodes  $\hat{s} = \hat{S}(\hat{u}, \hat{v})$ . For nodes along the joints, the average position is taken and the surrounding nodes are adjusted by local RBF interpolation based on the Wendland C0 function; this ensures the validity of the CFD surface grid in case computational nodes are close to each other according to the user-defined geometric tolerance associated with the B-Rep model.

Figures 9 and 10 illustrate the procedure for displacing the surface grid in the NURBS parametric space for the hub surface of a turbine stator. The reference grid in Figure 10 is projected onto and displaced into the NURBS parametric space of the hub surface, Figure 9. The displaced surface grid in Cartesian space is illustrated in Figure 10.

## 5 Method Demonstration

This section demonstrates the application of the proposed method and the programmed software in three shape optimization case studies. First, the method is used in a 2D case, namely to re-parameterize an airfoil, by also computing its effect on the search space through the computation of the PE. Then, the method contributes to the gradient-based optimization of a double elbow duct and a multi-objective evolutionary algorithm (EA)-based optimization of an aircraft wing. Finally, the results of an EA-based optimization of a compressor stator by using CAD software for turbomachinery blade modeling and the proposed method are presented and compared. All CFD simulations are performed using the in-house GPU-enabled Reynolds-Averaged Navier-Stokes (RANS) equations solver [24] employing the Spalart-Allmaras turbulence model. The volume grids are displaced with the in-house two-step RBF-based grid displacement software [25], which is able to displace grids with millions of nodes in a couple of minutes (on a high-performance computing node), given the displacement of their surface nodes. EA-based optimizations are performed using the in-house software EASY (Evolutionary Algorithm System) [26], whereas gradient-based optimizations rely upon the SQP method [2]. Although the proposed method and software support an arbitrary number of deformation steps, all applications presented herein make use of a two-step deformation approach.

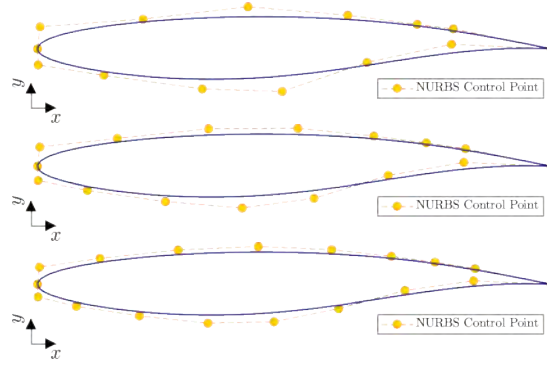


Figure 11: RAE 2822 airfoil. The NURBS-based parameterizations employing the NURBS control points ordinates as DoFs are illustrated. During the optimization, the control points corresponding to the leading and trailing edges are kept fixed. From top to bottom the airfoil is parameterized with 12, 14 and 16 DoFs.

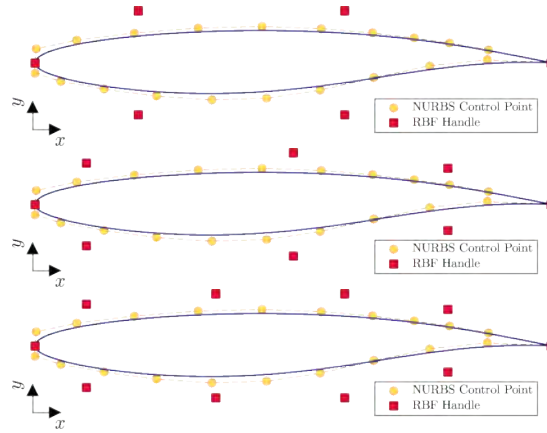


Figure 12: RAE 2822 airfoil. The RBF-based parameterizations employing the RBF handles displacements in the  $y$ -direction as DoFs are illustrated. The RBF handles displacements control the position of the NURBS control points. During the optimization, the handles corresponding to the leading and trailing edges are kept fixed. From top to bottom the airfoil is parameterized with 4, 6 and 8 DoFs.

## 5.1 A 2D Study: RAE 2822 Airfoil

Several studies on the RAE 2822 airfoil can be found [27]. The purpose here is to assess the quality of shapes computed by employing the proposed parameterization method compared to the outcome of an optimization with the NURBS control point positions as Degrees of Freedom (DoFs). To this end, the PE of the two methods is also computed. In detail, the two parameterization methods to be compared, with different numbers of DoFs each, are:

- Three airfoil parameterizations based on NURBS curves with different numbers of control points each. NURBS curves for the pressure and suction sides with 8, 9 or 10 points are used. During the optimization, only 6, 7 or 8 control points per curve, respectively, are allowed to move in the  $y$ -direction yielding 12, 14 or 16 DoFs in total. The parameterizations are illustrated in Figure 11.
- Three parameterizations based on RBF handles which control the NURBS curves. The pressure and suction sides are approximated with NURBS curves with 10 control points each; their positions are controlled by 6, 8 or 10 handles placed around the airfoil, Figure 12. During the optimization, 4, 6 or 8 handles, respectively, move in the  $y$ -direction; thus, the optimization is conducted with 4, 6, or 8 DoFs. The parameterizations are illustrated in Figure 12. The inverse multiquadric activation function with global support is used.

The free-stream flow conditions correspond to test case 6 of [28] with Mach number equal to 0.725, flow angle  $2.92^\circ$  and Reynolds number  $6.5 \cdot 10^6$ . The function to be minimized is the drag coefficient

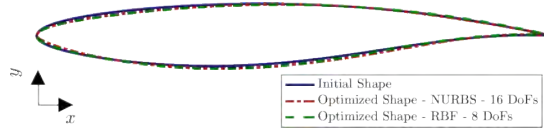


Figure 13: RAE 2822 airfoil. Original shape compared to the ones with improved drag obtained from different shape optimizations employing the NURBS control point positions and RBF handles displacements as DoFs. Differences in the shapes are minor.

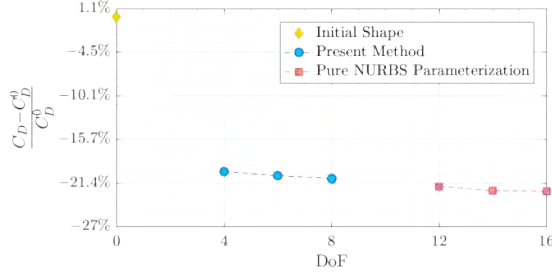


Figure 14: RAE 2822 airfoil.  $C_D$  improvement obtained by optimizing the shape of the airfoil with different numbers of DoFs and two different parameterization methods. Constraints are imposed on  $C_L$ ,  $C_m$  and  $A$ .  $C_D^0$  is the drag coefficient of to the initial shape.

$C_D$ . Inequality constraints are imposed on the lift coefficient  $C_L \geq C_L^0$ , the pitching moment coefficient  $C_m \geq C_m^0$  and the airfoil area  $A \geq A^0$ .  $C_L^0$ ,  $C_m^0$  and  $A^0$  are the values associated with the initial shape. Leading and trailing edge are kept fixed, which results in constant axial chord length. The gradient-based optimization is conducted using an SQP-based algorithm by also bounding the DoFs.

The PE of the NURBS-based parameterization with 16 DoFs is equal to 0.65 for the baseline shape. The use of the proposed method with 8 DoFs yields a PE value of 0.55. Even if the parameterization based on handles reduces the number of DoFs by 50%, the PE is, however, reduced by just 16%.

Figure 14 illustrates the  $C_D$  that the two parameterization methods with different numbers of DoFs achieve. As expected, by increasing the DoFs lower  $C_D$  values are found. However, the reduction in the number of DoFs by the proposed method implies just a small reduction of the capabilities of the optimization. The proposed parameterization finds designs that reduce  $C_D$  up to  $\sim 20\%$ . Comparatively, the pure NURBS parameterization finds designs that reduce  $C_D$  up to  $\sim 22\%$  but with significantly more DoFs. To conclude, the parameterizations based on NURBS curves controlled by RBF handles can reach an almost similar  $C_D$  reduction and shape with a reduced number of DoFs, compared to that based purely on NURBS, for the case under consideration. Figure 13 illustrates some of the improved shapes compared to the initial one: differences are small.

## 5.2 Double Elbow Duct

The double elbow duct B-Rep model is composed of a single untrimmed NURBS surface with 2,128 control points. RBF deformation actions do not involve changes in the domain  $\Omega$ , because the surface is not trimmed, and parametric nodes  $(u, v)$  do not change. The reference shape and parameterization are illustrated in Figure 5. The shape-morphing action is based on a two-step deformation. In the first step, 4 handles placed off the surface control the entire shape, whereas the second step uses 30 handles on the surface, with local support. Each handle is allowed to move in all three Cartesian directions, thus contributing three DoFs to the optimization. Therefore, the total number of DoFs is 102. Fixed handles are placed at the inlet and outlet, in positions corresponding to the NURBS control points, to constrain the displacement of these sections of the duct. The adjoint-based shape optimization is conducted to reduce the mass-averaged total pressure losses  $\Delta p_t$  of the duct, defined as

$$\Delta p_t = \frac{- \int_{S_i} p_t \rho \mathbf{V} \cdot \mathbf{n} dS - \int_{S_o} p_t \rho \mathbf{V} \cdot \mathbf{n} dS}{\int_{S_i} (p_t - p) \rho \mathbf{V} \cdot \mathbf{n} dS}, \quad (9)$$

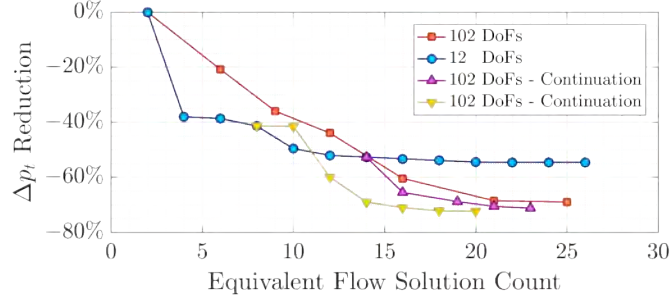


Figure 15: Double elbow duct. Convergence history in terms of equivalent flow solution counts (the gradient computation cost is about that of one flow solution) of four SQP-based optimizations using various numbers of DoFs. One is conducted by using 12 DoFs (first deformation step) and the other by using 102 DoFs (first and second steps). Two more convergence histories are obtained by suspending the optimization with 12 DoFs and re-starting it with 102 DoFs, by retaining the values of the first 12 DoFs; these are referred to as "Continuation".

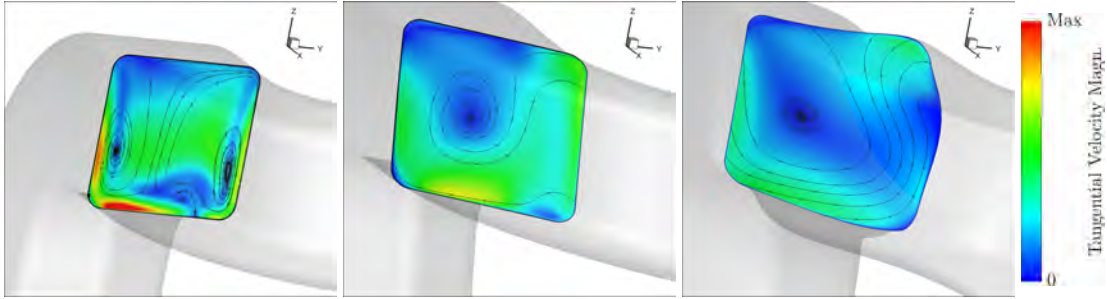


Figure 16: Double elbow duct. View of the secondary flow structure at a cross-section between the two elbows for the reference shape (left) and the ones optimized by employing 12 DoFs (middle) and 102 DoFs (right). Contours represent the normalized velocity magnitude over the cross-section plane.

where  $p$  is the pressure,  $p_t$  the total pressure,  $\rho$  the density,  $V$  the velocity,  $\mathbf{n}$  the outward unit vector normal to the surface and  $S_O$  and  $S_I$  the duct outlet and inlet, respectively. The fluid is incompressible, and the Reynolds number based on the hydraulic diameter of the inlet is equal to  $10^6$ . Figure 17 depicts geometric sensitivities for the displacement of a handle along the  $z$ -direction. Figure 15 compares the optimization convergence histories by using different numbers of DoFs for the parameterization. The convergence of the optimization conducted using a "one-shot" run with all 102 DoFs simultaneously is compared to that obtained by starting with a low-dimensional design space allowing low-frequency shape changes (first step) and, then, adding the DoFs which are responsible for high-frequency shape modifications (second step). It appears that a small number of DoFs offers a great gain in the optimization in the first cycles. In contrast, using all handles simultaneously from the beginning leads to more extended SQP line searches and higher cost. Specifically, the SQP-based algorithm is able to reduce the total pressure losses w.r.t. the starting shape by almost 40% using 12 DoFs, at half the cost of the first SQP cycle employing 102 DoFs, which enables a reduction of "just" 20%. However, after a few cycles, the latter can make better design improvements by exploiting the greater number of DoFs. The best convergence rate is obtained by using a sequential optimization strategy in which only a few DoFs are initially used for a few SQP cycles and the optimization continues with a larger number of DoFs. This strategy also results in a slightly better design.

Figure 5 illustrates the improved geometries computed by the proposed parameterization method and compares them with the reference geometry. Figure 16 presents the secondary flow between the two elbows for the reference shape and the optimal ones, resulting from optimizations conducted using 12 and 102 DoFs. The first deformation step changed the duct shape mainly by increasing its cross-sectional area, thus slowing down the flow, which reduces the wall shear stress and diffusion losses. The increased length between the two elbows and their enlarged radii suppress secondary flows and centripetal forces required to turn the flow, hence reducing diffusion losses and areas with flow separation. The second deformation step introduces more local control and performs local changes; the result is fewer secondary flow patterns and lower total pressure losses.



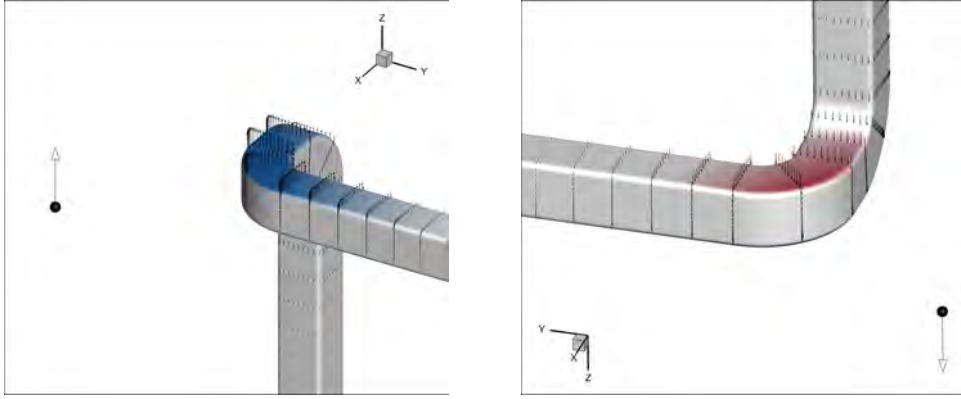


Figure 17: Double elbow duct. Geometric sensitivity w.r.t. a design variable, namely the displacement in the  $z$ -direction of the handle illustrated as a sphere. Colors quantify normal sensitivities. Blue indicates that the surface is pulled outwards the fluid domain by a displacement of the handle in the positive  $z$ -direction, whereas red that the surface is pushed in the opposite direction. Vectors illustrate 3D geometric sensitivities.

### 5.3 DPW2 Aircraft Model

In this section, aerodynamic shape optimization is conducted for the DPW2 wing-body configuration [29]. The DPW2 common research model is representative of a commercial aircraft. The wing geometry is defined by 13 NURBS patches, containing 3,638 control points overall; half of the fuselage is formed by 11 patches, containing 4,577 control points overall. Figure 19 illustrates the wing parameterization adopted, which is similar to the parameterization illustrated in Figure 6 for the DPW6 test case. That is, in the first step, one handle is allowed to move in the  $XZ$ -plane and controls the overall shape of the wing. In the second step, nodes placed close to the leading and trailing edges control the shape locally by moving in the  $Z$ -direction, whereas the fuselage shape is kept intact by excluding the corresponding surfaces from the morphing action. The intersection between fuselage and wing is updated by re-trimming. The two-objective shape optimization aims at minimum drag and maximum lift at flow conditions corresponding to a Mach number of 0.75, Reynolds number of  $3 \cdot 10^6$  and  $0^\circ$  infinite flow angle. Figure 18 illustrates the Pareto front for the two objectives, found at the cost of 250 design evaluations using EASY. In the Pareto front, a solution that increases the  $C_L/C_D$  ratio by 3.0%, improving both lift and drag, can be identified. Another design improves the  $C_L/C_D$  ratio by 4.3%; however, in this case  $C_L$  is improved at the expense of  $C_D$ . [30] was able to reduce  $C_D$  by 2.6% by keeping the  $C_L$  equal to that of the initial geometry by varying the angle of attack. The optimization employed a gradient-based method and a CAD-free lattice-based FFD parameterization for the wing and wing-body junction, with 96 DoFs overall. A similar improvement is achieved with the proposed parameterization, yielding  $\approx 2.75\%$  reduction in  $C_D$  while keeping  $C_L$  equal to that of the initial geometry, though with just 12 DoFs.

### 5.4 TUB Compressor Stator Optimization

The TurboLab Berlin compressor stator common research model has been tested in the measurement rig of the Chair for Aero Engines at TU Berlin. In the past, several research groups from academia and industry studied this case [31–33] and experiments were conducted on the original geometry and one optimized by Rolls-Royce Deutschland [34]. This stator is an aerodynamic model of a blade typically found in modern jet engine compressors. The geometry in CAD format, flow conditions and the list of constraints can be downloaded from [35]. This case is used to check whether the proposed parameterization finds competitive designs compared to a constructive CAD-based one, in the framework of a multi-objective optimization. Obviously, different parameterizations give rise to different design spaces. The objectives are to minimize the mass-averaged total pressure losses between the inlet and outlet of the CFD domain and the mass-averaged deviation of the exit flow from the axial direction. The mass-averaged total pressure losses are defined in equation 9; the mass-averaged deviation of the exit flow from the axial direction is

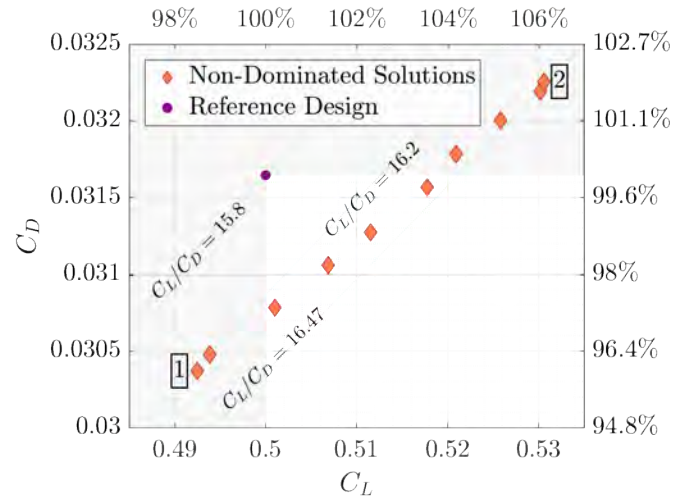


Figure 18: DPW2 aircraft. Results of the EA-based optimization for maximum lift and minimum drag. The wing shapes corresponding to the two Pareto points marked with integers are compared to the reference design shown in Figure 19.

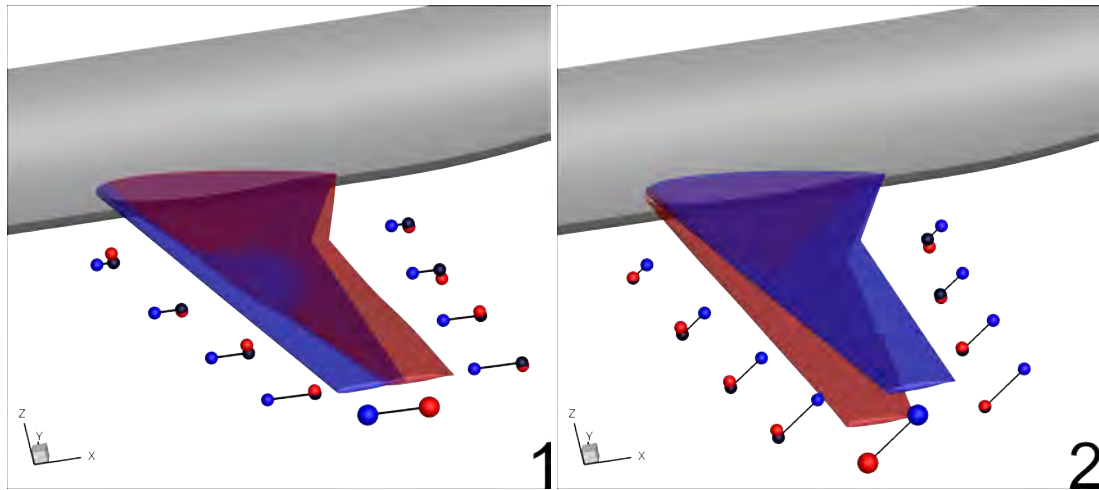


Figure 19: DPW2 aircraft. Two selected designs (red) from the front on non-dominated solutions in Figure 18 are shown in comparison to the reference design (blue), with the corresponding handle displacements. The fuselage shape remains intact (gray). The shape-morphing action is based on a two-step deformation: the single handle of the first step appears as the sphere with the biggest radius.

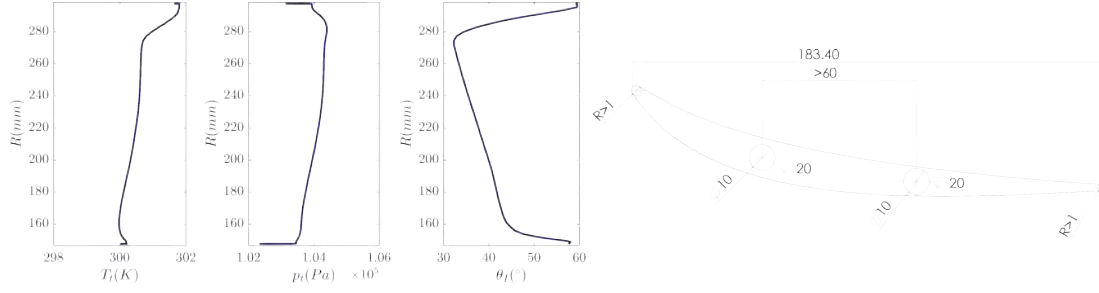


Figure 20: TUB stator. Left: From left to right, radial inlet profiles for the total temperature, total pressure and whirl angle. Right: Geometric definitions illustrated for the baseline geometry at the hub and shroud. Dimensions are in millimeters.

defined as

$$\alpha = \left( \frac{\int_{S_o} \left( \cos^{-1} \left( \frac{V_a}{\|\mathbf{V}\|} \right) \right)^2 \rho V_a \, dS}{\int_{S_o} \rho V_a \, dS} \right)^{\frac{1}{2}}, \quad (10)$$

where  $V_a$  is the axial velocity component. The shape is optimized for the operating point with the inlet total temperature, inlet total pressure and inlet whirl angle spanwise profiles of Figure 20. The pitch angle and inlet turbulence intensity are equal to  $0^\circ$  and 4%, respectively, being spanwise uniform. The outlet static pressure is adjusted to keep the mass flow at 9.0 kg/s (full annulus).

A series of geometric constraints must be imposed; some of them are illustrated in Figure 20. Specifically, the number of blades is fixed to 15; the axial chord of the blade is also constant; the minimum values of the leading and trailing edge circles' radius is 1 mm. Additionally, to mount the blade on the casing (hub and shroud), two holes for the fixture in the middle of the blade, with a radius of 2.5 mm and a depth of 20 mm each, are required. Thus, the blade thickness at these positions has to accommodate a cylinder of material with a radius of 5 mm and a depth of 20 mm to allow for cutting of the thread at both hub and shroud. The two holes can be placed arbitrarily inside the profile shape but must be at least 60 mm apart from each other. Geometric constraints are handled either explicitly or implicitly. The easiest way to handle geometric constraints is to explicitly define them by bounding the design space with lower and upper limits on the design variables. Displacing the blade only in the azimuthal direction of the cylindrical coordinates guarantees that the axial chord of the blade remains constant. To fit the cylinders inside the blade, an appropriate constraint function is defined by exploiting the availability of the CAD description into the optimization loop. This is based on the distance between them and the blade surface. Figure 22 illustrates the results of such a constraint handling for one of the optimal designs.

The parameterization employs the proposed two-step deformation method: the first step is controlled by 6 and the second by 24 RBF handles. Handles are displaced by changing their cylindrical coordinates in the azimuthal direction so that each one gives a DoF for the optimization. Hub and shroud must remain surfaces of revolution: because a non-rigid displacement of the NURBS control points would violate this constraint, these surfaces are extended in the spanwise direction and excluded from the morphing action, and their p-curves are updated by re-trimming as described in Section 3. The results of the optimization, employing the proposed parameterization, are illustrated in Figure 21 and are compared with the results obtained by an optimization assisted by a parametric turbomachinery row modeler [3]. This blade modeler is based on conventional multi-section blade parameterization, with sections defined by parameters with physical meaning defining the camber surface and thickness distributions. Such distributions of parameters are controlled by NURBS so as to provide a smooth design space to the optimization algorithm. The number of design variables for the blade modeler is 34. Both fronts of non-dominated solutions, Figure 21, are computed at the cost of 300 CFD simulations. The CAD-based-morphing optimization computes a front of non-dominated solutions competitive with that computed by the dedicated parametric CAD kernel. Some of the shapes resulting from the optimization are shown in Figure 22. Figures 23 and 24 illustrate fields related to the two objectives at the outlet and midspan of the CFD domain.

[34] employed 192 DoFs and optimized the blade for three operating points using as objective function a weighted combination of total pressure losses and stator exit flow angle (formulated differently than in this paper) and slightly different test case conditions. By giving equal weights to the two normalized

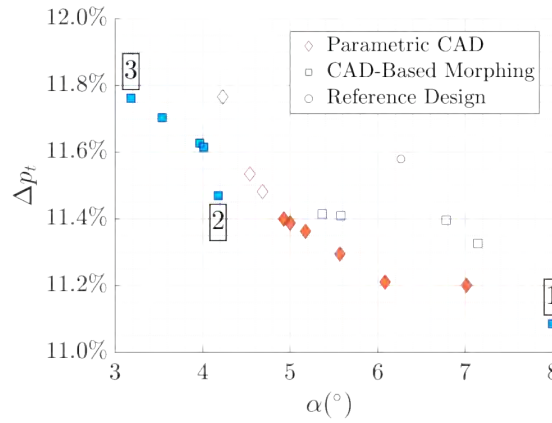


Figure 21: TUB stator. EA-based optimization with two parameterization methods: the fronts of non-dominated solutions computed using the CAD-based shape-morphing and a dedicated parametric CAD tool for the designs of turbo-machinery rows, are compared. The two independent optimizations were performed at the same computational cost. The combined front is shown as filled symbols. The shapes of the three Pareto points, marked with integers, are compared to the reference design shown in Figure 22.

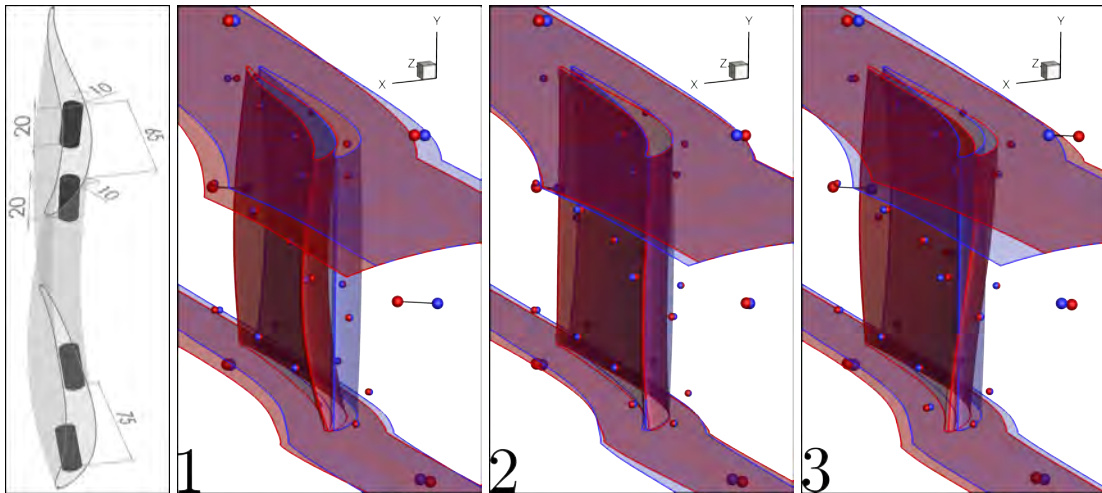


Figure 22: TUB stator. Left: Compliance with the geometric constraint regarding the fixture holes for the shape marked with "1" (dimensions are in millimeters). Right: Three shapes (red) from the front on non-dominated solutions in Figure 21, in comparison to the reference shape (blue), with the corresponding handle displacements. The shape-morphing action is based on a two-step deformation: the handles of the first step are shown with bigger symbols.

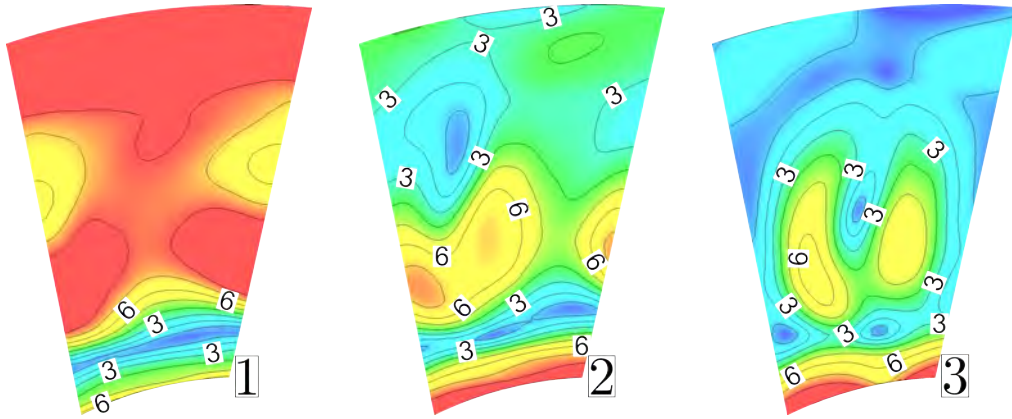


Figure 23: TUB stator.  $(\cos^{-1}(\frac{V_a}{|V|}))^2 \rho V_a$  (see equation 10) iso-areas at the stator outlet for three designs from the front on non-dominated solutions of Figure 21. Higher values imply higher objective function values.

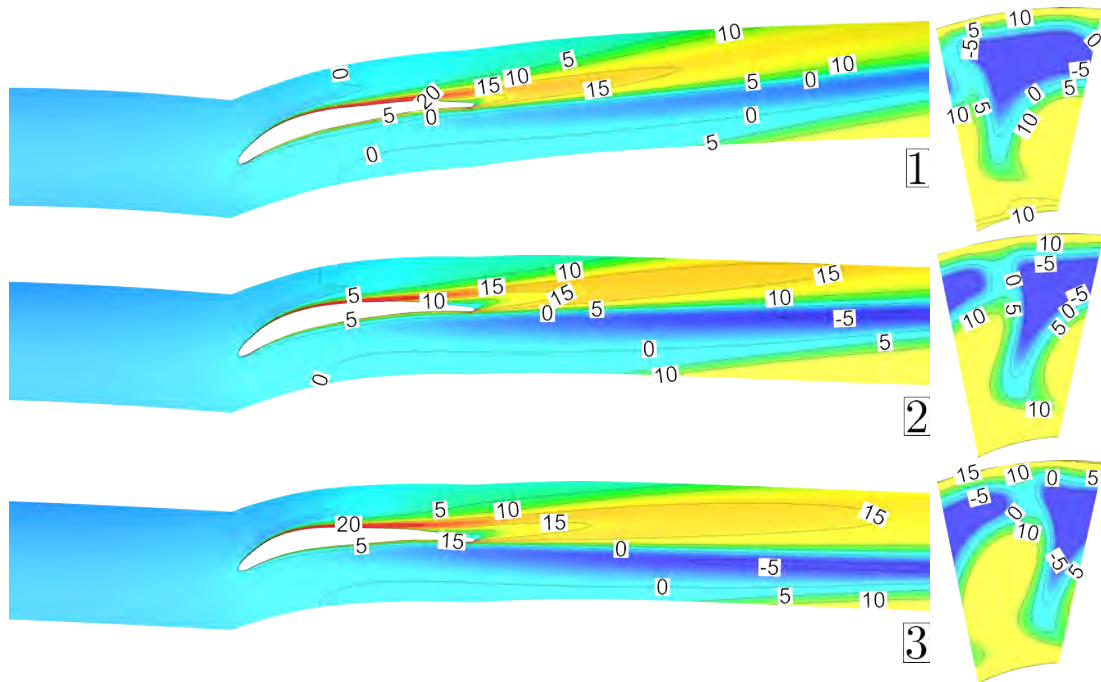


Figure 24: TUB stator.  $((p_t - \int_{S_t} p_t \rho V_a dS / \int_{S_t} \rho V_a dS) / (\int_{S_y} (p_t - p) \rho V_a dS / \int_{S_t} \rho V_a dS)) \rho V_a$  (see equation 9) iso-areas at the stator midspan (left) and outlet (right) for three designs from the front on non-dominated solutions of Figure 21. Higher values imply higher objective function values.

objectives, [34] decreased the exit flow angle by 38.6% and the total pressure losses by 3.2% w.r.t. the baseline. Comparatively, the Pareto front member marked with [2] in Figure 21 reduces the exit flow angle by 32.2% and the total pressure losses by 0.8% w.r.t. the baseline by employing 30 DoFs. However, differences in the test case conditions and constraints prevent a full comparison to be made.

## 6 Conclusions

This paper presented a new shape-morphing method that acts directly on CAD models, i.e., the B-Rep model of the body to be designed, which can thus be integrated into any aerodynamic (or hydrodynamic) shape optimization loop by ensuring that the optimal solution to be designed remains CAD compatible. The morphing action involves a small number of user-defined "handles" placed around or on the B-Rep shapes to be optimized. Their displacements are controlled by the optimization loop in a hierarchical or multi-step manner and affect the NURBS control points pertaining to the B-Rep model through RBF-based interpolations, with local or global support, depending on the step. An indispensable part of the proposed method is the adaptation of the 3D CFD grid to the changed B-Rep, still based on the RBF theory. An advantage is that the proposed method overcomes the need for re-meshing, which is typical in an optimization scheme with the CAD model into-the-loop. As demonstrated in the compressor stator case, Section 5.4, the proposed tool enables the exploration of shape variations that constructive CAD methods, despite their indisputable robustness, might "miss". Compared to B-Rep deformation methods that use the positions of NURBS control points as design variables directly, the proposed method enables the parameterization of generic shapes provided in standard CAD format with a relatively small number of parameters. For instance, in Section 5.3 a wing with tens of thousands of NURBS control points was parameterized with dozens of design variables, enabling the carrying out of efficient gradient-free optimization. The adoption of NURBS surfaces and curves enables the use of B-Rep models in standard formats, such as STEP format, to import and export shapes for further analysis in the design workflow. It also allows for their use in optimization loops, such as in the imposition of geometric constraints, as demonstrated for the compressor stator in Section 5.4. An additional advantage of the proposed method is that thanks to the multi-step deformations, it acts as a smooth multi-frequency shape-morphing tool. Moreover, it provides a scalable number of shape parameters employable in aerodynamic optimizations, enabling to enrich design spaces sequentially as shown in Section 5.2 for a duct shape. To conclude, the purpose of this paper was to propose a method that may reduce a lot the number of DoFs, without damaging the potential the optimization method has; of course, minor damages are of no importance at all. In this context, according to Section 5.1, the PE of the proposed method in a 2D case was compared with the standard NURBS parameterization. Even if the proposed method reduced the number of DoFs by 50%, the PE was reduced by just 16%, which is absolutely satisfactory. Results of shape optimizations conducted with different numbers of DoFs appear to be of similar quality; a similar reduction in drag was obtained with less than half of the DoFs.

## Acknowledgments

This research was funded by the People Programme (ITN Marie Curie Actions) of the European Union's H2020 Framework Programme (MSCA-ITN-2014-ETN) under REA Grant Agreement no. 642959 (IODA project). The first author was an IODA Early Stage Researcher.

The authors wish to express their sincere appreciation to Dr. K. Tsiakas and Dr. X. Trompoukis for the help with the in-house GPU-enabled RANS solver, Dr. V. Asouti for the support regarding the EASY software and D. Kapsoulis for the assistance with the computing software parametric effectiveness.

## References

- [1] Patrick N Koch et al. "Statistical approximations for multidisciplinary design optimization: the problem of size". In: *Journal of Aircraft* 36.1 (1999), pp. 275–286. URL: <https://doi.org/10.2514/2.2435>.
- [2] Stephen Wright and Jorge Nocedal. *Numerical Optimization*. Springer Science, 1999. URL: <https://doi.org/10.1007/b98874>.

- [3] Konstantinos T Tsiakas et al. “Shape optimization of turbomachinery rows using a parametric blade modeller and the continuous adjoint method running on GPUs”. In: *7th ECCOMAS Conference Proceedings*. 2016. URL: <https://doi.org/10.7712/100016.2085.9204>.
- [4] Jamshid A Samareh. “Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization”. In: *AIAA Journal* 39.5 (2001), pp. 877–884. URL: <https://doi.org/10.2514/2.1391>.
- [5] Athanasios G Liatsikouras and Guillaume Pierrot. “Soft handle triggering: a CAD-free parameterization tool for adjoint-based optimization methods”. In: *6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7) Conference Proceedings*. 2018. URL: <http://www.eccm-ecfd2018.org/admin/files/filePaper/p805.pdf>.
- [6] Daniel Sieger, Stefan Menzel, and Mario Botsch. “On shape deformation techniques for simulation-based design optimization”. In: *New Challenges in Grid Generation and Adaptivity for Scientific Computing*. Springer, 2015, pp. 281–303. URL: [https://doi.org/10.1007/978-3-319-06053-8\\_14](https://doi.org/10.1007/978-3-319-06053-8_14).
- [7] AM Morris, CB Allen, and TCS Rendall. “CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation”. In: *International Journal for Numerical Methods in Fluids* 58.8 (2008), pp. 827–860. URL: <https://doi.org/10.1002/flid.1769>.
- [8] ME Biancolini, IM Viola, and M Riotte. “Sails trim optimisation using CFD and RBF mesh morphing”. In: *Computers & Fluids* 93 (2014), pp. 46–60. URL: <https://doi.org/10.1016/j.compfluid.2014.01.007>.
- [9] Trevor T Robinson et al. “Optimizing parameterized CAD geometries using sensitivities based on adjoint functions”. In: *Computer-Aided Design and Applications* 9.3 (2012), pp. 253–268. URL: <https://doi.org/10.3722/cadaps.2012.253-268>.
- [10] Mario Jaime Martín et al. “Non-uniform rational B-splines-based aerodynamic shape design optimization with the DLR TAU code”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 226.10 (2012), pp. 1225–1242. URL: <https://doi.org/10.1177/0954410011421704>.
- [11] Shenren Xu, Wolfram Jahn, and Jens-Dominik Müller. “CAD-based shape optimisation with CFD using a discrete adjoint”. In: *International Journal for Numerical Methods in Fluids* 74.3 (2014), pp. 153–168. URL: <https://doi.org/10.1002/flid.3844>.
- [12] Alexander Brune, Thiago Weber Martins, and Reiner Anderl. “Morphing boxes for the integration of shape optimization in the product design process”. In: *Computer-Aided Design and Applications* 15.2 (2018), pp. 219–226. URL: <https://doi.org/10.1080/16864360.2017.1375672>.
- [13] Robert McNeel et al. *Rhinoceros Version 6*. 2018. URL: <https://www.rhino3d.com/>.
- [14] Qiang Zou and Hsi-Yung Feng. “Push-pull direct modeling of solid CAD models”. In: *Advances in Engineering Software* 127 (2019), pp. 59–69. URL: <https://doi.org/10.1016/j.advengsoft.2018.10.003>.
- [15] Flavio Gagliardi, Konstantinos Tsiakas, and Kyriakos Giannakoglou. “A two-step mesh adaptation tool based on RBF with application to turbomachinery optimization loops”. In: *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*. Springer, 2019, pp. 127–141. URL: [https://doi.org/10.1007/978-3-319-89890-2\\_9](https://doi.org/10.1007/978-3-319-89890-2_9).
- [16] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 1997. URL: <https://doi.org/10.1007/978-3-642-59223-2>.
- [17] A. de Boer, M.S. van der Schoot, and H. Bijl. “Mesh deformation based on radial basis function interpolation”. In: *Computers and Structures* 85.1114 (2007), pp. 784–795. ISSN: 0045-7949. URL: <https://doi.org/10.1016/j.compstruc.2007.01.013>.
- [18] Gregory E Fasshauer. *Meshfree approximation methods with Matlab*. Vol. 6. World Scientific Publishing Company, 2007. URL: <https://doi.org/10.1142/6437>.

- [19] Trevor T Robinson, Cecil G Armstrong, and Hung Soon Chua. “Determining the parametric effectiveness of a CAD model”. In: *Engineering with Computers* 29.1 (2013), pp. 111–126. URL: <https://doi.org/10.1007/s00366-011-0248-4>.
- [20] Dheeraj Agarwal et al. “Using parametric effectiveness for efficient CAD-based adjoint optimization”. In: *Computer-Aided Design and Applications* 16.4 (2019), pp. 703–719. URL: <https://doi.org/10.14733/cadaps.2019.703-719>.
- [21] *6th AIAA CFD Drag Prediction Workshop*. URL: <https://aiaa-dpw.larc.nasa.gov/> (visited on 04/01/2018).
- [22] James W. Demmel, John R. Gilbert, and Xiaoye S. Li. “An asynchronous parallel supernodal algorithm for sparse Gaussian elimination”. In: *SIAM J. Matrix Analysis and Applications* 20.4 (1999), pp. 915–952. URL: <https://doi.org/10.1137/s0895479897317685>.
- [23] Holger Wendland. “On the smoothness of positive definite and radial functions”. In: *Journal of Computational and Applied Mathematics* 101.1-2 (1999), pp. 177–188. URL: [https://doi.org/10.1016/S0377-0427\(98\)00218-0](https://doi.org/10.1016/S0377-0427(98)00218-0).
- [24] Varvara G Asouti et al. “Unsteady CFD computations using vertex-centered finite volumes for unstructured grids on Graphics Processing Units”. In: *International Journal for Numerical Methods in Fluids* 67.2 (2011), pp. 232–246. URL: <https://doi.org/10.1002/flid.2352>.
- [25] Flavio Gagliardi and Kyriakos C. Giannakoglou. “A two-step radial basis function-based CFD mesh displacement tool”. In: *Advances in Engineering Software* 128 (2019), pp. 86–97. ISSN: 0965-9978. DOI: <https://doi.org/10.1016/j.advengsoft.2018.11.011>. URL: <http://www.sciencedirect.com/science/article/pii/S0965997818304241>.
- [26] *The EASY (Evolutionary Algorithms SYstem) software*. URL: <http://velos0.ltt.mech.ntua.gr/EASY> (visited on 04/01/2018).
- [27] Mark Drela. “Pros and cons of airfoil optimization”. In: *Frontiers of Computational Fluid Dynamics* 1998 (1998), pp. 363–381.
- [28] JJ Thibert, M Granjacques, and LH Ohman. *AGARD advisory report no. 138 - Experimental database for computer program assessment*. Advisory Group for Aerospace Research and Development, 1979.
- [29] *2nd AIAA CFD Drag Prediction Workshop*. URL: <https://aiaa-dpw.larc.nasa.gov/Workshop2/> (visited on 04/01/2018).
- [30] Joël Brezillon and Richard P Dwight. “Aerodynamic shape optimization using the discrete adjoint of the Navier-Stokes equations: Applications towards complex 3D configurations”. In: *KATnet II Conference on Key Aerodynamic Technologies paper*. 36-1. 2009. URL: <http://resolver.tudelft.nl/uuid:7c8ee2b9-735c-430d-bca1-8e5581b6f852>.
- [31] Ilias Vasilopoulos et al. “Aerodynamic optimization of the TurboLab Stator: A comparative study between conventional and adjoint-based approaches”. In: *International Conference on Numerical Optimization Methods for Engineering Design NOED*. 2016.
- [32] Jens-Dominik Mueller, Jan Hueckelheim, and Orest Mykhaskiv. “STAMPS: a finite-volume solver framework for adjoint codes derived with source-transformation AD”. In: *2018 Multidisciplinary Analysis and Optimization Conference*. 2018, p. 2928.
- [33] Orest Mykhaskiv et al. “NURBS-based and parametric-based shape optimization with differentiated CAD kernel”. In: *Computer-Aided Design and Applications* 15.6 (2018), pp. 916–926. URL: <https://doi.org/10.1080/16864360.2018.1462881>.
- [34] Jan Mihalyovics et al. “Numerical and experimental investigations on optimized 3D compressor airfoils”. In: *ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition*. 2018, V02AT39A038. URL: <https://doi.org/10.1115/GT2018-76826>.
- [35] *TU Berlin TurboLab Stator Description*. URL: <http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/> (visited on 04/01/2018).