COMPUTER-AIDED DESIGN & APPLICATIONS
2018, VOL. 15, NO. 6, 916–926
https://doi.org/10.1080/16864360.2018.1462881

NURBS-based and parametric-based shape optimization with differentiated CAD kernel

Orest Mykhaskiv[a], Mladen Banović[b], Salvatore Auriemma[c], Pavanakumar Mohanamuraly[a], Andrea Walther[b], Herve Legrand[c] and Jens-Dominik Müller[a]

[a]Queen Mary University of London, UK; [b]University of Paderborn, Germany; [c]OpenCASCADE

ABSTRACT
Numerical optimization is becoming an essential industrial method in engineering design for shapes immersed in fluids. High-fidelity optimization requires fine design spaces with many design variables, which can only be tackled efficiently with gradient-based optimization methods. CAD packages that are open-source or commercially available do not provide the required shape derivatives, but impose to compute them with expensive, inaccurate and non-robust finite-differences.

The present work is the first demonstration of obtaining exact shape derivatives with respect to CAD design parametrization by applying algorithmic differentiation to a complete CAD system, in this case the Open Cascade Technology (OCCT) CAD-kernel. The extension of OCCT to perform shape optimization is shown by using parametric models based on explicit parametrizations of the CAD model and on implicit parametrizations based on the BRep (NURBS). In addition, we demonstrate the imposition of geometric constraints for both approaches, a salient part of industrial design, and an intuitive method of storing them in standard CAD format. The proposed method is demonstrated on a turbo-machinery test case, namely the optimization of the TU Berlin Stator.

KEYWORDS
aerodynamic shape optimization; CAD-based sensitivities; gradient method

1. Introduction

In engineering workflows it is common practice to maintain master CAD models, which then serve as a foundation for further design and development. In complex system design these base geometries enable cross-discipline collaboration and therefore minimize the time needed to bring an industrial component to production. In an optimization workflow it is hence important to maintain consistency between CAD model and the meshes of the analysis disciplines, which is typically done through parametric CAD models. Such workflows can be driven by stochastic optimizers and only require evaluation of the CAD shape.

High-fidelity shape optimization of immersed bodies subject to fluid dynamics, such as aeroplanes, turbines, vehicles or ducts, requires very rich design spaces with many design variables. Tackling these rich design spaces is not computationally feasible with stochastic optimization methods such as Evolutionary Algorithms: the convergence to the optimum requires far too many evaluations of an expensive computational model such as CFD. Gradient-based optimization has been shown to be feasible, and has widely been adopted for these problems.

Significant progress has been made over the past decades with computing gradients of objective functions with respect to mesh point perturbation for Computational Fluid Dynamics (CFD) solvers. In particular, the adjoint method [5,8,10,12,15] allows to compute these gradients accurately, consistently and with low computational cost.

Adjoint CFD methods can efficiently compute the sensitivity of the objective function w.r.t. a perturbation of an individual grid node, the next term in the chain-rule of the gradient computation is the derivative of the grid node position w.r.t. design parameters, i.e. we need to define a parametrization and compute its shape derivative.

A simple parametrization is the node-based approach where the design variables are the displacement of the grid nodes on the shape. Although the mesh node positions present the richest design space that computational tools can consider, the approach allows surfaces with oscillatory high-frequency noize. This can be addressed by regularization (smoothing) [9]. Alternative approaches are Free Form Deformation [14] or radial basis function [3]. At convergence to the optimum,

CONTACT Orest Mykhaskiv o.mykhaskiv@qmul.ac.uk; Mladen Banović mladen.banovic@math.upb.de
Salvatore Auriemma salvatore.auriemma@opencascade.com

the optimized mesh is re-created in CAD, but this usually inquires significant approximations and inaccuracies degrading the quality of the optimum.

As an alternative, CAD-based methods maintain a consistent CAD model throughout the optimization. The challenge here is to compute the shape sensitivities of the CAD model. Robinson et al. [13] apply finite differences to parametric CAD models created in commercial 'black-box' CAD systems. The CAD sensitivity is computed as the local distances between surface triangulations of the original CAD model and the one with the perturbed design parameters. A similar approach is followed in [4], where finite differences are applied to the open-source CAD-kernel Open CASCADE Technology (OCCT). Where feasible the authors compute analytical derivatives e.g. for CAD primitives and shapes (cube, sphere, etc.).

If source code is available, algorithmic differentiation (AD) can be used to compute derivatives of any computational algorithm. In [19] a small in-house CAD kernel supporting NURBS was automatically differentiated and provides analytical derivatives. In this paper we exploit recently differentiated version of essentially complete OCCT kernel [1], [2]. Although differentiation of a complete CAD-kernel is a non-trivial and time-consuming task, the differentiation of OCCT allows to get exact derivatives without numerical noise in any of CAD modelling algorithms available in OCCT. This makes the differentiated OCCT practical for a wide range of parametrizations and geometrical manipulations. Moreover, the efficiency of the computation of the shape sensitivities is superior to finite differences and more robust, which encourages shape exploration in large-dimensional CAD spaces.

The definition of the design space is crucial for aerodynamic shape optimization in CAD-based methods: an optimal result can only be achieved if the relevant mode is present that can harness the important aspect of the flow physics. Therefore, widely used parametric CAD models require from the designer a proper engineering judgement during initial design. To respond to these challenges, several application-specific parametrization tools were developed [6], [17]. Taking to account extensive engineering experience, these tools allow to parametrize the shapes with conventional and intuitive design parameters (trailing/leading edge radius, blade thickness, wing span, etc.). These parameters are then varied during the design optimization loop. Furthermore, an explicit control over design variables also allows to incorporate geometrical constraints directly in the parametrization. These approaches, termed here 'explicit' parametrizations as they need to be set up manually. They are widely used for typical case scenarios and flows, good experience in their definition is available. However, increasingly 'out of the box' designs are required to work in new configurations, work with new materials or better exploit the interaction between disciplines in multi-disciplinary optimization. In these situations a good choice of design parametrization is often not evident.

Alternatively to the previous approach, instead of changing the parameters of the model's construction algorithm, one can directly modify the geometry of the resulting shape, so-called BRep (Boundary Representation) [19], [21]. We term this approach 'implicit' parametrizations, as there is no specific user effort to define the design space. Changes to this BRep data (control point positions and weights of corresponding NURBS patches) eliminate the initial parametrization, but propose rich design spaces, which can straightforwardly be refined adaptively by inserting additional control points. The resulting design space can be made to guarantee to include all relevant modes, and combined with adjoint gradient computation, there is no computational penalty. However, convergence of the optimising algorithm such as steepest descent may be slower, and preconditioning methods will be needed. The NURBS-based method is CAD-vendor independent, and requires only a generic CAD-file (STEP, IGES, etc.), eluding problems with parametrization tree and making the optimization more automatic.

The paper proposes two major elements to overcome obstacles with integration of CAD into the design loop:

a) We describe the application of automatic differentiation to a complete CAD system and demonstrate the accuracy and efficiency of computation of the shape sensitivities. These advances allow us to build gradient-based optimization workflows with the CAD-model being updated inside the optimization loop.
b) We present two alternative approaches that are supported by this differentiated CAD kernel, both with their merits and disadvantages. The 'explicit' parametrization is closer to current practice in aeronautics and turbo-machinery, but may limit the optimum due to restrictive design spaces. The alternative 'implicit' parametrization allows to automatically derive a sufficiently rich design space, however may impair convergence to the optimum.

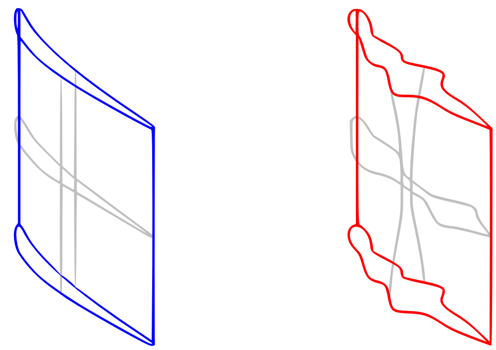
In this paper the differentiated OCCT is used to optimize TU Berlin Stator test case [23]. A brief introduction to the OCCT differentiation can be found in Sec. 2. Parametrization of the stator blade with conventional turbomachinery parameters is described in Sec. 4. Section 5 describes the necessary ingredients for NURBS-based optimization including corresponding constraint impositions, followed by results for both approaches in Sec. 6.

2. Automatic differentiation of OCCT CAD-kernel

Geometrical sensitivities of CAD model w.r.t. its parametrization are necessary to perform CAD-based shape optimization. The exact derivatives are obtained by algorithmic differentiation (AD) of the open-source CAD-kernel *Open CASCADE Technology* (OCCT) using the AD software tool ADOL-C (*Automatic Differentiation by OverLoading in C++*) [2]. The ADOL-C tool requires all variables that may be considered as differentiable quantities to be declared as an `adouble` type to denote an *active* variable. This requires one to replace the type declaration of almost all floating point variables in the source code to the `adouble` type. The idea although straightforward to implement requires significant man-hours to fix compile and run-time errors.

ADOL-C [18] provides two kinds of differentiation options: (i) *trace-based* and (ii) *traceless*. Each one implements a different version of the `adouble` class leading to two distinct computational algorithms. In the *trace-based* option, operator-overloading is used to generate an internal representation (*trace*) of the function to be differentiated. Then the ADOL-C driver routines are executed on the generated *trace* to compute the necessary gradients. In the *traceless* mode, the gradient computation is propagated directly during the function evaluation, along with the function values. This mode is simpler to use as every overloaded operator embeds both primal and gradient code in its definition. On the contrary, it is not as powerful as the *trace-based* option since only the *forward/tangent* mode of AD is possible. In the *trace-based* option both the *forward/tangent* and *reverse/adjoint* mode of AD are possible, where reverse mode of AD can dramatically reduce the temporal complexity of the gradient computation. The theory behind the forward and reverse mode can be found in reference [7].

We successfully differentiated OCCT using both *trace-based* and *traceless* modes provided by ADOL-C. Hence it is possible to compute the CAD sensitivities (i.e., gradients of CAD surface points w.r.t. design parameters of the model) both in the forward and reverse mode of AD. We verified the correctness of the differentiated CAD kernel against (central difference) finite difference results. The geometric algorithms involved in the TU Berlin Stator blade parametrization in OCCT were also individually verified for correctness against finite difference. A qualitative comparison of AD and FD surface sensitivities w.r.t. one design parameter is shown in Fig. 2. In order to validate the reverse mode differentiation of OCCT against the forward mode of AD, we developed an optimization test case within the CAD system. It is organized as follows:



(a) Original stator blade shape (b) Perturbed stator blade shape

Figure 1. CAD optimization with two stator blades

1. Construct two blades: original and perturbed one, see Fig. 1.
2. Sample final NURBS surfaces with 20K pairs of (u, v) parametric coordinates. These parametric coordinates are later used in NURBS algorithms to evaluate the corresponding three-dimensional points (x, y, z).
3. Define an objective function as the sum of squared distances of all (x, y, z) point pairs.
4. Declare the original design parameters as independent variables of the system.
5. Minimize the objective function by using the limited-memory BFGS optimization algorithm with boundary constraints (L-BFGS-B) [22].

The primary objective of this test case is to match the surfaces of two blades by modifying the design parameters of the original blade. Since we have two versions of differentiated OCCT kernel, one compiled with *traceless* and another one with *trace-based* ADOL-C headers, the optimization was performed twice. We observed small differences between the gradients obtained from the two modes of AD, as shown in Fig. 3. Similar differences (same order of magnitude) were present in the primal results. We attribute the difference to floating-point round-off errors, since the floating point operations (and order) differ between *traceless* and *trace-based* ADOL-C headers. The differences relative to the objective function value, which is $O(10^5)$ are quite insignificant. The high peaks in differences occur, for example near gradient index 100 (Fig. 3), in the regions of low sensitivity values where round-off errors dominate.

The run-time ratios of the optimization test case for both forward and reverse mode AD are shown in Table 1. Note that the run-time ratio is defined as the ratio

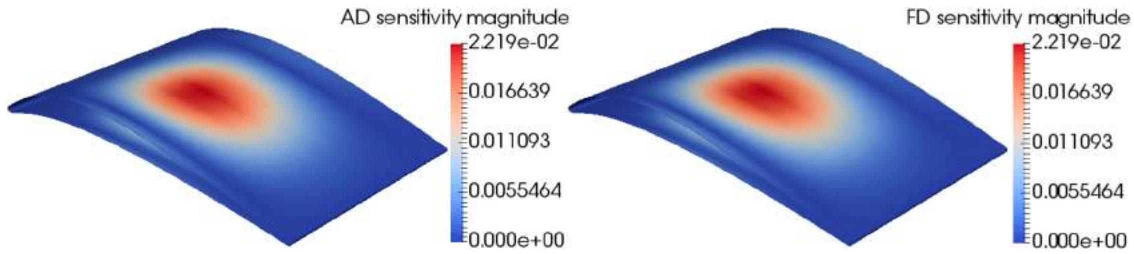


Figure 2. Blade sensitivities evaluated by AD (left) and Finite Differences (right)

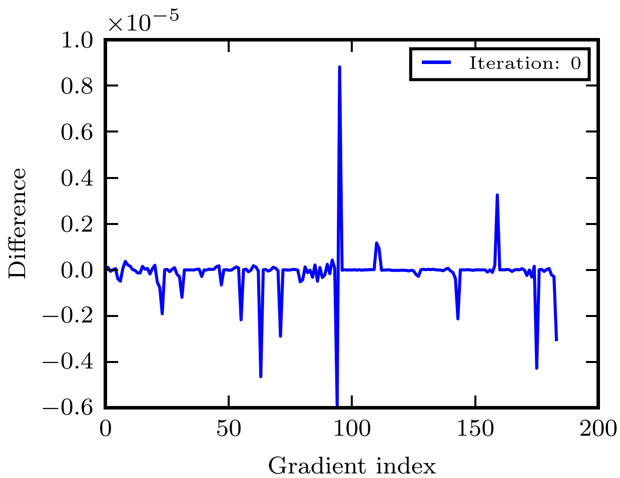


Figure 3. Gradient differences between two modes of AD for initial blade optimization iteration

Table 1. Timings for initial blade optimization iteration done with original and differentiated (AD) sources

	Original sources	AD Forward mode (traceless)	AD Reverse mode (trace-based)
Avg. time (seconds)	0.09	13.27	6.99
Run-time ratio		147.44	77.67

between the original and differentiated OCCT sources. The *trace-based* reverse mode AD is quite efficient and it is overall 47% faster than the *traceless* forward mode for the optimization test case.

3. Differentiated OCCT and adjoint CFD coupling

In a typical aerodynamic shape optimization process one minimizes a cost function J (usually scalar like lift, drag, etc.) with respect to the CAD geometry with design parameters α and subject to geometry and flow constraints R [8]:

$$\min_{\alpha} \; J(U(X(\alpha)), X(\alpha), \alpha) \tag{1}$$

$$R(U(X(\alpha)), X(\alpha)) = 0. \tag{2}$$

Equation (2) describes the flow field within the domain of interest by system of Reynolds-Averaged Navier-Stokes equations, with the state variable U and a computational mesh coordinates X, which depend on design parameters α. In case of large amount of design parameters (usually the case in industrial applications) the adjoint method proves to be computationally efficient and could be derived by application of a chain rule to the system (1)-(2) yielding:

$$\frac{dJ}{d\alpha} = \left[\frac{dJ}{dX} + \nu^T f\right]\frac{\partial X}{\partial \alpha}, \tag{3}$$

where

$$f = -\frac{\partial R}{\partial X}. \tag{4}$$

Here ν represents the solution of adjoint equations:

$$\left(\frac{\partial R}{\partial U}\right)^T \nu = \frac{\partial J}{\partial U}. \tag{5}$$

After computing the solution of primal and adjoint equations (2),(5), one can rewrite cost function gradient in terms of surface grid points derivatives:

$$\frac{dJ}{d\alpha} = \frac{dJ}{dX_S}\frac{dX_S}{d\alpha}. \tag{6}$$

Here, the relation (spring analogy, inverse distance weighting) between volume and surface grid points displacement is used $X = X(X_S)$. The first term in (6), usually called *CFD sensitivity*, corresponds to the flow sensitivity in the surface grid points X_S. These derivatives could be calculated by several available CFD solvers that have implemented the adjoint method. In this work we use our in-house discrete adjoint solver STAMPS (previously *mgopt*) [20].

The second term (*CAD sensitivity*) represents the derivative of the surface grid points X_S with respect to the CAD model design parameters. This part is calculated in the automatically differentiated version of OCCT [1]. The differentiated OCCT provides the derivatives for almost every possible CAD parametrization and geometrical manipulation.

Equipped with these derivatives, we compose them in the total gradient, which is then used in iterative gradient-based optimization loop:

$$\alpha^{(n+1)} = A\left(\alpha^{(n)}, \frac{dJ}{d\alpha}(\alpha^{(n)})\right), \tag{7}$$

with A as an optimization algorithm. Next sections describe two cases of the above mentioned method, depending on the nature of CAD design parameter α: as design variable in parametric CAD model or BRep/NURBS parametrization.

4. Parametric CAD-model for TU Berlin Stator and constraints

The TU Berlin TurboLab Stator Blade [23] is a typical turbomachinery optimization test case, where geometrical constraints strongly influence the final optimized shape. The test case prescribes the following geometrical constraints on the blade: (i) minimum radius of the leading and the trailing edge, (ii) minimum thickness of the blade (iii) minimum thickness near the hub and the shroud to accommodate the four mounting bolts and (iv) constant axial chord length. In the present work, we re-parametrized the blade in OCCT such that all constraints except the thickness constrains for the mounting bolts are explicitly embedded in the parametrization. These constrains can readily be provided to any optimizer workflow.

4.1. 2D Profile parametrization

The blade parametrization starts by defining a 2d profile. We used B-spline curves to represent the 2d blade profiles, since they provide a rich and flexible space for the parametrization [17]. The 2d blade profile is generated using a camber-line (shown in Fig. 4) represented by a B-spline curve and characterized by seven control points. We distribute eight reference points ($P1, \ldots, P8$), as shown in Fig. 4, along the camber line using a cosine function. The cosine function is used to cluster points near the leading and trailing edge (*LE* and *TE*) of the camber-line. The control points for the suction and pressure side B-splines curves are generated as equidistant offsets of the reference points normal to the camber-line (Fig. 4). Finally, the suction and pressure side curves are smoothly joined using the specified radius of curvature satisfying G2 continuity.

The AB length (Fig. 4) in a B-spline curve of degree n is:

$$AB = \sqrt{curvature \cdot CH \cdot \frac{n-1}{n}} \tag{8}$$

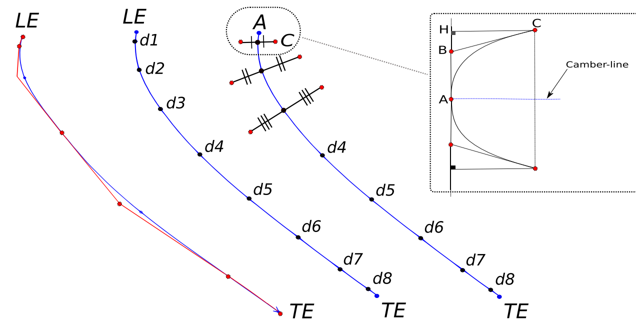


Figure 4. Left: Camber-line (blue) with corresponding control polygon (red) and uniform point distribution; Right: Construction of pressure/suction control points; Imposition of curvature (G2 continuity) at the LE

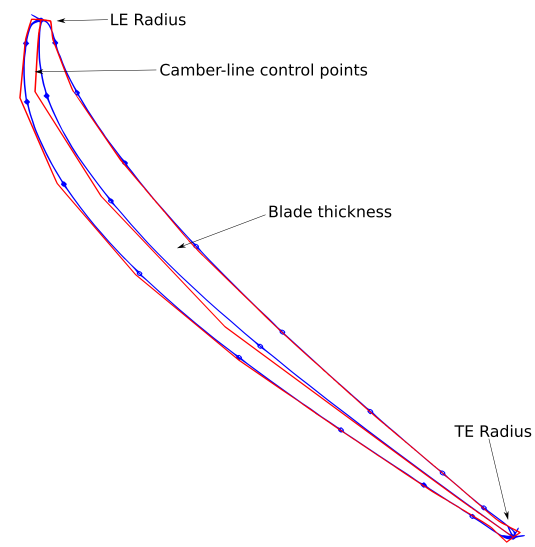


Figure 5. Section parameters

where AB is the distance between control point A and B and CH is the distance of control point C from the AB line. Therefore, it is possible to impose the curvature in the point A.

This approach is applied to suction and pressure B-splines. In particular, the two curves have the same radius of curvature at the LE. This radius is controlled as design parameter of the optimization. The same approach is also used for the TE radius. Thus the G2 continuity is kept along all the section.

In summary, the 2d profile consists of 23 parameters of which, (i) 10 parameters control thickness (2 of them are the radii of TE and LE) and (ii) 13 parameters control the camber-line movement and, therefore, its angle, as shown in Fig. 5.

4.2. 3D Parametrization

The 3d blade parametrization is based on a cross-sectional design approach - the *lofting*. This approach

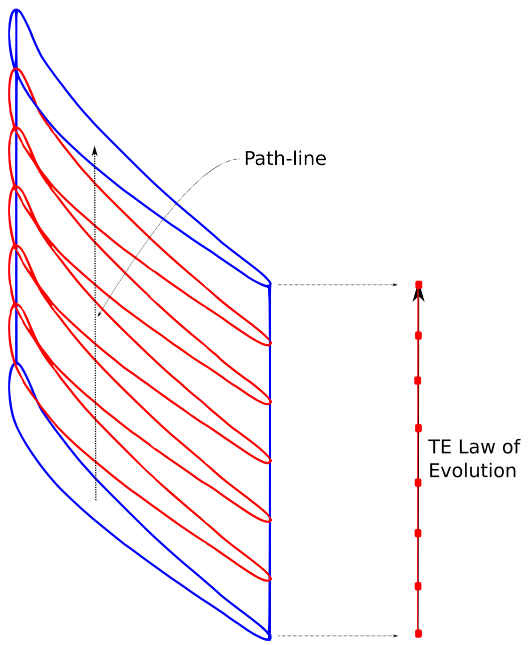


Figure 6. Blade skeleton and TE law of evolution in the 3D domain

takes *n*-slices (2d profiles) as input and constructs final B-spline surface using an OCCT approximation tool. The slices are generated along a blade span defined as a B-spline curve, the *path-line*. Each 2d profile parameter is characterized by a law of evolution along the *path-line*. The laws are defined as B-spline curves, consisting of 8 control points each. These control points are the design parameters of the optimization. Their total number is 184 ($23 \cdot 8$). An example of the blade construction using seven slices is shown in Fig. 6.

4.3. Optimization constraints

The limited memory BFGS algorithm with boundary constraints (L-BFGS-B) is used as optimizer. The constraints specified in the L-BFGS-B are as follows:

- G2 continuity: imposed along all the section based on the geometrical construction.
- Axial chord: the axial-coordinate of the last camber-line control point is set equal to the axial-coordinate of the first control point plus the constant axial chord value.
- Thickness distribution: the thickness between the suction and pressure surface is approximated using the corresponding B-Spline control point distances. Therefore this constraint has to be verified a posteriori.
- LE and TE radii: The lower bound values are specified.

5. NURBS-based optimizations and constraints

The NURBS-based optimization technique with continuity and geometrical constraints (so-called NSPCC approach) was initially proposed in [19] and [21]. In this paper, we extend and automate the NSPCC method further. The authors in [19] use a modest in-house CAD kernel, but we substitute it with a comprehensive OCCT CAD kernel and benefit from the extensive CAD functionality. The major updates and novelties are related to the refinement of the CAD-space, new constraints capabilities (curvature), recovery of the violated geometrical constraints and the storage of the constraints in standard CAD formats. In the current NSPCC version the role of the CAD tool is more profound, while the amount of manual constraint set-up is reduced. This brings NURBS-based optimization closer to the industrial workflows and creates an alternative to parametric CAD-models optimization.

5.1. NURBS-based design

The advantage of NURBS-based approach is that in most cases no preprocessing (understanding of initial parametrization tree, re-parametrization in another tool, aerodynamic intuition to define proper CAD space, etc.) is needed. CAD-vendor neutral boundary representation could be retrieved directly from the standard CAD files (STEP, IGES, etc.), which usually contain collection of NURBS patches.

Since OCCT is already equipped with an efficient reader of standard CAD formats, its differentiated version allows to compute the sensitivity information in any point of the surface with respect to control points position of governing NURBS. Therefore the CAD sensitivity defined in Eq. 6 is obtained for every surface:

$$\frac{\partial X_S}{\partial \alpha} = \frac{\partial X_S}{\partial P} = \underset{3\times M}{\Bigg\downarrow} \overset{\xrightarrow{\quad 3\times N \quad}}{\begin{pmatrix} \frac{\partial X_{S1}}{\partial P_1} & \frac{\partial X_{S1}}{\partial P_2} & \cdots & \frac{\partial X_{S1}}{\partial P_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial X_{SM}}{\partial P_1} & \frac{\partial X_{SM}}{\partial P_2} & \cdots & \frac{\partial X_{SM}}{\partial P_N} \end{pmatrix}}. \qquad (9)$$

Here M and N are the total number of surface mesh points X_S and control points P respectively. Moreover, with OCCT one can easily and intuitively refine design space by adding extra control points with knot insertion algorithm [11]. This operation does not change the shape or degree of the surface, but establishes more local control due to local support properties of the splines. This is

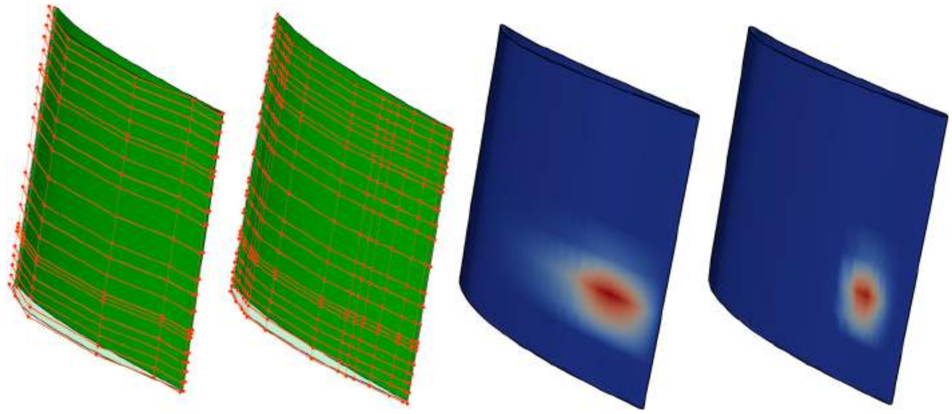


Figure 7. Left: Initial (9x22) and refined(18x22) Control Point Net of TUB Stator; Right: Corresponding changes in CAD Surface Sensitivity with respect to movement of the single control point

clearly visible in the changing pattern of the CAD sensitivities shown in Fig. 7. These very narrow sensitivities potentially could cope better with small flow features not 'visible' for more global parametric sensitivity (Fig. 2).

At the moment, the refinement is performed manually prior to the optimization, but this process can be automated with the CFD sensitivity field as a sensor for refinement.

It is important to note that in some cases the NURBS surfaces extracted from standard files are not suitable for direct NURBS-based optimization due to enormous clustering of control points (sometimes as fine as the computational mesh). The root of this problem lies in the creation of the initial shape (morphed from STL, extensive surface trimming, etc.). In these cases reverse engineering and re-approximation of the surfaces might be required.

5.2. Constraints

CAD models are usually constructed from multiple adjacent patches. Therefore, modifications of control points individually on patches can violate (i) patch-continuity (holes between the CAD faces, non-smooth shapes) or (ii) other geometrical constraints. We alleviate this problem by filtering out the shape modes with undesired constraints violations using discrete spaces constructed using test-points [19]. Conceptually, the approach requires that the constraints are satisfied on the particular set of points defined on the surface (test-points). We avoid distinguishing the continuity and the geometrical constraints by bringing them under one framework.

In the TU Berlin Stator test case, several geometrical constraints are present and were introduced in the previous section. Several methods were devized to accelerate and automate the process of test-point distribution. Firstly, we identify topological entities (e.g. edges, parts of surfaces, etc.) necessary for constraint imposition. For instance, to distribute test-points along the leading edge (curvature and continuity constraints), we use OCCT to find two parametric curves (*PCurves*) of the edge on two adjacent faces (Fig. 8). Then we use OCCT to uniformly distributes points (in $1d$ parametric space) along each *PCurve*. As a result two pairs of test-points are generated each belonging to the respective *PCurves*. The test-point pairs along the *edge1* (LE) and *edge2* (TE) are then used to impose continuity and curvature constraints. It is also possible to generate test-point pairs on *PCurves* at arbitrary location on a given patch face (connect the predefined endpoints $(u1, v1)$ and $(u2, v2)$ in the parametric space of the face). The generated test-points pairs are then used to impose thickness constraints between the two patches of the Stator. The treatment of constraints on a topological level allows to store these *PCurves* in a standard CAD file. This enables visualization and inspection of the constraints during optimization. For example in Fig. 9 the pairs of *PCurves* are shown, where *PCurves* pairs are identically colored. In addition, the *PCurves* can be stored and visualized as wire-frame objects with vertices as test-points.

Once all necessary test-points are distributed, standard OCCT geometric algorithms (*distance, curvature, normal, etc.*) can be used to compute the following constraints:

- Distance constraints
  To fix the distance d_r between two test-points (X_{t1}, X_{t2}), the following function is constructed:

$$C_d = distance(X_{t1}, X_{t2}) - d_r = 0\,. \qquad (10)$$

  This constraint is used to ensure G0 continuity ($d_r = 0$ is then used) and the constant axial chord length. Similarly, the minimum thickness (T_{min}) constraint, which is required in the middle of the blade and for

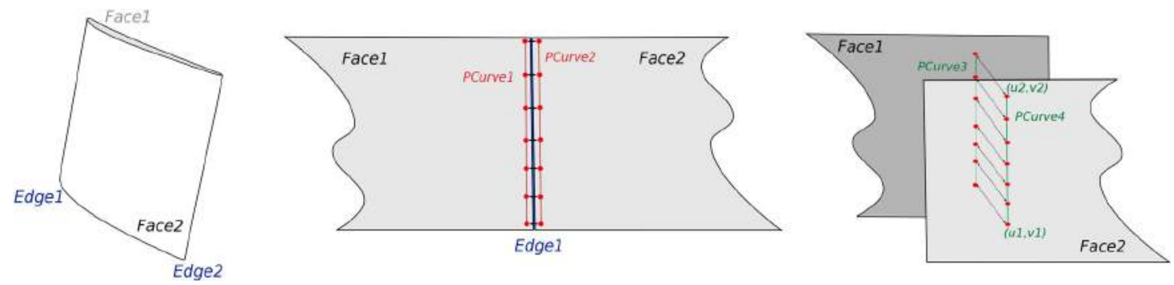


Figure 8. Left: TUB Stator Topology; Middle: Test-points distribution along *PCurves* of the common edge; Right: Test-points on generic *PCurves* of surfaces

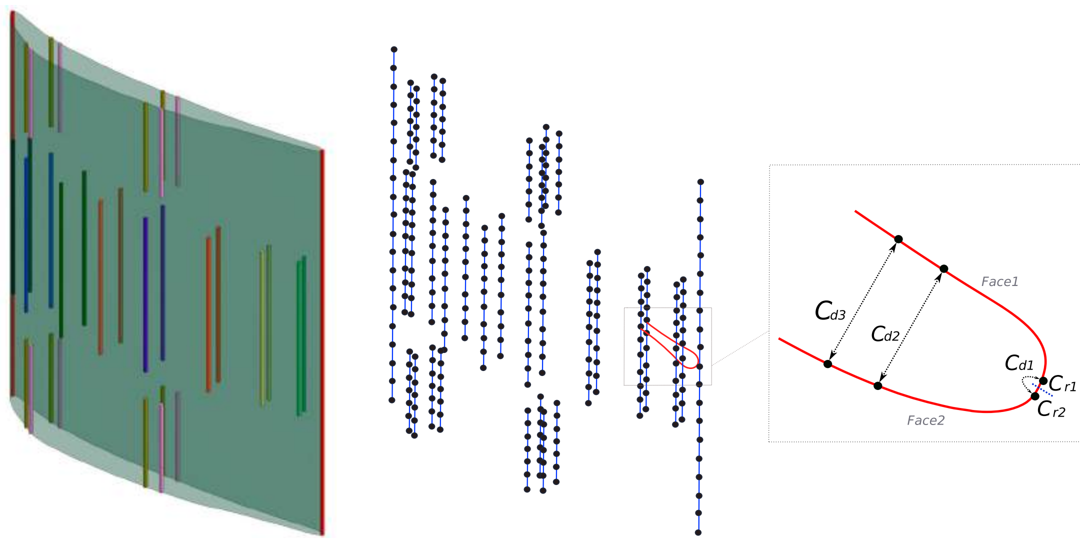


Figure 9. Left: Constraint visualization from the STEP file; Right: Constraints computation on the testpoints

the bolts, corresponds to inequality constraint and is represented with:

$$C_d = 1 - min\Big(1, \frac{distance(X_{t1}, X_{t2})}{T_{min}}\Big) = 0\,. \quad (11)$$

- Radius of curvature constraint
  OCCT allows to compute minimum and maximum curvature in any point of the surface. Therefore, the radius in the test-point corresponding to TE and LE can be calculated as $r = 1/curvature(X_{t1})$. Constraint function bounding the minimum radius value to (r_{min}) is:

$$C_r = 1 - min\Big(1, \frac{r}{r_{min}}\Big) = 0\,. \quad (12)$$

- Smoothness constraint
  G1 continuity can be imposed as:

$$C_s = normal(X_{t1}) \times normal(X_{t2}) = 0\,. \quad (13)$$

The *min* operator in Eq. 11 and Eq. 12 is used to 'activate' inequality constraint if it gets violated, and 'deactivate' it (constraint value is zero) otherwize. With differentiated OCCT we assemble derivatives of all constraint-functions into the constraint matrix:

$$C = \frac{\partial C_x}{\partial P} =$$

$$\begin{array}{cc} & \xrightarrow{\hspace{6em} T \hspace{6em}} \\ 3\times N \big\downarrow & \begin{pmatrix} \frac{\partial C_{d1}}{\partial P_1} & \frac{\partial C_{d2}}{\partial P_1} & \cdots & \frac{\partial C_{cr1}}{\partial P_1} & \frac{\partial C_{cr2}}{\partial P_1} & \cdots & \frac{\partial C_{cr}}{\partial P_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_{d1}}{\partial P_N} & \frac{\partial C_{d2}}{\partial P_N} & \cdots & \frac{\partial C_{cr1}}{\partial P_N} & \frac{\partial C_{cr2}}{\partial P_N} & \cdots & \frac{\partial C_{cr}}{\partial P_N} \end{pmatrix} \end{array}. \quad (14)$$

Here T correspond to the number of all above-mentioned constraints. Afterwards, the constraint matrix is used in the finite step update:

$$P^{n+1} = P^n + t \cdot Ker(C)\Big[(\nabla J)Ker(C)\Big]^T, \quad (15)$$

where

$$\nabla J = \frac{\partial J}{\partial X_S}\frac{\partial X_S}{\partial P}. \quad (16)$$

The last equation is equivalent to one step of a projected gradient method with a step size of t. Here $Ker(C)$ is the kernel of the constraint matrix. This ensures that the control point perturbations are in the null space of the constraint matrix i.e., control points are modified without violating the constraints (at least for infinitesimal step-size).

5.3. Constraint recovery

Due to non-linearity of constraints (G1, curvature) and inequality constraints (some constraints are inactive) they could be violated after the finite step - Eq. (15). To overcome this, we have extended the continuity recovery method proposed previously in [19] to all type of constraints.

First, by means of OCCT we indicate constraints that are indeed violated ($|\delta G_{d,r,s}| = |C_{d,r,s}| > \epsilon$) and input them into violation vector $\delta G_{violated} = (\delta G_1, \ldots, \delta G_{N_{violations}})$. We decompose constraint matrix into two matrices $C = C_{violated} \cup C_{satisfied}$ with columns entries corresponding to violated or satisfied constraints respectively. Afterwards, the necessary control point update could be defined:

$$C_{violated}\delta P_{upd} + \delta G_{violated} = 0, \tag{17}$$

which also have to satisfy the rest of the constraints:

$$\delta P_{upd} = Ker(C_{satisfied})\delta\alpha, \tag{18}$$

where α corresponds to the coefficients of linear combination of null space vectors. This could be further developed as:

$$\delta P_{upd} = -Ker(C_{satisfied})\left[C_{violated}Ker(C_{satisfied})\right]^{+} \delta G_{violated}. \tag{19}$$

Here, superscript $^+$ corresponds to the pseudoinverse of rectangular matrix and usually only few Newton steps are needed to recover constraints. The implications of this approach goes beyond shape optimization and could be applied directly on CAD shapes, which does not satisfy some certain requirements. We have used the TU Berlin Stator model with TE radius $r = 0.7$ and imposed minimum curvature/radius constraint there $r = 1$. This created constraints violations corresponding to every test-point located on the TE. Results of the recovery step with single Newton step are shown in Fig. 10, with all constraints satisfied for the updated red surface.

Figure 10. Recovery/Increase of TE radius from initial (grey) to updated (red)

6. Aerodynamic shape optimization of TU Berlin Stator

6.1. Optimization workflow

This subsection summarizes the main steps that we use for redesign of the TU Berlin Stator. The algorithm is generic and can be applied without major changes to any other aerodynamic shape optimization problem. To set up a new test case, one has to provide new parametrization and the corresponding CFD mesh. We refer to two aforementioned parametrization as a) for parametric and b) for BRep. Both of them could be used in two distinct optimization procedures:

1. Define parametrization and design surfaces.
2. Perform mesh point inversion (find mesh points X_S that belong to the design surfaces).
3. Run primal and adjoint CFD (get cost function value), compute CFD sensitivity: $\frac{dJ}{dX_S}$.
4. Compute CAD sensitivity $\frac{dX_S}{d\alpha}$ depending on the chosen parametrization:
   - 4.a) parametric approach: use differentiated OCCT to get sensitivities w.r.t the explicit design parameters (Sec. 4);
   - 4.b) NURBS-based approach: use differentiated OCCT to compute gradients w.r.t. the control points positions and to construct corresponding constraint matrix C (Sec. 5).
5. Compose total gradient $\frac{dJ}{d\alpha} = \frac{dJ}{dX_S}\frac{dX_S}{d\alpha}$.
6. Update design parameters using $\frac{dJ}{d\alpha}$, change CAD geometry and corresponding mesh:
   - 6.a) update design parameters using L-BFGS-B optimizer. Geometrical constraints are automatically satisfied and lie within prescribed bounding values (Eq. 7);
   - 6.b) update control points positions. Geometrical constraints are satisfied due to the projected

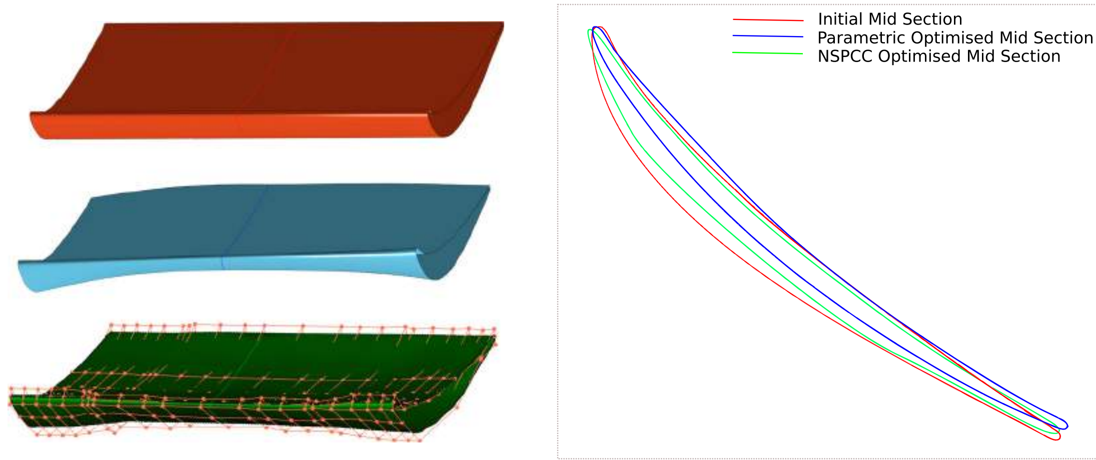


Figure 11. Left: Initial TU Stator (red), Optimized Parametric model (blue) and Optimized NURBS (green); Right: Comparison of mid-sections

gradient. Constraint matrix 'filter' ensures updates do not violate constraints (Eq. 15).

7. Repeat 3-6 until no further cost function improvement is possible.
8. Retrieve the optimized shape directly in the CAD format.

6.2. Optimization results

The main focus of this paper is to demonstrate a feasible approach to include a complex CAD model in a gradient-based optimization chain. At this stage we use low-fidelity CFD simulations, but without any limitation this allows to test and demonstrate the strength of both aforementioned CAD parametrizations in the design chain, however restrain us from the detailed discussions on the physics of the initial and optimized flow results. Therefore, we generated a coarse computational grid with ICEM CFD from the existing CAD model and used it for flow simulations in the STAMPS solver.

We perform two optimizations (explicit Parametric-based and implicit NURBS-based) to minimize the total pressure losses between the inlet and the outlet of the TU Berlin Stator. Two different optimizers were used, L-BFGS-B for the explicit and Steepest-Descent (with projected gradient) for the implicit parametrization. The corresponding optimized CAD models are shown in Fig. 11 together with the initial shape. In both cases we observe similar patterns of decrease of the leading edge and trailing edge radius and reduction of the blade thickness, while all geometrical constraints are satisfied. The optimized parametric and NURBS models improve the cost function by 14% and 13% respectively.

As highlighted in the comparison (Fig. 11) at the mid-section, the two optimized shapes are different, which originates from the differences in parametrizations and constraints. Judging by the CAD sensitivities (NURBS space generates very local sensitivities) the NUBRS-based approach could actually provide superior results. But contrary to the parametric model, it includes additional constraints for the mounting bolts. This results in a thicker blade towards the shroud and hub ends of the blade where the bolts are located. The imposition of identical constraints and use of high-fidelity CFD (increases the impact of CAD sensitivity locality) could enable further investigation of the occurring differences between two parametrizations.

7. Conclusions

We have successfully demonstrated the integration of a large-scale CAD system into the design chain for shape optimization of immersed bodies. The OCCT CAD kernel that is algorithmically differentiated to compute shape derivatives is the cornerstone ingredient: it provides the necessary CAD sensitivities efficiently, accurately and robustly. The approach allows to maintain CAD-models throughout the optimization loop thus enabling work in a multi-disciplinary framework. In addition to aerodynamic shape optimization, the coupling of the differentiated OCCT with structural analysis, conjugate heat transfer and robust design problems will be investigated in the future. The derivative information available in OCCT is also useful in a purely CAD context: (i) re-parametrization of the models (formulated as the optimization problem that tweaks parameters values to find the best 'fit' to the target geometry); (ii) recovery of the violated geometrical constraints.

Two different parametrization techniques for aerodynamic shape optimization of an industrial turbomachinery blade were proposed. For both of them the recipes for imposing manufacturing (geometrical) constraints were detailed. The storage of constraints in the standard CAD files and hence their visualization and inspection is possible for the NURBS-based approach. The choice of either of the parametrization for optimization of an arbitrary CAD-model is case-dependent, since both approaches perform design explorations in the different spaces. The parametric CAD models are useful for the applications when decent parametrizations are well established through the previous engineering experience. The NURBS-based approach could then serve as the complementary or the alternative technique which is advantageous for the optimization of non-conventional components.

Acknowledgements

This research is a part of the IODA project - Industrial Optimal Design using Adjoint CFD. IODA is *Marie Sklodowska-Curie Innovative Training Network*. This work was supported by European Commission [Grant Agreement Number 6429].

References

[1] Auriemma, S.; Banovic, M.; Mykhaskiv, O.; Legrand, H.; Müller, J.D.; Walther, A.: Optimization of a U-bend using CAD-based adjoint method with differentiated CAD kernel, ECCOMAS Congress, 2016. https://doi.org/10.7712/100016.2089.10065

[2] Banovic, M.; Mykhaskiv, O.; Auriemma, S.; Walther, A.; Legrand, H.; Müller, J. F.: Automatic Differentiation of the Open CASCADE Technology CAD System and its coupling with an Adjoint CFD Solver, http://www.optimization-online.org/DB_HTML/2017/03/5931.html

[3] De Boer, A.; Van der Schoot, M. S.; Bijl, H.: New method for mesh moving based on radial basis function interpolation, In ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, 2006.

[4] Dannenhoffer, J.; Haimes, R.: Design sensitivity calculations directly on CAD-based geometry. In 53rd AIAA Aerospace Sciences Meeting, 2015. https://doi.org/10.2514/6.2015-1370

[5] Economon, T. D.; Palacios, F.; Copeland, S. R.; Lukaczyk, T. W.; Alonso, J. J.: SU2: An Open-Source Suite for Multiphysics Simulation and Design, AIAA Journal, 2015, 54(3), 828–846.

[6] Gräsel, J.; Keskin, A.; Swoboda, M.; Przewozny, H.; Saxer, A.: A full parametric model for turbomachinery blade design and optimization, In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2004,907–914.

[7] Griewank, A.; Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd ed., Society for Industrial Mathematics, 2008. https://doi.org/10.1137/1.9780898717761

[8] Jameson, A.: Aerodynamic Design via Control Theory, Journal of Scientific Computing, 3, 1988, 233–260. https://doi.org/10.1007/BF01061285

[9] Jameson, A.; Vassberg, J. C.: Studies of alternative numerical optimization methods applied to the brachistochrone problem, Computational Fluid Dynamics Journal, 9(3), 2000, 281–296.

[10] Jones, D.; Müller, J. D.; Christakopoulos, F.: Preparation and assembly of discrete adjoint CFD codes, Computers & Fluids, 46(1), 2011, 282–286. https://doi.org/10.1016/j.compfluid.2011.01.042

[11] Piegl, L.; Tiller, W.: The NURBS book, Springer Science & Business Media, 2012.

[12] Pironneau, O.: On optimum design in fluid mechanics, Journal of Fluid Mechanics, 64, 1974, 97–100.

[13] Robinson, T. T; Armstrong, C.G.; Chua, H. S.; Othmer, C.; Grahs, T.: Optimizing parameterized CAD geometries using sensitivities based on adjoint functions, Computer-Aided Design and Applications, 9(3), 2012, 253–268. https://doi.org/10.3722/cadaps.2012.253-268

[14] Samareh, J.: Aerodynamic shape optimization based on free-form deformation, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004.

[15] Stück, A.: Adjoint Navier-Stokes methods for hydrodynamic shape optimization. Ph.D. Thesis, Technical University of Hamburg-Harburg, 2012.https://doi.org/10.2514/6.2004-4630

[16] Vasilopoulos, I. Flassig; Reiche, K.: 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.

[17] Verstraete, T.: CADO: a computer aided design and optimization tool for turbomachinery applications, In 2nd Int. Conf. on Engineering Optimization, 2010, 6–9.

[18] Walther, A.; Griewank, A.: Getting Started with ADOL-C, In Combinatorial scientific computing, Chapman & Hall/CRC Computational Science, Dagstuhl Seminar Proceedings, 2012, 181–202. https://doi.org/10.1201/b11644-8

[19] Xu, S.; Wolfram, J.; Müller, J. D.: CAD-based shape optimization with CFD using a discrete adjoint, International Journal for Numerical Methods in Fluids, 74(3), 2013, 153–168. https://doi.org/10.1002/fld.3844

[20] Xu, S.: CAD-based CFD shape optimization using discrete adjoint solvers, Ph.D. Thesis, Queen Mary University of London, 2015.

[21] Xu, S.; Radford, D.; Müller, J.-D.: CAD-based adjoint shape optimization of a one-stage turbine with geometric constraints, ASME Turbo Expo, 2015

[22] Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Transactions on Mathematical Software, 23, 1997, 550–560. https://doi.org/10.1145/279232.279236

[23] TU Berlin Stator test case: http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3