A Unified Geometry Parametrization Method for Turbomachinery Blades

Roberto Agromayor, Nitish Anand, Jens-Dominik Müller, Matteo Pini, Lars O. Nord

1. Department of Energy and Process Engineering, NTNU – The Norwegian University of Science and Technology, Kolbj. Hejes v. 18, NO-7491, Trondheim, Norway
2. Propulsion & Power, Aerospace Engineering Faculty, Delft University of Technology, Kluverweg 1, 2629 HS, Delft, The Netherlands
3. School of Engineering and Materials Science, Queen Mary University of London, Mile End Road, E1 4NS London, United Kingdom

A R T I C L E   I N F O

Article history:
Received 29 July 2020
Received in revised form 19 October 2020
Accepted 12 December 2020

Keywords:
Turbine
Compressor
Shape optimization
Gradient
CAD
NURBS

A B S T R A C T

Turbomachinery design is increasingly carried out by means of automated workflows based on high-fidelity physical models and optimization algorithms. The parametrization of the blade geometry is an essential aspect of such workflows because it defines the design space in which an optimal solution can be found. Currently, parametrization methods used for this purpose are often tailored to one particular type of turbomachinery blade, do not provide shape derivatives required for gradient-based optimization, or are not suited to re-parametrize a baseline blade geometry defined by a set of scattered point coordinates in a systematic way. This paper thus presents a general blade parametrization method for axial, radial, and mixed flow blades based on typical turbomachinery design variables and NURBS curves and surfaces. The shape derivatives are computed by means of the complex-step method, allowing the integration of the parametrization into gradient-based shape optimization workflows. In addition, the method enables the re-parametrization of a blade geometry defined by a cloud of points by solving a two-step optimization problem. The capabilities of the method are demonstrated by replicating eight blade geometries in two and three dimensions with an accuracy comparable to the tolerances of current manufacturing technologies.

1. Introduction

Driven by the ever-increasing requirements in performance, environmental impact, and life-time cost, turbomachinery design is increasingly carried out by means of automated workflows [1]. These workflows integrate geometry parametrization tools, high-fidelity physical models, and optimization algorithms to systematically explore the design space. The parametrization of the geometry is an essential aspect of the design chain because it defines the design space within which the optimization algorithm can find the optimal solution [2]. Ideally, a parametrization method for turbomachinery blades should:

1. Support any type of blade configuration and contain the shapes that achieve the required design objectives.
2. Allow the designer to impose geometric constraints due to mechanical or manufacturing requirements.
3. Provide the sensitivity of the shape with respect to the design variables to enable gradient-based shape optimization [3].
4. Use conventional engineering parameters with an intuitive geometrical meaning.
5. Produce smooth geometries with continuous curvature ($\mathbf{G}_2$ continuity) and continuous rate of change of curvature to avoid velocity spikes that may lead to flow separation [4].
6. Retain compatibility with Computer-Aided Design (CAD) software for further analysis, geometry manipulation, and manufacturing.
7. Be computationally cheap in terms of execution time and memory usage.

Shape parametrization methods can be classified into deformation and constructive methods. Deformation methods can be used to modify an existing geometry (a mesh or a CAD model) and are widely used in the context of turbomachinery shape optimization. These methods include mesh point displacement [5,6], CAD model control point displacement [7,8], superposition of shape functions such as Hicks–Henne bumps [9,10], and space morphing methods based on Free-Form Deformation (FFD) [11,12] or on Radial Basis Function (RBF) interpolation [13,14]. Although these methods enable the exploration of rich design spaces, they are not suited for an effective handling of geometric constraints, making it difficult to obtain feasible shapes out of the optimization process. As a notable exception, the NSPCC method [8]...
allows the designer to impose geometric and continuity constraints by evaluating these constraints at a finite number of test-points and using a projected gradient optimization algorithm to maintain feasibility. In contrast, constructive methods can be used to generate the geometry of a new blade from scratch, or possibly using design variable values obtained from preliminary design models such as mean-line [15,16] or throughflow models [17,18]. In addition, they allow one to impose geometric constraints such as minimum blade thickness in a natural and non-intrusive way. Due to these strengths, constructive methods are widely used for turbomachinery blade parametrization and a large number of such methods have been developed over the years. Table 1 provides a comprehensive review of constructive blade parametrizations up to the present day.

The early constructive parametrization methods used circular arcs and polynomials in monomial-basis form (that is, polynomials in the form \( \sum_{i=0}^{n} a_i x^i \) to define the geometry of the blades [19–24]. This type of parametrization gained significant popularity among industry practitioners, but it has severe limitations arising from the use of a monomial basis. Specifically: (1) the polynomial coefficients convey little insight about the shape of the blade, (2) ensuring geometric continuity at the connecting points between segments requires the solution of a linear system that may not have a unique solution, (3) the surface of the blade is prone to undesirable inflection points, and (4) the resulting shapes are not compatible with the geometric representation used by modern CAD systems.

To overcome these shortcomings, several authors proposed new constructive parametrizations based on Bézier [25–31], B-spline [32–37], and NURBS [38–42] curves and surfaces. These mathematical functions have become the standard to represent geometric objects in modern CAD packages due to their favorable mathematical properties and the availability of a wide range of algorithms to define and manipulate curves and surfaces [43,44]. Currently, most of the constructive CAD-based parametrizations for turbomachinery blades described in the open literature are not suitable for automated design workflows. This is because they do not offer a robust way to handle trimming and intersection operations [45,46] or do not provide sensitivity information required by gradient-based optimization algorithms [3]. In addition, to optimize an existing blade, it is essential to find a parametric representation of the baseline geometry, available, for instance, in the form of a large set of points in the Cartesian space. Solving this reverse engineering problem by trial and error is doable for simple cases [36,42], but it becomes impractical for complex blade geometries. Despite the practical relevance of this problem, a robust and automatic method to re-parametrize the geometry of a blade defined by a scattered set of points is still lacking.

In response to the limitations of the existing methods, this paper presents a general constructive parametrization method for axial, radial, and mixed-flow turbomachinery blades. The method exploits conventional engineering design variables (leading/trailing edge radius, metal angles, blade thickness, etc.) and NURBS curves and surfaces to represent the blade geometry. The method is formulated in an explicit way that avoids the use of intersection and trimming operations to define the geometry of the blade and flow domain and produces blades satisfying \( C^2 \) continuity by construction. The sensitivity of the geometry with respect to the design variables is computed with machine accuracy by means of the complex-step method [47–49]. In addition, the method is also adapted to re-parametrize the geometry of an existing blade defined by a scattered set of point coordinates. This problem, often referred to as blade matching, is formulated as a two-step optimization problem and it allows one to find the set of design variable values that best approximates the prescribed geometry in a systematic way. The flexibility and accuracy of the

### Nomenclature

#### Latin symbols

- \( c \): Chord length (m)
- \( c_{ax} \): Axial chord length (m)
- \( C(u) \): NURBS curve values (m)
- \( d \): Tangent proportion (-)
- \( f, g \): NURBS endpoint curvature functions (m)
- \( h \): Step size (-)
- \( J \): Objective function \((m^2)\)
- \( L \): Arc length (m)
- \( N_b \): Number of blades (-)
- \( N_0 \): Number of prescribed points (-)
- \( N_{i,p}, N_{i,q} \): B-spline basis polynomials (-)
- \( n \): Unitary normal vector (-)
- \( n, m \): NURBS highest indices (-)
- \( p, q \): NURBS degree (-)
- \( P_i, P_{ij} \): NURBS control point coordinates (m)
- \( Q_i \): Prescribed point coordinates (m)
- \( r \): Radius of curvature or radial direction (m)
- \( S(u, v) \): NURBS surface values (m)
- \( s \): Blade spacing, also known as pitch (m)
- \( t \): Thickness distribution (m)
- \( u, v \): NURBS parametric coordinates (-)
- \( \hat{u}, \hat{v} \): NURBS sample points (-)
- \( U, V \): NURBS knot vectors (-)
- \( w_i, w_{ij} \): NURBS control point weights (-)
- \( x, y, z \): Cartesian coordinates (m)

#### Greek symbols

- \( \alpha \): Set of design variables (-)
- \( \theta \): Metal angle or circumferential angle (°)
- \( \kappa \): Curvature \((m^{-1})\)
- \( \xi \): Stagger angle, also known as setting angle (°)
- \( \tau \): Unitary tangent vector (-)

#### Abbreviations

- AD: Algorithmic Differentiation
- B-spline: Basis Spline
- BFGS: Broyden–Fletcher–Goldfarb–Shanno
- C-FD: Central Finite Differences
- CAD: Computed-Aided Design
- CS: Complex-Step
- F-FD: Forward Finite Differences
- FD: Finite Differences
- FFD: Free Form Deformation
- NSPCC: NURBS-based Parametrization with Complex Constraints
- NURBS: Non Uniform Rational Basis Spline
- RBF: Radial Basis Functions

#### Subscripts and superscripts

- \( b \): Blade
- \( c \): Camber
- \( in, out \): Inlet and outlet
- \( l, u \): Lower and upper sides
- \( m \): Meridional
- \( u, v \): \( u \) \& \( v \) parametric directions
- \( 1, 2, 3, 4 \): Meridional channel edges
The proposed method is demonstrated by replicating the geometry of eight turbomachinery blades in two and three dimensions.

The rest of the paper is organized as follows. Section 2 documents the definition and properties of NURBS curves and surfaces. The blade parametrization in two and three dimensions is described in Sections 3 and 4, respectively, and the computation of the geometry sensitivity using the complex-step method is introduced and verified in Section 5. The blade matching method is presented and applied to replicate a wide range of blade geometries in Section 6. Finally, the software implementation of the method is described in Section 7 and the conclusions are summarized in Section 8.

2. Background on NURBS curves and surfaces

The origin of Non-Uniform Rational Basis Spline (NURBS) curves and surfaces can be traced back to the research efforts in computer-aided geometric design in the late 60s and early 70s [50]. Since then, NURBS curves and surfaces have been universally used for geometrical modeling thanks to their intuitive geometrical interpretation, favorable mathematical properties, and efficient computational algorithms. A NURBS curve, see Fig. 1(a), is a parametric curve defined by

\[
C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}, \quad 0 \leq u \leq 1, \tag{1}
\]

where \( p \) is the degree of the curve, the coefficients \( P_i \) and \( w_i \) are the coordinates and weights of the \( n+1 \) control points, and \( N_{i,p} \) are B-spline basis functions defined on the non-decreasing, clamped knot vector

\[
U = [0, \ldots, 0, u_{p+1}, \ldots, u_n, 1, \ldots, 1] \in \mathbb{R}^{p+1}
\]

with \( r = n + p + 1 \). The B-spline basis functions are given by the recursive relation

\[
N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{3}
\]

\[
N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{4}
\]

Similarly, a NURBS surface, see Fig. 1(b), is a parametric surface defined by

\[
S(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) w_{ij} P_{ij}}{\sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) w_{ij}}, \tag{5}
\]

with \( 0 \leq u, v \leq 1 \),

where \( p \) and \( q \) are the degrees of the surface in the \( u \)- and \( v \)-directions, the coefficients \( P_{ij} \) and \( w_{ij} \) are bidirectional nets containing the coordinates and weights of the \( (n+1) \times (m+1) \) control points, and \( N_{i,p}(u) N_{j,q}(v) \) are the product of univariate B-spline basis functions defined on the non-decreasing, clamped knot vectors

\[
U = [0, \ldots, 0, u_{p+1}, \ldots, u_n, 1, \ldots, 1] \in \mathbb{R}^{p+1}
\]

\[
V = [0, \ldots, 0, v_{q+1}, \ldots, v_n, 1, \ldots, 1] \in \mathbb{R}^{q+1},
\]

with \( r = n + p + 1 \) and \( s = m + q + 1 \). The \( u \)-direction basis functions \( N_{i,p}(u) \) are given by Eqs. (3) and (4), whereas the \( v \)-direction basis functions \( N_{j,q}(v) \) are defined in an analogous way replacing the variable \( u \) by \( v \) and the indices \( i \) and \( p \) by \( j \) and \( q \), respectively.

NURBS curves and surfaces have the following mathematical properties that make them particularly suited for geometric modeling [43, pp. 117–139]:

---

### Table 1

| Reference               | Basis function | Configuration | Cont. | Grad. |
|-------------------------|----------------|---------------|-------|-------|
| Dunham (1974)           | Monomial       | 2D profile    | G1    | n.a.  |
| Crouse (1981)           | Monomial       | 3D axial      | G1    | n.a.  |
| Ye (1984)               | Monomial       | 2D profile    | G1    | n.a.  |
| Pritchard (1985)        | Monomial       | 2D profile    | G1    | n.a.  |
| Korakianitis (1993)     | Monomial       | 2D profile    | G2    | n.a.  |
| Aungier (2006)          | Monomial       | 2D profile    | G2    | n.a.  |
| Engeli et al. (1974)    | Bézier         | 3D axial      | G2    | n.a.  |
| Casey (1983)            | Bézier         | 3D general    | G2    | n.a.  |
| Goel et al. (1996)      | Bézier         | 3D axial      | G2    | n.a.  |
| Giannakoglou (1999)     | Bézier         | 2D profile    | G1    | n.a.  |
| Trigg et al. (1999)     | Bézier         | 2D profile    | G1    | n.a.  |
| Pierret et al. (1999)   | Bézier         | 2D profile    | G2    | n.a.  |
| Pierre et al. (2000)    | Bézier         | 3D axial      | G2    | n.a.  |
| Oyama et al. (2004)     | B-spline       | 3D axial      | G1    | n.a.  |
| Hupertz et al. (2007)   | B-spline       | 2D profile    | G1    | n.a.  |
| Verstraete (2010)       | B-spline       | 3D axial      | G2    | n.a.  |
| Verstraete (2010)       | B-spline       | 3D general    | G2    | n.a.  |
| Siddappaji et al. (2012)| B-spline       | 3D general    | G2    | n.a.  |
| Torreguita et al. (2018)| B-spline       | 2D profile    | G1    | AD    |
| Mykhaskiv et al. (2018) | B-spline       | 3D axial      | G2    | AD    |
| Müller et al. (1996)    | NURBS (1996)   | 3D general    | G2    | n.a.  |
| Grzel et al. (2004)     | NURBS          | 3D general    | G2    | n.a.  |
| Koini et al. (2009)     | NURBS          | 3D general    | G2    | n.a.  |
| Müller et al. (2017)    | NURBS          | 3D general    | G2    | CS    |
| Anand et al. (2018)     | NURBS          | 2D profile    | G2    | FD    |
| Present work            | NURBS          | 2D/3D general | G2    | CS    |

* Slope continuity (G1) or curvature continuity (G2).
* Method used for gradient computation: not available (n.a.), Finite Differences (FD), Complex-Step (CS), or Algorithmic Differentiation (AD).
* The parametrization satisfies curvature continuity everywhere except at the trailing edge.
The upper and lower thickness distributions, $t^u(u)$ and $t^l(u)$, are given by B-spline polynomials of degree three with an arbitrary
number of control points, \( \{ t_u^u \} \) and \( \{ t_l^l \} \), specified by the user, see Fig. 2(c). The unitary vectors normal to the camber line \( \mathbf{n}(u) \) are computed from the unitary tangent vector \( \tau(u) \) according to

\[
\mathbf{n}(u) = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -\tau_y \\ \tau_x \end{bmatrix}, \quad \text{with} \quad \tau(u) = \frac{\mathbf{C}(u)}{\|\mathbf{C}(u)\|},
\]

(15)

where \( \mathbf{C}(u) \) is computed using analytical derivative formulas for B-spline curves \([43, \text{pp. 91–100}]\). The functions \( f(r) \) and \( g(r) \) appearing in Eqs. (12) and (13) are used to impose the radii of curvature \( r_{in} \) and \( r_{out} \) at the leading and trailing edges, ensuring that the parametrization satisfies \( C^2 \) continuity by construction, see Fig. 2(d). This feature is important for the aerodynamic design of turbomachinery blades because a sudden change in curvature could cause a spike in the surface pressure distribution or even a local separation bubble \([4]\). The functions \( f(r) \) and \( g(r) \) are based on the end point curvature formulas for NURBS curves and their derivation is detailed in the Appendix. Once that the upper and lower sides are defined, they can be combined into a single B-spline curve \( \mathbf{C}_b(u) = \mathbf{C}_l \cup \mathbf{C}_u \) that represents the entire blade profile.

When performing the assessment of the fluid-dynamic performance of the blades via computational fluid dynamics, it is necessary to define the geometry of the flow domain around the blade. For the majority of turbomachinery flow problems one can resort to the periodicity of the flow to reduce the size of the computational domain. In this case, it is therefore sufficient to describe the flow domain around a single blade, which is characterized by the inflow, outflow, and periodic boundaries, as illustrated in Fig. 3. The periodic boundaries are given by two cubic B-spline curves defined by extending the camber line while keeping zero slope at the inlet and outlet. The periodic boundaries
are located at an offset distance of half of the blade spacing, \( s \), with respect to the blade camber line. Finally, the inflow and outflow boundaries are defined as two straight lines connecting the upper and lower periodic boundaries.

The proposed parametrization produces blade profiles that have continuous curvature and rate of change of curvature, therefore reducing the possibility of flow separation [4]. This contrasts with most of the two-dimensional methods available in the open literature, which produce blades with discontinuous curvature [19,22,28,29,33], or rate of change of curvature [21,23,24,30]. As a notable exception, the second and third methods proposed by Korakianitis [23], see also [51,52], produce blade profiles with continuous curvature and slope-curvature. However, the methods proposed by Korakianitis involve the solution of systems of equations, are not compatible with CAD representations, and are not easily extended from two to three dimensions. In addition, to the knowledge of the authors, it is the first time that the endpoint curvature formulas for NURBS curves are used to impose the curvature of turbomachinery blades at the leading and trailing edges. This is different than what is documented in previous publications [34,37,42], where all the reported methods used the endpoint curvature formulas for Bézier curves to ensure \( G^2 \) continuity, with the limitation that the curvature is not imposed exactly when the blades are described by B-spline or NURBS curves.

### Application

The flexibility of the proposed two-dimensional blade parametrization method is demonstrated by reconstructing the four blade profiles illustrated in Fig. 4. Each blade profile was defined using 6 control points for each thickness distribution, resulting in a total of 22 design variables. The LS89 [53,54] and T106A [55] are representative of high-pressure and low-pressure axial gas turbine blade rows, respectively. In addition, the SIRT profile is typical of a supersonic impulse turbine rotor [56] and the STD10 profile is representative of an axial compressor blade derived from a NACA 0006 airfoil profile [57]. It can be observed that the parametrization method produces blades with smooth curvature variations, which is essential to avoid spikes and dips in the surface-pressure distribution. The numerical values of the design variables used to produce the blade profiles were computed from a set of scattered point coordinates using the method described in Section 6.

### 4. Blade parametrization in three dimensions

The proposed three-dimensional parametrization is formulated as an extension of the two-dimensional parametrization and uses the design variables listed in Table 3. Similar to the two-dimensional case, the parametrization starts by defining a camber surface and subsequently imposing two independent thickness distributions perpendicular to the camber surface in a way that ensures \( G^2 \) continuity.

The camber surface is determined by the shape of the blade in the meridional plane and the spanwise variation of the design variables. The shape of the blade in the meridional plane is described by four curves, namely, leading edge, trailing edge, hub, and shroud, as illustrated in Fig. 5. In contrast with other
parametrization methods that are limited to axial turbomachines [20,25,27,31,32], the proposed method is suited to describe any kind of turbomachinery configuration, including axial, radial, and mixed-flow machines. The number of control points required to describe the shape of the blade in the meridional plane depends on the complexity of the geometry. For instance, it is possible to define a purely axial turbine using only four control points, but it might be necessary to use 10–20 control points to describe a shape of a mixed-flow machine such as a centrifugal compressor.

The spanwise variation of the some design variables \( a(v) \), see Table 3 footnote, is defined as law of evolution through a B-spline of, at most, degree three with an arbitrary number of control points as illustrated in Fig. 7. The number of control points used for each design variable is specified by the user, and its selection is based on the complexity of the blade geometry. As an example, it is sufficient to use a single constant value to define a prismatic blade, but it might be necessary to use 3–6 control points to describe the geometry of a blade with large twist from the root to the tip.

As illustrated in Fig. 9(a), the camber surface \( S_i(u,v) \) is defined as a bi-quartic B-spline surface with control points \( P_{ij} = \{x_{ij}, y_{ij}, z_{ij}\} \). The coordinates of the control points are computed using the shape of the blade in the meridional plane and the spanwise evolution of the design variables. More specifically, the \((x, z)\) coordinates of the camber surface control points are computed by transfinite interpolation [58] of the four curves that define the meridional plane, see Fig. 6, and are given by

\[
\begin{align*}
\mathbf{x}(u,v) &= (1 - u) \cdot \mathbf{C}_1 m(u) + u \cdot \mathbf{C}_4 m(v) \\
\mathbf{z}(u,v) &= (1 - v) \cdot \mathbf{C}_2 m(u) + v \cdot \mathbf{C}_3 m(u) \\
&+ (1 - v)(1 - u) \cdot \mathbf{Q}_{1,2}^m - v u \cdot \mathbf{Q}_{1,4}^m \\
&- v(1 - u) \cdot \mathbf{Q}_{1,1}^m - (1 - v)u \cdot \mathbf{Q}_{3,3}^m.
\end{align*}
\]  

(16)

In addition, the \( y \) coordinates of the camber surface control points at each spanwise location \( v \) are given by a third order B-spline curve \( y_i(u,v) \) with control points \( \{y_{0,i}, y_{1,i}, y_{2,i}, y_{3,i}\} \) that are computed according to

\[
\begin{align*}
y_{0,i}(v) &= y_{in} \\
y_{1,i}(v) &= y_{0,i} + d_{in} \cdot L \tan \theta_{in} \\
y_{2,i}(v) &= y_{0,i} - d_{out} \cdot L \tan \theta_{out} \\
y_{3,i}(v) &= y_{in} + L \tan \xi
\end{align*}
\]  

(17-20)

This formulation ensures that the metal angles at the leading and trailing edges, \( \theta_{in} \) and \( \theta_{out} \), are respected, as illustrated in Fig. 8. The arc length of the blade meridional plane at each span location \( L(v) \) is defined as

\[
L(v) = \int_{u=0}^{u=1} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dz}{du}\right)^2} \, du
\]  

(21)

and it is computed using 8th order Gaussian quadrature [59], which provides a satisfactory trade-off between computational speed and accuracy.

The upper and lower sides of the blade, \( S_i(u,v) \) and \( S_i(u,v) \), are defined as B-spline surfaces of degree four as it is the lowest degree that guarantees continuous rate of change of curvature at the spline knots. The coordinates of the control points \( \{P_{ij}^m\} \) and \( \{P_{ij}^p\} \), see Fig. 9(b), are computed according to

\[
\begin{align*}
P_{ij}^m &= \begin{cases}
S_i(\tilde{u}_i, \tilde{v}_j), & \text{for } i = 0 \\
S_i(\tilde{u}_i, \tilde{v}_j) - \mathbf{n}(\tilde{u}_i, \tilde{v}_j) \cdot f(\tilde{r}_{in}(\tilde{v}_j)), & \text{for } i = 1 \\
S_i(\tilde{u}_i, \tilde{v}_j) - \mathbf{n}(\tilde{u}_i, \tilde{v}_j) \cdot f(\tilde{r}_{in}(\tilde{v}_j)), & \text{for } i = 2 : n - 2 \\
S_i(\tilde{u}_i, \tilde{v}_j) - \mathbf{n}(\tilde{u}_i, \tilde{v}_j) \cdot g(\tilde{r}_{out}(\tilde{v}_j)), & \text{for } i = n - 1 \\
S_i(\tilde{u}_i, \tilde{v}_j), & \text{for } i = n
\end{cases}
\]  

(22)
Design variable \( \alpha(v) \)

![Diagram](image)

**Fig. 7.** Spanwise variation of a design variable.

\[ n = -\frac{\tau_u \times \tau_v}{\| \tau_u \times \tau_v \|} \]

where the tangent vectors \( \tau_u \) and \( \tau_v \) are given by

\[ \tau_u = \frac{\partial S}{\partial u} \quad \text{and} \quad \tau_v = \frac{\partial S}{\partial v} \]

The partial derivatives of the camber surface with respect to \( u \) and \( v \) are computed analytically using B-spline surface derivative formulas [43, pp. 110–115]. In addition, the functions \( f(r) \) and \( g(r) \) appearing in Eqs. (22) and (23) are used to impose the radius of curvature at the leading and trailing edges, ensuring that the upper and lower surfaces of the blade are smoothly joined with \( G^2 \) continuity. The derivation of the functions \( f(r) \) and \( g(r) \) is detailed in the Appendix. Once that the upper and lower sides are defined, they can combined into a single B-spline surface \( S^i(u, v) = S^u \cup S^l \) that represents the entire blade.

The parametrization just described is suitable to model linear cascades, which are commonly used for wind tunnel tests. However, in actual turbomachines, the blades are arranged in an axisymmetric way forming an annular cascade. In order to achieve this, the linear blade configuration is transformed into an annular one with the mapping \( \mathcal{H} : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \) given by

\[ P^i_{\text{annular}} = \mathcal{H}(P^i_{\text{linear}}) = [x, z \cdot \sin(y/z), z \cdot \cos(y/z)] \]

The rationale behind this transformation is to associate the Cartesian coordinates \((x, y, z)\) of an annular cascade and then convert from cylindrical to Cartesian coordinates.

The flow domain around a blade is characterized by the hub, shroud, inlet, outlet, and periodic boundaries as illustrated in Fig. 10. The hub boundary consists of two surfaces that conform with the blade at its root, see Fig. 10(a). Each of these surfaces is defined as a Coons patch [43, pp. 456–507] that is characterized by four edges. The blade edge is given by a B-spline curve formed by extending the lower side of the blade into the upstream and downstream directions following the slope of the camber line at...
the leading and trailing edges, respectively. The periodic edge is formed by extending the camber line in a similar way and rotating the resulting B-spline about the x-axis through an angle \( \theta_b / 2 \), where \( \theta_b = 2\pi / N_b \). Finally, the inlet and outlet edges are defined as NURBS circular arcs that connect the periodic edge with the blade edge. The shroud surface, see Fig. 10(b), is defined in an analogous way and, for the case of rotor blades, it is possible specify a clearance between the tip of the blade and the shroud. Once that the hub and shroud surfaces are defined, it is straightforward to construct the inlet, outlet and periodic surfaces as ruled surfaces [43, pp. 337–340] that connect the limits of the hub and shroud surfaces as illustrated in Figs. 10(c) and 10(d). Note that the parametrization of the blade and flow domain is watertight by construction and it does not rely on intersection and trimming operations. This contrasts with other blade parametrization methods [34,35,37–41] that rely on intersection operations between the blade surface and the hub/shroud surfaces and produce trimmed NURBS patches that have to be specially treated during a shape optimization workflow [45,46].

5. Sensitivity computation and verification

One simple way to approximate partial derivatives of a function is by using a finite difference approximation such as forward finite differences given as

\[
\frac{\partial F}{\partial \alpha} = \frac{F(\alpha + h) - F(\alpha)}{h} + \mathcal{O}(h),
\]

or central finite differences given as

\[
\frac{\partial F}{\partial \alpha} = \frac{F(\alpha + h) - F(\alpha - h)}{2h} + \mathcal{O}(h^2),
\]

where \( F(\alpha) \) can be identified with \( C^2(u, \alpha) \) in two dimensions or \( S^0(u, v, \alpha) \) in three dimensions and \( h \) is the step size used for finite difference computation. Finite difference approximations are susceptible to cancellation error when the step size is small because of the subtraction of very similar numbers in the numerator [66, pp 229–232]. As a result, one is faced with the dilemma of selecting a small step size that minimizes the truncation error but does not lead to a large cancellation error.
An alternative method that avoids the occurrence of cancellation error is the complex-step method [47–49]. This method uses a complex argument to compute the first derivative of a real-valued function. Indeed, the Taylor series expansion of $F(\alpha)$ in the imaginary axis gives

$$F(\alpha + ih) = F(\alpha) + ih \frac{\partial F}{\partial \alpha} + \frac{h^2}{2!} \frac{\partial^2 F}{\partial \alpha^2} + \mathcal{O}(ih^3).$$  \hspace{1cm} (31)$$

Re-arranging the imaginary part of the equation leads to

$$\frac{\partial F}{\partial \alpha} = \frac{\text{Im}(F(\alpha + ih))}{h} + \mathcal{O}(h^2),$$  \hspace{1cm} (32)$$

which is the complex-step method formula. In contrast to finite difference approximations, this method is not susceptible to subtraction error, allowing one to compute first derivatives accurate to the round-off precision of floating point arithmetic by using an arbitrarily small step size.

Algorithmic Differentiation (AD) provides yet another alternative to compute the derivatives of a function with machine precision [67]. AD is a set of techniques to numerically evaluate the derivatives of a function specified as a computer program by applying the chain rule of differentiation to each arithmetic operation of the program. This method offers more functionality and computational efficiency (first and higher order derivatives, forward and reverse modes) than the complex-step method (first derivatives and forward mode only), but it is also more difficult to implement [49].

In this work, the complex-step method was adopted to compute the sensitivity of the surface coordinates with respect to the design variables due to its accuracy, simplicity, and ease of implementation. Fig. 12 illustrates the sensitivity of the blade surface with respect to one thickness distribution control point in two and three dimensions. It can be observed that the sensitivity of the blade changes from point to point and that there may be regions where the sensitivity is zero as a result of the compact-support property of NURBS basis functions [43, p.118].

To verify the correctness of the sensitivity computation the authors performed a convergence study comparing the sensitivities computed using forward finite differences, central finite differences, and the complex-step method for the NASA R67 test case [61,62]. The geometry of the NASA R67 rotor, see Fig. 11, was sampled with 10000 uniformly spaced points within the box $(u, v) \in [0, 1] \times [0, 1]$ and the sensitivity was computed with respect to one design variable (the stagger angle at the hub) for different step sizes in the interval $h \in [10^{-1}, 10^{-15}]$. The error of the sensitivity computation was defined as the mean-square-root deviation between the exact and the estimated sensitivities. The exact sensitivity was assumed to be that computed with the complex-step method using a step size $h = 2.22 \cdot 10^{-16}$, which corresponds to the machine epsilon of double-precision arithmetic [66, pp 8–11].

The results of the convergence study are shown in Fig. 13. For the case of the complex-step method (CS), reducing the step size decreases the error until the trend flattens to a value close to the machine precision. In contrast, the forward finite difference (F-FD) and central finite difference (C-FD) errors decrease as the step size decreases down to a minimum value and then increase because the cancellation error becomes more prominent than the...
subject to $0 \leq \alpha \leq 1$.

In two dimensions and by

$$\min_{\alpha \in \mathbb{R}^N} J(\alpha) = \sum_{i=1}^{N_3} \|C^i(u_i, \alpha) - Q\|_2^2$$

(37)

in three dimensions, where $\alpha$ are the design variables listed in Tables 2 and 3, respectively. The gradient of these objective functions is computed using the complex-step method as described in Section 5. In contrast to the point projection phase, the geometry of the parametrized blade is updated until the deviation with respect to the prescribed geometry is minimized, see Fig. 14(c). In order to improve the matching of $(u, v)$ and $\alpha$, the point projection and geometry update problems are solved alternatively until the relative deviation between the prescribed and the parametrized blades does not change more than a small tolerance, e.g. $10^{-8}$.

To demonstrate its flexibility and accuracy, the blade matching method was applied to replicate the geometry of eight exemplary blades. The set of test cases is summarized in Table 4 and it

### Table 4
Summary of the test cases and matching results.

| Name and reference | Description | Dim. | DVs | Error abs. (mm) | Error rel. (%) |
|-------------------|-------------|------|-----|----------------|---------------|
| LS89 [53,54]      | High-stagger reaction turbine profile | 2D   | 22  | 0.047          | 0.067         |
| T106A [55]        | High-deflection reaction turbine profile | 2D   | 22  | 0.057          | 0.046         |
| SITR [56]         | Supersonic impulse turbine profile | 2D   | 22  | 0.067          | 0.087         |
| STD10 [57]        | Slender compressor profile | 2D   | 22  | 0.020          | 0.020         |
| AACHEN [60]       | Axial flow stator blade | 3D   | 26  | 0.060          | 0.084         |
| NASA R67 [61,62]  | High-twist turbo fan blade | 3D   | 99  | 0.107          | 0.107         |
| XPROP [63]        | High-twist propeller blade | 3D   | 113 | 0.127          | 0.377         |
| APU [64,65]       | Mixed-flow turbine rotor blade | 3D   | 86  | 0.057          | 0.080         |

*Defined as the arithmetic mean deviation between the prescribed and the matched blades.

*Defined as the quotient of the mean error and the arc length of blade camber line (three-dimensional cases use the camber line at the hub).
was conceived to cover a wide range of turbomachinery blade geometries in two (Fig. 4) and three dimensions (Fig. 11). The results of the blade matching in terms of absolute and relative error are summarized in Table 4.

The relative matching error is below 0.38% for all cases and the highest absolute error is 0.127 mm for the XPROP test case, which is of the same order of magnitude as the tolerances used to manufacture axial gas turbine blades ($\approx0.05$ mm) [68]. In addition, the deviation could be further reduced by increasing the number of control points used to parametrize the blade. This is illustrated in the convergence study shown in Fig. 15, where the number of control points used to describe the thickness distribution of the T106A case is increased from 3 to 10 points. Specifically, the mean deviation is reduced from 0.057 mm to 0.031 mm when the number of control points is increased from 6 to 10. In addition, Fig. 16 shows the curvature distribution for the T106A blade described using 6 and 10 thickness distribution control points. It can be observed that the curvature variation is smooth for both cases and that increasing the number of control points does not introduce high-frequency undulations that would deteriorate the fluid dynamic performance of the blade. These results indicate that the parametrization and matching methodologies proposed in this work enable the replication of a wide range of geometries with an accuracy comparable to the typical tolerances of modern manufacturing techniques and that the re-parametrization accuracy can be increased by refining the design space.

7. Software structure

The blade parametrization method proposed in this work was implemented in the Python programming language [69] and released under a permissive open source license as the ParaBlade software package [70]. The code was written using object-oriented programming principles and the structure of the package is subdivided in the classes shown in Fig. 17. The implementation integrates the in-house NurbsPy package [71] to define and manipulate NURBS curves and surfaces. In addition, ParaBlade relies on the NumPy library [72] for array operations and on the Pagmo/Pygmo optimization library [73] to solve the blade matching optimization problems by means of the limited-memory BFGS algorithm [74,75].

8. Conclusions

This paper presented a general constructive parametrization method for turbomachinery blades. The method uses typical turbomachinery design variables and NURBS curves and surfaces to produce blade geometries with continuous curvature and rate of change of curvature. In contrast with existing methods, the flow domain parametrization was formulated in an explicit way that avoids intersection and trimming operations and the sensitivity of the geometry is computed by means of the complex-step method, allowing the integration of the parametrization into automated, gradient-based shape optimization workflows.

In addition, the method enables the re-parametrization of a baseline blade geometry defined by a set of scattered point coordinates in a systematic way by solving a two-step optimization
problem. To demonstrate its capabilities, the re-parametrization method was applied to replicate the geometry of eight exemplary blades, showing that the proposed parametrization can replicate the geometry of a wide range of turbomachinery blades with an accuracy comparable to the tolerances of current manufacturing techniques for axial gas turbine profiles.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the financial support from the Research Council of Norway (EnergiX grant number 14711) and the user partners Equinor, Hydro, Alcoa, GE Power Norway and FrioNordica.

Appendix. Derivation of the endpoint curvature formulas

This appendix contains the derivation of the functions $f(r_{in})$ and $g(r_{out})$ that are used to impose the radii of curvature $r_{in}$ and $r_{out}$ at the leading and trailing edges. According to Goldman [76], the curvature of a parametric curve $C(u)$ is given by

$$
\kappa(u) = \frac{\|\hat{C}(u) \times \ddot{C}(u)\|}{\|\hat{C}(u)\|^3}.
$$

(A.1)

In addition, it can be shown [43, pp. 125–127], that the first and second derivatives of a NURBS curve at its start point are given by

$$
\hat{C}(u=0) = \left( \frac{p}{u_{p+1}} \right) \left( \frac{w_1}{w_0} \right) (P_1 - P_0)
$$

(A.2)
where the plus and minus signs correspond to the upper and lower sides of the blade, respectively, and the function \( g(r_{\text{out}}) \) is given by

\[
g(r_{\text{out}})^2 = r_{\text{out}} \left( \frac{p - 1}{p} \right) \left( 1 - u_n \right) \left( \frac{u_n}{w_n - u_n} \right) \left( \frac{w_{n-2}}{w_{n-1} - w_{n-2}} \right) \left\| \left( \mathbf{P}_{n-2} - \mathbf{P}_n \right) \times \mathbf{n} \right\|.
\]

(A.12)

Solving for \( \left\| \mathbf{P}_{n-1} - \mathbf{P}_n \right\| \), the location of the control point \( \mathbf{P}_{n-1} \) that guarantees that the radius of curvature at the trailing edge is \( r_{\text{out}} \) is given by

\[
\mathbf{P}_{n-1} = \mathbf{P}_0 \pm \left\| \mathbf{P}_{n-1} - \mathbf{P}_n \right\| \cdot \mathbf{n} = \mathbf{P}_0 \pm g(r_{\text{out}}) \cdot \mathbf{n}.
\]

(A.13)

Note that this construction guarantees that the blade is \( C^2 \) continuous at the leading and trailing edges since the radius of curvature is the same at the points connecting the upper and lower sides of the blade.

References

[1] Verstreeta T. Toward gradient-based optimization of full gas turbines. Mech Eng 2019;141(03):54–5. http://dx.doi.org/10.1115/1.4035833.
[2] Samareh JA. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. AIAA J 2001;39(5):877–84. http://dx.doi.org/10.2514/2.1391.
[3] Banovc M, Mykhaskin O, Auriemma S, Walther A, Legrand H, Muller J-D. Algorithmic differentiation of the open CASCADE technology CAD kernel and its coupling with an adjoint CFD solver. Optim Methods Softw 2018;33(4-6):813–28. http://dx.doi.org/10.1080/10556788.2018.1431235.
[4] Korakianitis T, Papagiannidis P. Surface-curvature-distribution effects on turbine-cascade performance. Trans ASME, J Turbomach 1993;115(2):334–41. http://dx.doi.org/10.1115/1.2929239.
[5] Wu H-Y, Liu F, Tsai H-M. Aerodynamic design of turbine blades using an adjoint equation method. In: 43rd AIAA aerospace sciences meeting and exhibit. 2005, p. 1–13. http://dx.doi.org/10.2514/6.2005-1006.
[6] Walther B, Nadarajah S. Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. Trans ASME, J Turbomach 2012;135(2):1–10. http://dx.doi.org/10.1115/1.4007502.
[7] Pini M, Persico G,Pasquale D, Rebay S. Adjoint method for shape optimization in real-gas flow applications. Trans ASME, J Eng Gas Turbines Power 2014;137(3):1–13. http://dx.doi.org/10.1115/1.4028495.
[8] Xu S, Radford D, Meyer M, Muller J-D. CAD-based adjoint shape optimization of a one-stage turbine with geometric constraints. In: ASME turbo expo: Power for land, sea, and air. 2015, p. 1–14. http://dx.doi.org/10.1115/GT2015-42237.
[9] Luo J, Zhou C, Liu F. Multipoint design optimization of a transonic compressor blade by using an adjoint method. Trans ASME, J Turbomach 2013;135(5):1–10. http://dx.doi.org/10.1115/1.4025164.
[10] Walther B, Nadarajah S. Optimum shape design for multirow turbomachinery configurations using a discrete adjoint approach and an efficient radial basis function deformation scheme for complex multiblock grids. Trans ASME, J Turbomach 2015;137(8):1–20. http://dx.doi.org/10.1115/1.4029550.
[11] Vitale S, Albring TA, Pini M, Gauger NR, Colonna P. Fully turbulent discrete adjoint solver for non-ideal compressible flow applications. J Glob Power Propuls Soc 2017;1:252–70. http://dx.doi.org/10.22261/JGPPS.Z1FV0L.
[12] John A, Shahpar S, Qin N. Novel compressor blade shaping through a free-form method. Trans ASME, J Turbomach 2017;139(8):1–11. http://dx.doi.org/10.1115/1.4035833.
R. Agromayor, N. Anand, J.-D. Müller et al.

Computer-Aided Design 133 (2021) 102987

[38] Tang X, Luo J, Liu F. Adjoint aerodynamic optimization of a transonic fan rotor blade with a localized two-level mesh deformation method. Aerosp Sci Technol 2011;17:267–77. http://dx.doi.org/10.1016/j.ast.2011.04.003.

[40] Gagliardi F, Giannakoglou KC. RBF-based morphing of B-Rep models for use in aerodynamic shape optimization. In: ASME Turbo-soc 2019, pp. 1–13. http://dx.doi.org/10.1115/2019-VGT-102724.

[42] Anand N, Vitale S, Pini M, Colonna P. Assessment of FFD and CAD-based shape parameterization methods for adjoint-based turbomachinery blade optimization. In: Proceedings of the Montreal 2018 global power and propulsion forum. 2018, p. 1–8. http://dx.doi.org/10.1115/2018-GT-02270.

[44] Arts T, LambertdeRouvroit M, Rutherford AW. Aero-Thermal investigation of a highly loaded transonic linearturbine guidevane cascade: A testcase for inviscid and viscousflowcomputations. Tech. rep. 174, VKI Training Center for Experimental Aerodynamics; 1990, p. 1–97. http://dx.doi.org/10.2514/6.2015-3408.

[46] Xu S, Timme S, Mykhaskiv O, Müller J-D. Wing-body junction optimization with CAD-based parametrisation including a moving intersection. Aerosp Sci Technol 2017;68:54–66. http://dx.doi.org/10.1016/j.ast.2017.06.014.

[48] Koini GN, Sarakinos SS, Nikolos IK. A software tool for parametric design of turbomachinery blades. Adv Eng Softw 2009;40(1):41–51. http://dx.doi.org/10.1016/j.advengsoft.2008.03.008.

[50] Cartwright RM, Randles A, Carpenter B. The use of CAD-based parametrisation including a moving intersection. Aerosp Sci Technol 2017;68:54–66. http://dx.doi.org/10.1016/j.ast.2017.06.014.

[52] Martin JS, Kuo Y, Achenbach T. A two-dimensional Navier-Stokes solver and artificial neural network. Trans ASME, J Eng Power 1983;105(2):288–95. http://dx.doi.org/10.1115/83-GT-07.

[54] Anand N, Rubino A, Colonna P. Improved turbine-blade design techniques using 4th-order parametric spline segments. Comput-Aided Des 2013;45(5):86–98. http://dx.doi.org/10.1016/j.cad.2012.10.014.

[56] Anand N, Rubino A, Colonna P, Pini M. Adjacent-based aerodynamic design optimization using a harmonic balance method. In: ASME turbo expo: Turbomachinery technical conference and exhibition. American Society of Mechanical Engineers Digital Collection; 20011, p. 1–8. http://dx.doi.org/10.1115/2001-GT-0311.

[58] Abgrall R, Schneider D, Viozzi D. A high-order upwind scheme for unsteady Navier-Stokes flow calculations. Comput Fluids 2012;60:86–98. http://dx.doi.org/10.1016/j.compfluid.2012.03.001.

[60] Stephan B, Gallus HE, Niehuis R. Experimental investigation of the 2-stage turbomachinery. In: ASME turbo expo: Power for land, sea, and air. 1996, p. 1–10. http://dx.doi.org/10.1115/2000-GT-0613.
[61] Pierzga MJ, Wood JR. Investigation of the three-dimensional flow field within a transonic fan rotor: Experiment and analysis. Trans ASME, J Eng Gas Turbines Power 1985;107(2):436–48. http://dx.doi.org/10.1115/1.3239744.

[62] Pierzga MJ, Wood JR. Laser anemometer measurements in a transonic axial-flow fan rotor. Tech. rep. TP 2879, NASA Lewis Research Center; Cleveland, OH, United States; 1989. p. 1–216.

[63] van Arnhem N, de Vries R, Sinnige T, Vos R, Etelberg G, Veldhuis L. Engineering method to estimate the blade loading of propellers in nonuniform flow. AIAA J 2020;58(12):5332–46. http://dx.doi.org/10.2514/1.J059485.

[64] Jones AC. Design and test of a small, high pressure ratio radial turbine. Trans ASME, J Turbomach 1996;118(2):362–70. http://dx.doi.org/10.1115/1.2836651.

[65] Keep JA, Vitale S, Pini M, Burigana M. Preliminary verification of the open-source CFD solver SU2 for radial-inflow turbine applications. Energy Procedia 2017;129:1071–7. http://dx.doi.org/10.1016/j.egypro.2017.09.130.

[66] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes: The art of scientific computing. 3rd ed.. Cambridge University Press; 2007.

[67] Griewank A, Walther A. Evaluating derivatives: Principles and techniques of algorithmic differentiation. 1st ed.. SIAM; 2008.

[68] Bunker RS. The effects of manufacturing tolerances on gas turbine cooling. Trans ASME, J Turbomach 2009;131(4):1–11. http://dx.doi.org/10.1115/1.3072494.

[69] Python v3.7. 2020. https://www.python.org/.

[70] Parablude v1.0. 2020. http://dx.doi.org/10.5281/zenodo.3894778.

[71] Nurbspy v1.1. 2020. http://dx.doi.org/10.5281/zenodo.3894755.

[72] NumPy v1.18. 2020. https://numpy.org/.

[73] Pagmo/Pymoo v2. 2020. http://dx.doi.org/10.5281/zenodo.3738182.

[74] Nocedal J. Updating quasi-Newton matrices with limited storage. Math Comp 1980;35(151):773–82. http://dx.doi.org/10.2307/2006193.

[75] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program 1989;45(1–3):503–28. http://dx.doi.org/10.1007/BF01589016.

[76] Goldman R. Curvature formulas for implicit curves and surfaces. Comput Aided Geom Design 2005;22(7):632–58. http://dx.doi.org/10.1016/j.cagd.2005.06.005.