

A Unified Geometry Parametrization Method for Turbomachinery Blades

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

DOI

[10.1016/j.cad.2020.102987](https://doi.org/10.1016/j.cad.2020.102987)

Publication date

2021

Document Version

Final published version

Published in

CAD Computer Aided Design

Citation (APA)

Agromayor, R., Anand, N., Müller, J. D., Pini, M., & Nord, L. O. (2021). A Unified Geometry Parametrization Method for Turbomachinery Blades. *CAD Computer Aided Design*, 133, Article 102987. <https://doi.org/10.1016/j.cad.2020.102987>

Important note

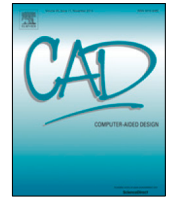
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Application paper

A Unified Geometry Parametrization Method for Turbomachinery Blades[☆]Roberto Agromayor^{a,*}, Nitish Anand^b, Jens-Dominik Müller^c, Matteo Pini^b, Lars O. Nord^a^a Department of Energy and Process Engineering, NTNU – The Norwegian University of Science and Technology, Kolbj. Hejes v. 1B, NO-7491, Trondheim, Norway^b Propulsion & Power, Aerospace Engineering Faculty, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, The Netherlands^c School of Engineering and Materials Science, Queen Mary University of London, Mile End Road, E1 4NS London, United Kingdom

ARTICLE INFO

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

ABSTRACT

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.

© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

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 (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]

[☆] This paper has been recommended for acceptance by Xiaoping Qian.

* Corresponding author.

E-mail address: roberto.agromayor@ntnu.no (R. Agromayor).

Nomenclature

Latin symbols

c	Chord length (m)
c_{ax}	Axial chord length (m)
$\mathbf{C}(u)$	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_Q	Number of prescribed points (-)
$N_{i,p}, N_{j,q}$	B-spline basis polynomials (-)
$\mathbf{n}$	Unitary normal vector (-)
n, m	NURBS highest indices (-)
p, q	NURBS degree (-)
$\mathbf{P}_i, \mathbf{P}_{ij}$	NURBS control point coordinates (m)
$\mathbf{Q}_j$	Prescribed point coordinates (m)
r	Radius of curvature or radial direction (m)
$\mathbf{S}(u, v)$	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 (-)
$w_i, w_{i,j}$	NURBS control point weights (-)
x, y, z	Cartesian coordinates (m)

Greek symbols

α	Set of design variables (-)
θ	Metal angle or circumferential angle ($^\circ$)
κ	Curvature (m^{-1})
ξ	Stagger angle, also known as setting angle ($^\circ$)
τ	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 and v parametric directions
1, 2, 3, 4	Meridional channel edges

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 G^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

Table 1

Review of constructive blade parametrization methods documented in the open literature.

Reference		Basis function	Configuration	Cont. ^a	Grad. ^b
Dunham (1974)	[19]	Monomial	2D profile	G ¹	n.a.
Crouse (1981)	[20]	Monomial	3D axial	G ¹	n.a.
Ye (1984)	[21]	Monomial	2D profile	G ²	n.a.
Pritchard (1985)	[22]	Monomial	2D profile	G ¹	n.a.
Korakianitis (1993)	[23]	Monomial	2D profile	G ²	n.a.
Aungier (2006)	[24]	Monomial	2D profile	G ²	n.a.
Engeli et al. (1974)	[25]	Bézier	3D axial	G ²	n.a.
Casey (1983)	[26]	Bézier	3D general	G ²	n.a.
Goel et al. (1996)	[27]	Bézier	3D axial	G ²	n.a.
Giannakoglou (1999)	[28]	Bézier	2D profile	G ¹	n.a.
Trigg et al. (1999)	[29]	Bézier	2D profile	G ¹	n.a.
Pierret et al. (1999)	[30]	Bézier	2D profile	G ^{2c}	n.a.
Pierret et al. (2000)	[31]	Bézier	3D axial	G ^{2c}	n.a.
Oyama et al. (2004)	[32]	B-spline	3D axial	G ¹	n.a.
Huppertz et al. (2007)	[33]	B-spline	2D profile	G ¹	n.a.
Verstraete (2010)	[34]	B-spline	3D axial	G ^{2c}	n.a.
Verstraete (2010)	[34]	B-spline	3D general	G ^{2c}	n.a.
Siddappaji et al. (2012)	[35]	B-spline	3D general	G ²	n.a.
Torreguitart et al. (2018)	[36]	B-spline	2D profile	G ^{2c}	AD
Mykhaskiv et al. (2018)	[37]	B-spline	3D axial	G ²	AD
Miller et al. (1996)	[38]	NURBS	3D general	G ²	n.a.
Gräsel et al. (2004)	[39]	NURBS	3D general	G ²	n.a.
Koini et al. (2009)	[40]	NURBS	3D general	G ²	n.a.
Müller et al. (2017)	[41]	NURBS	3D general	G ^{2c}	CS
Anand et al. (2018)	[42]	NURBS	2D profile	G ²	FD
Present work		NURBS	2D/3D general	G ²	CS

^aSlope continuity (G¹) or curvature continuity (G²).^bMethod used for gradient computation: not available (n.a.), Finite Differences (FD), Complex-Step (CS), or Algorithmic Differentiation (AD).^cThe parametrization satisfies curvature continuity everywhere except at the trailing edge.

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

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) w_i}, \quad \text{with } 0 \leq u \leq 1, \quad (1)$$

where p is the degree of the curve, the coefficients $\mathbf{P}_i$ 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

$$U = [\underbrace{0, \dots, 0}_{p+1}, \underbrace{u_{p+1}, \dots, u_n}_{n-p}, \underbrace{1, \dots, 1}_{p+1}] \in \mathbb{R}^{r+1} \quad (2)$$

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

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

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (4)$$

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

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij}}, \quad (5)$$

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 $\mathbf{P}_{ij}$ 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

$$U = [\underbrace{0, \dots, 0}_{p+1}, \underbrace{u_{p+1}, \dots, u_n}_{n-p}, \underbrace{1, \dots, 1}_{p+1}] \in \mathbb{R}^{r+1} \quad (6)$$

$$V = [\underbrace{0, \dots, 0}_{q+1}, \underbrace{v_{q+1}, \dots, v_m}_{m-q}, \underbrace{1, \dots, 1}_{q+1}] \in \mathbb{R}^{s+1}, \quad (7)$$

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]:

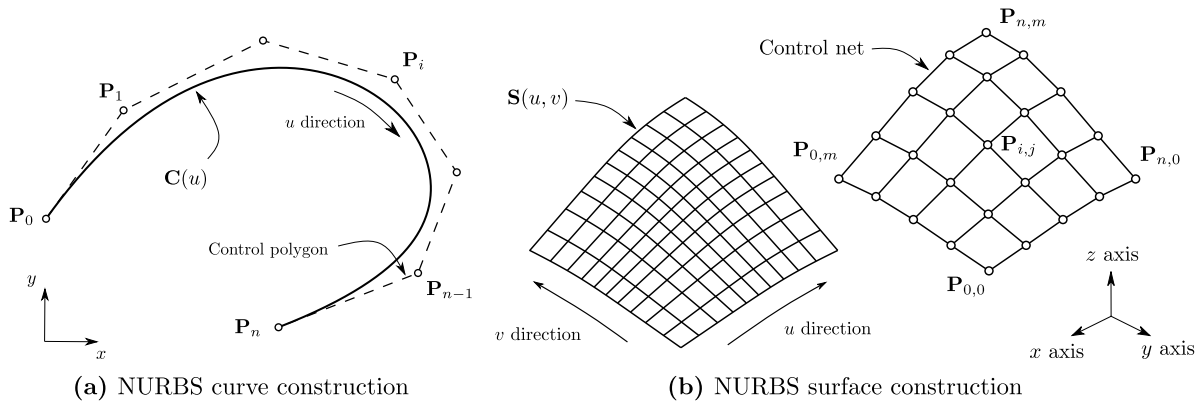


Fig. 1. Construction of a NURBS curve (left) and surface (right). Note that the NURBS curve interpolates its endpoints and it is tangent to the control polygon at its ends. The control net of the NURBS surface interpolates its four corner points and it was represented at an offset distance in the x -direction for clarity.

Table 2

Two-dimensional design variables. Each design variable is provided as a scalar value, except for the upper and lower thickness that are given as sets of control points.

Variable name	Symbol
Spacing	s
Leading edge abscissa and ordinate	x_{in}, y_{in}
Axial chord length	c_{ax}
Stagger angle	ξ
Inlet and exit metal angles	$\theta_{in}, \theta_{out}$
Inlet and exit tangent proportions	d_{in}, d_{out}
Inlet and exit radii of curvature	r_{in}, r_{out}
Upper and lower thickness distributions	t^u, t^l

- **Affine invariance.** It is possible to apply affine transformations such as rotations, displacements, and scalings to NURBS curves and surfaces by applying the transformation to their control points.
- **Convex hull.** NURBS curves and surfaces are within the convex hull of their control points. When the control points are contained in a certain region of space, this property guarantees that the curve or surface will not blow up arbitrarily far away from this region.
- **Endpoint interpolation.** NURBS curves and surfaces coincide with the polytope formed by the control points at the endpoints.
- **Endpoint tangency.** NURBS curves and surfaces are tangent to the polytope formed by the control points at the endpoints.
- **Generalization.** Bézier curves and surfaces are a special case of NURBS when $p = n$ and $q = m$. In addition, B-spline curves and surfaces are a special case of NURBS when all the weights have the same value.

Most of the curves and surfaces used in the proposed blade parametrization method are B-splines. However, the parametrization is formulated in a general way using NURBS so that the user can include the control point weights as design variables to gain more control over the resulting geometry.

3. Blade parametrization in two dimensions

The proposed two-dimensional blade parametrization is based on typical blade design variables which are listed in Table 2. The geometry of the blade is generated by defining a camber line and subsequently imposing on it two independent thickness distributions in a way that ensures G^2 continuity at the junction between the upper and the lower sides.

The camber line $C^c(u)$ is a cubic B-spline curve defined by four control points as shown in Fig. 2(a). The coordinates of the control points are given by

$$\mathbf{P}_0^c = \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}, \quad (8)$$

$$\mathbf{P}_1^c = \mathbf{P}_0^c + d_{in} \begin{bmatrix} c \cos(\theta_{in}) \\ c \sin(\theta_{in}) \end{bmatrix}, \quad (9)$$

$$\mathbf{P}_2^c = \mathbf{P}_3^c - d_{out} \begin{bmatrix} c \cos(\theta_{out}) \\ c \sin(\theta_{out}) \end{bmatrix}, \quad (10)$$

$$\mathbf{P}_3^c = \mathbf{P}_0^c + \begin{bmatrix} c \cos(\xi) \\ c \sin(\xi) \end{bmatrix}, \quad (11)$$

where ξ is the stagger angle, $c_{ax} = c \cos(\xi)$ is the axial chord length, θ_{in} and θ_{out} are the inlet and outlet metal angles, and d_{in} and d_{out} are the inlet and outlet tangent proportions. This construction of the camber line ensures that the blade has the specified axial chord length and that the slope at the leading and trailing edges agrees with the input metal angles thanks to the endpoint tangency property of B-spline curves [43, p. 97].

The upper and lower sides of the blade, $C^l(u)$ and $C^u(u)$, are defined as B-spline curves 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 $\{\mathbf{P}_i^l\}$ and $\{\mathbf{P}_i^u\}$, see Fig. 2(b), are computed according to

$$\mathbf{P}_i^l = \begin{cases} C^c(\hat{u}_i), & \text{for } i = 0 \\ C^c(\hat{u}_i) - \mathbf{n}(\hat{u}_i) \cdot f(r_{in}), & \text{for } i = 1 \\ C^c(\hat{u}_i) - \mathbf{n}(\hat{u}_i) \cdot t^l(\hat{u}_i), & \text{for } i = 2 : n - 2 \\ C^c(\hat{u}_i) - \mathbf{n}(\hat{u}_i) \cdot g(r_{out}), & \text{for } i = n - 1 \\ C^c(\hat{u}_i), & \text{for } i = n \end{cases} \quad (12)$$

and

$$\mathbf{P}_i^u = \begin{cases} C^c(\hat{u}_i), & \text{for } i = 0 \\ C^c(\hat{u}_i) + \mathbf{n}(\hat{u}_i) \cdot f(r_{in}), & \text{for } i = 1 \\ C^c(\hat{u}_i) + \mathbf{n}(\hat{u}_i) \cdot t^u(\hat{u}_i), & \text{for } i = 2 : n - 2 \\ C^c(\hat{u}_i) + \mathbf{n}(\hat{u}_i) \cdot g(r_{out}), & \text{for } i = n - 1 \\ C^c(\hat{u}_i), & \text{for } i = n. \end{cases} \quad (13)$$

The sampling values $\hat{u}_i$ are given by

$$\hat{u}_i = \begin{cases} 0, & \text{for } i = 0 \\ \frac{i-1}{n-2}, & \text{for } i = 1 : n - 1 \\ 1, & \text{for } i = n. \end{cases} \quad (14)$$

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

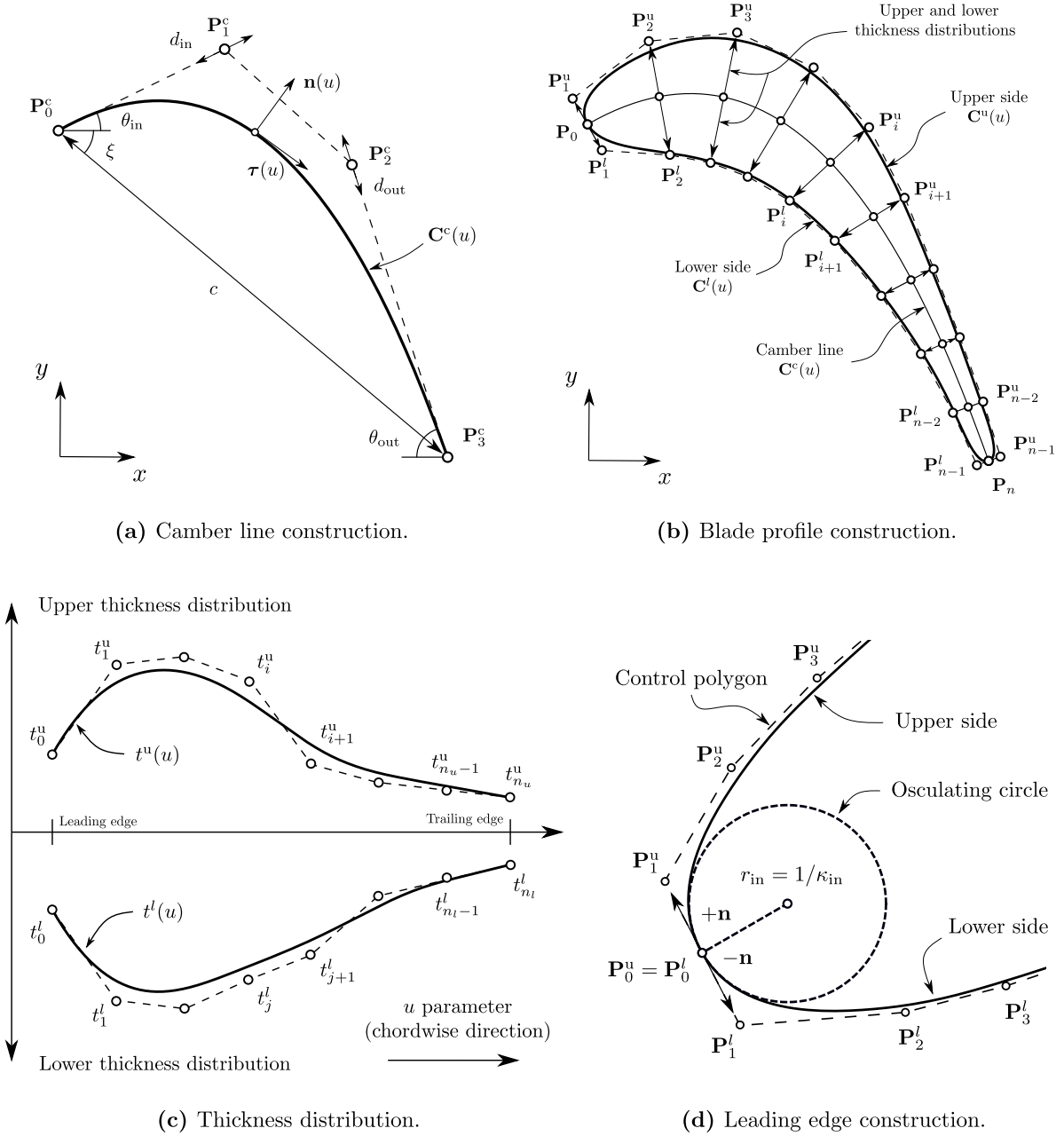


Fig. 2. Construction of the blade geometry in two dimensions. The upper and lower thickness distributions (bottom-left) are imposed in the direction normal to the camber line (top-left) to compute the location of the blade control points (top-right). The second and second-to-last control points are computed in a special way to impose the radii of curvature at the leading and trailing edges and to ensure that the blade profile is G^2 continuous (bottom-right).

number of control points, $\{t_i^u\}$ and $\{t_j^l\}$, 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 $\boldsymbol{\tau}(u)$ according to

$$\mathbf{n}(u) = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -\tau_y \\ \tau_x \end{bmatrix}, \text{ with } \boldsymbol{\tau}(u) = \frac{\dot{\mathbf{C}}^c(u)}{\|\dot{\mathbf{C}}^c(u)\|}, \quad (15)$$

where $\dot{\mathbf{C}}^c(u)$ is computed using analytical derivative formulas for B-spline curves [43, 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 G^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

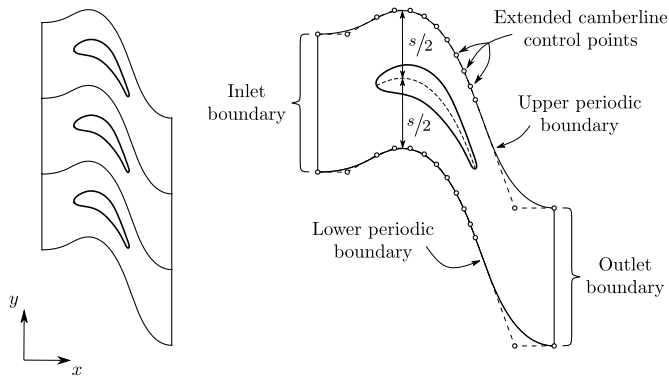


Fig. 3. Construction of the blade flow domain in two dimensions. The fluid-dynamic performance of the blade cascade (left) can be evaluated analyzing the flow around a single blade. The flow domain is defined by four boundaries: inflow, outflow, lower periodic, and upper periodic (right).

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-of-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.

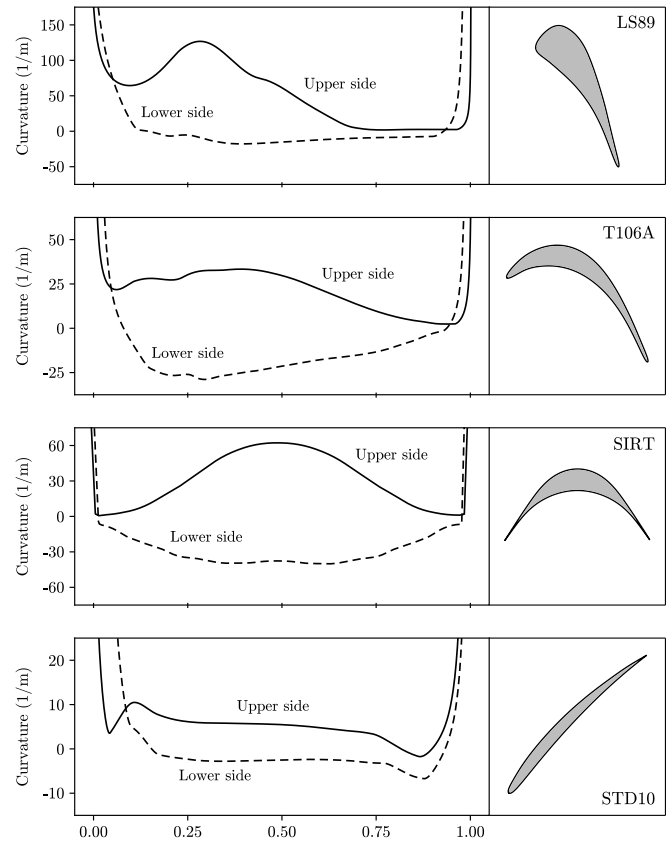


Fig. 4. Geometry and curvature distribution of the two-dimensional test cases. From top to bottom: LS89 [53,54], T106A [55], SIRT [56], and STD10 [57]. The abscissa of the curvature distribution is the normalized axial length.

Table 3

Three-dimensional design variables. Each design variable is provided as a set of control points that defines a continuous variation, except for the number of blades that is a single integer value.

Variable name	Symbol
Number of blades	N_b
Leading edge control points	x_1, z_1
Hub edge control points	x_2, z_2
Trailing edge control points	x_3, z_3
Shroud edge control points	x_4, z_4
Leading edge abscissa ^a	y_{in}
Stagger angle ^a	ξ
Inlet and exit metal angles ^a	$\theta_{in}, \theta_{out}$
Inlet and exit tangent proportions ^a	d_{in}, d_{out}
Inlet and exit radii of curvature ^a	r_{in}, r_{out}
Upper and lower thickness distributions	t^u, t^l

^aLaw of evolution in the spanwise direction.

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

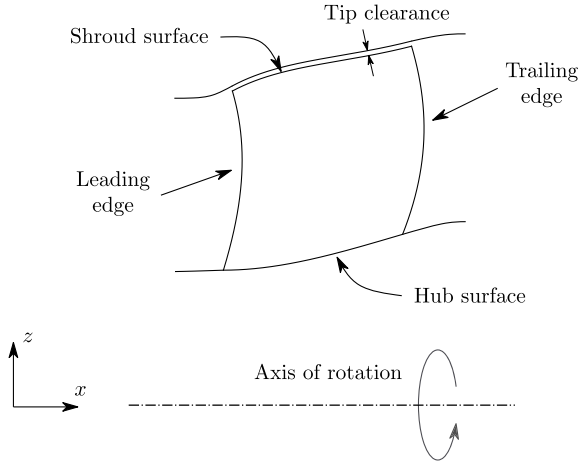


Fig. 5. Geometry of the blade in the meridional plane.

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 may be necessary to use 10–20 control points to describe the shape of a mixed-flow machine such as a centrifugal compressor.

The spanwise variation of the some design variables $\alpha(v)$, 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 $S^c(u, v)$ is defined as a bi-quartic B-spline surface with control points $P_{ij}^c = [x_{ij}^c, y_{ij}^c, z_{ij}^c]$. 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

$$\begin{aligned} \begin{bmatrix} x^c(u, v) \\ z^c(u, v) \end{bmatrix} &= (1-u) \cdot C_1^m(v) + u \cdot C_3^m(v) \\ &+ (1-v) \cdot C_2^m(u) + v \cdot C_4^m(u) \\ &- (1-v)(1-u) \cdot Q_{1,2}^m - v u \cdot Q_{3,4}^m \\ &- v(1-u) \cdot Q_{4,1}^m - (1-v)u \cdot Q_{2,3}^m, \end{aligned} \quad (16)$$

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 $y^c(u, v)$ with control points $\{y_0^c, y_1^c, y_2^c, y_3^c\}$ that are computed according to

$$y_0^c(v) = y_{in} \quad (17)$$

$$y_1^c(v) = y_0^c + d_{in} \cdot L \tan \theta_{in} \quad (18)$$

$$y_2^c(v) = y_3^c - d_{out} \cdot L \tan \theta_{out} \quad (19)$$

$$y_3^c(v) = y_{in} + L \tan \xi \quad (20)$$

This formulation ensures that the metal angles at the leading and trailing edges, θ_{in} and θ_{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

$$L(v) = \int_{u=0}^{u=1} \sqrt{\left(\frac{\partial x^c}{\partial u}\right)^2 + \left(\frac{\partial z^c}{\partial u}\right)^2} du \quad (21)$$

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, $S^l(u, v)$ and $S^u(u, v)$, 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 $\{P_{ij}^l\}$ and $\{P_{ij}^u\}$, see Fig. 9(b), are computed according to

$$P_{ij}^l = \begin{cases} S^c(\hat{u}_i, \hat{v}_j), & \text{for } i = 0 \\ S^c(\hat{u}_i, \hat{v}_j) - \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot f(r_{in}(\hat{v}_j)), & \text{for } i = 1 \\ S^c(\hat{u}_i, \hat{v}_j) - \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot t^l(\hat{u}_i, \hat{v}_j), & \text{for } i = 2 : n-2 \\ S^c(\hat{u}_i, \hat{v}_j) - \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot g(r_{out}(\hat{v}_j)), & \text{for } i = n-1 \\ S^c(\hat{u}_i, \hat{v}_j), & \text{for } i = n \end{cases} \quad (22)$$

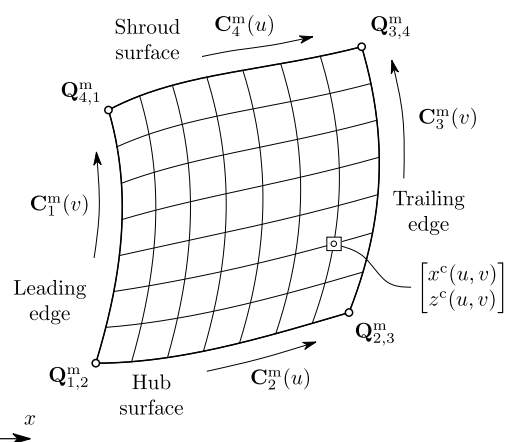
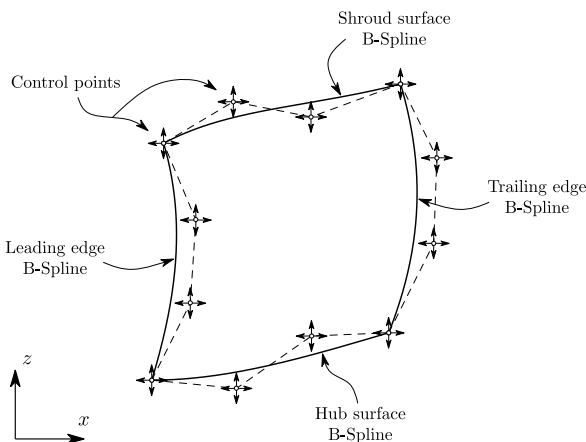


Fig. 6. Construction of the four B-splines that define the shape of the blade in the meridional plane (left) and point evaluation by transfinite interpolation (right). Note that the corner control points of the B-splines are shared.

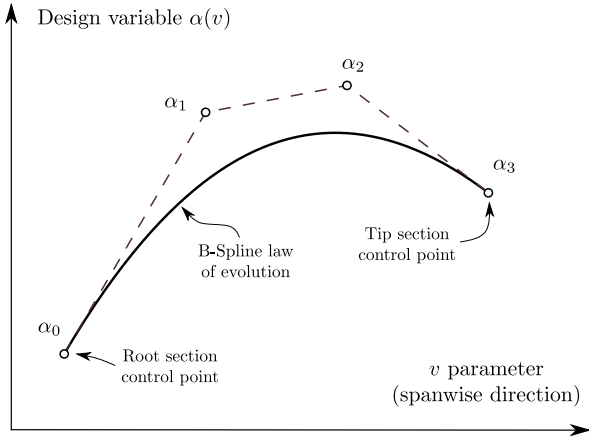


Fig. 7. Spanwise variation of a design variable.

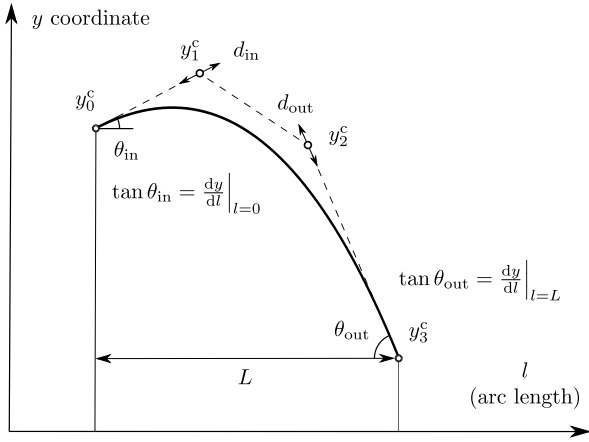


Fig. 8. Tangential camber line coordinate.

and

$$\mathbf{P}_{i,j}^u = \begin{cases} \mathbf{S}^c(\hat{u}_i, \hat{v}_j), & \text{for } i = 0 \\ \mathbf{S}^c(\hat{u}_i, \hat{v}_j) + \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot f(r_{in}(\hat{v}_j)), & \text{for } i = 1 \\ \mathbf{S}^c(\hat{u}_i, \hat{v}_j) + \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot t^u(\hat{u}_i, \hat{v}_j), & \text{for } i = 2 : n-2 \\ \mathbf{S}^c(\hat{u}_i, \hat{v}_j) + \mathbf{n}(\hat{u}_i, \hat{v}_j) \cdot g(r_{out}(\hat{v}_j)), & \text{for } i = n-1 \\ \mathbf{S}^c(\hat{u}_i, \hat{v}_j), & \text{for } i = n \end{cases} \quad (23)$$

where the sampling values $(\hat{u}_i, \hat{v}_j)$ are given by

$$\hat{u}_i = \begin{cases} 0, & \text{for } i = 0 \\ \frac{i-1}{n-2}, & \text{for } i = 1 : n-1 \\ 1, & \text{for } i = n \end{cases} \quad (24)$$

$$\hat{v}_j = \frac{j}{m}, \quad \text{for } j = 0 : m. \quad (25)$$

The upper and lower thickness distributions, $t^u(u, v)$ and $t^l(u, v)$, are given by bi-variate B-spline polynomials of degree three with an arbitrary number of control points $\{t_{i,j}^u\}$ and $\{t_{i,j}^l\}$, specified by the user. The unitary vectors normal to the camber surface $\mathbf{n}(u, v)$ are computed according to

$$\mathbf{n} = -\frac{\boldsymbol{\tau}_u \times \boldsymbol{\tau}_v}{\|\boldsymbol{\tau}_u \times \boldsymbol{\tau}_v\|}, \quad (26)$$

where the tangent vectors $\boldsymbol{\tau}_u$ and $\boldsymbol{\tau}_v$ are given by

$$\boldsymbol{\tau}_u = \frac{\partial \mathbf{S}^c}{\partial u} \quad \text{and} \quad \boldsymbol{\tau}_v = \frac{\partial \mathbf{S}^c}{\partial v}. \quad (27)$$

The partial derivatives of the camber surface with respect to u and v are computed analytically using B-spline surface derivative

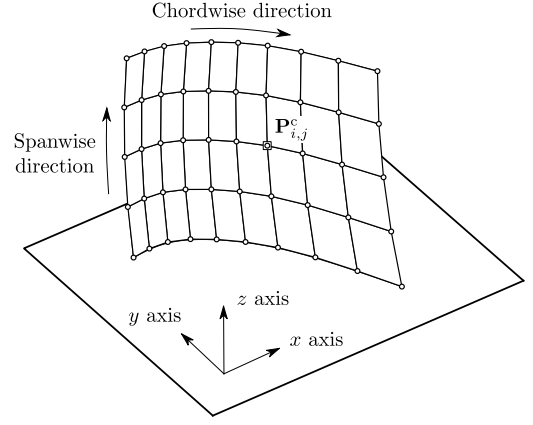


Fig. 9(a). Control points defining the camber surface.

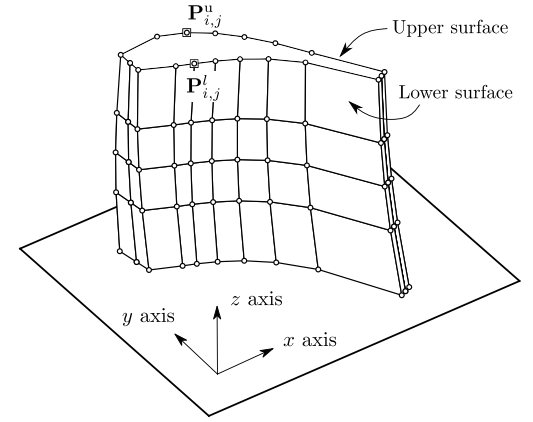


Fig. 9(b). Control points defining the blade surface.

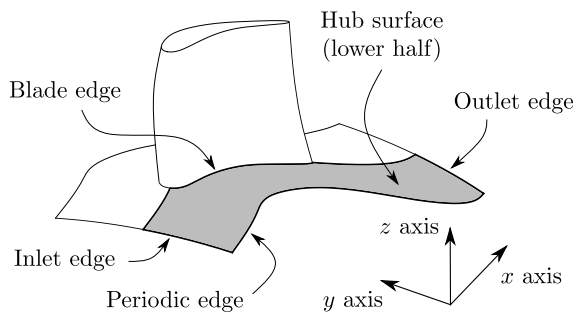
formulas [43, pp. 110–115]. In addition, the functions $f(r)$ and $g(r)$ appearing in Eqs. (22) and (23) are used to impose the radius of curvature at the leading and trailing edges, ensuring that the upper and lower surfaces of the blade are smoothly joined with G^2 continuity. The derivation of the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

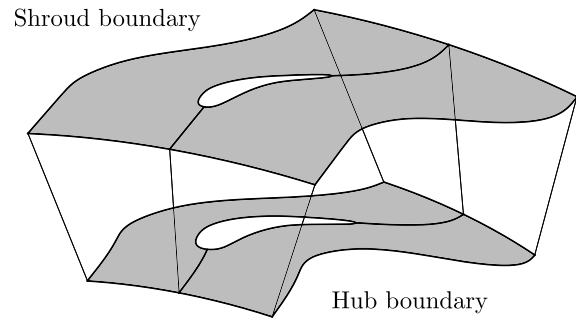
$$\mathbf{P}_{\text{annular}}^b = \mathcal{H}\{\mathbf{P}_{\text{linear}}^b\} = \mathcal{H}\{[x, y, z]\} = [x, z \cdot \sin(y/z), z \cdot \cos(y/z)]. \quad (28)$$

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

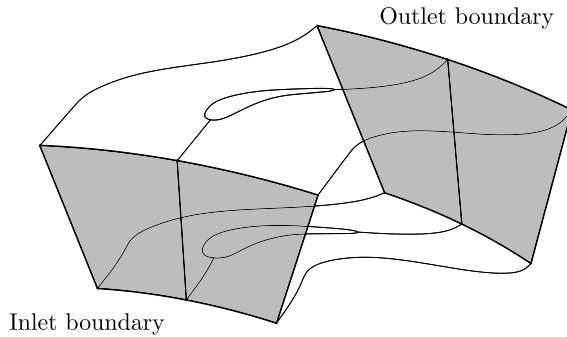
The flow domain around a blade is characterized by the hub, shroud, inlet, outlet, and periodic boundaries as illustrated in Fig. 10. The hub boundary consists of two surfaces that conform with the blade at its root, see Fig. 10(a). Each of these surfaces is defined as a Coons patch [43, pp. 456–507] 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



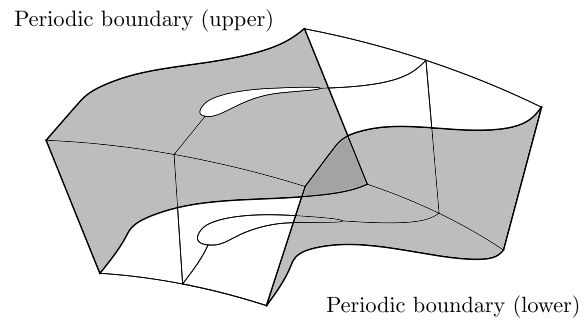
(a) Hub surface as a Coons patch



(b) Hub and shroud boundaries



(c) Inlet and outlet boundaries.



(d) Periodic boundaries.

Fig. 10. Construction of the blade flow domain in three dimensions.

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].

Application

The flexibility of the proposed three-dimensional blade parametrization method is demonstrated by reconstructing the four blade geometries shown in Fig. 11. The first example, the AACHEN case, is a prismatic axial turbine stator blade [60]. The meridional plane is defined by 4 control points and the design variables are constant in the spanwise direction (1 control point), resulting in a total of 26 design variables. The second case, NASA R67, is an axial compressor rotor blade [61,62]. The blade is highly twisted due to the change in blade speed from root to tip and it was necessary to use 4 control points to describe the spanwise variation of the design variables, resulting in a

total of 111 design variables. Similarly the XPROP case illustrates the geometry of an aircraft propeller blade [63]. In this case it was necessary to use 5 control points to describe the twist of the blades, resulting in 113 design variables. Finally, the APU blade is the rotor of a mixed-flow turbine (radial-inflow, axial-outflow) [64,65]. The complex shape of the blade in the meridional plane was described using 14 control points and the spanwise variation of the blade sections was described using 3 control points per design variable, giving rise to 86 design variables. The numerical values of the design variables used to produce the blades were computed from a set of scattered point coordinates using the method described in Section 6.

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

$$\frac{\partial F}{\partial \alpha} = \frac{F(\alpha + h) - F(\alpha)}{h} + \mathcal{O}(h), \quad (29)$$

or central finite differences given as

$$\frac{\partial F}{\partial \alpha} = \frac{F(\alpha + h) - F(\alpha - h)}{2h} + \mathcal{O}(h^2), \quad (30)$$

where $F(\alpha)$ can be identified with $\mathbf{C}^b(u, \alpha)$ in two dimensions or $\mathbf{S}^b(u, v, \alpha)$ 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.

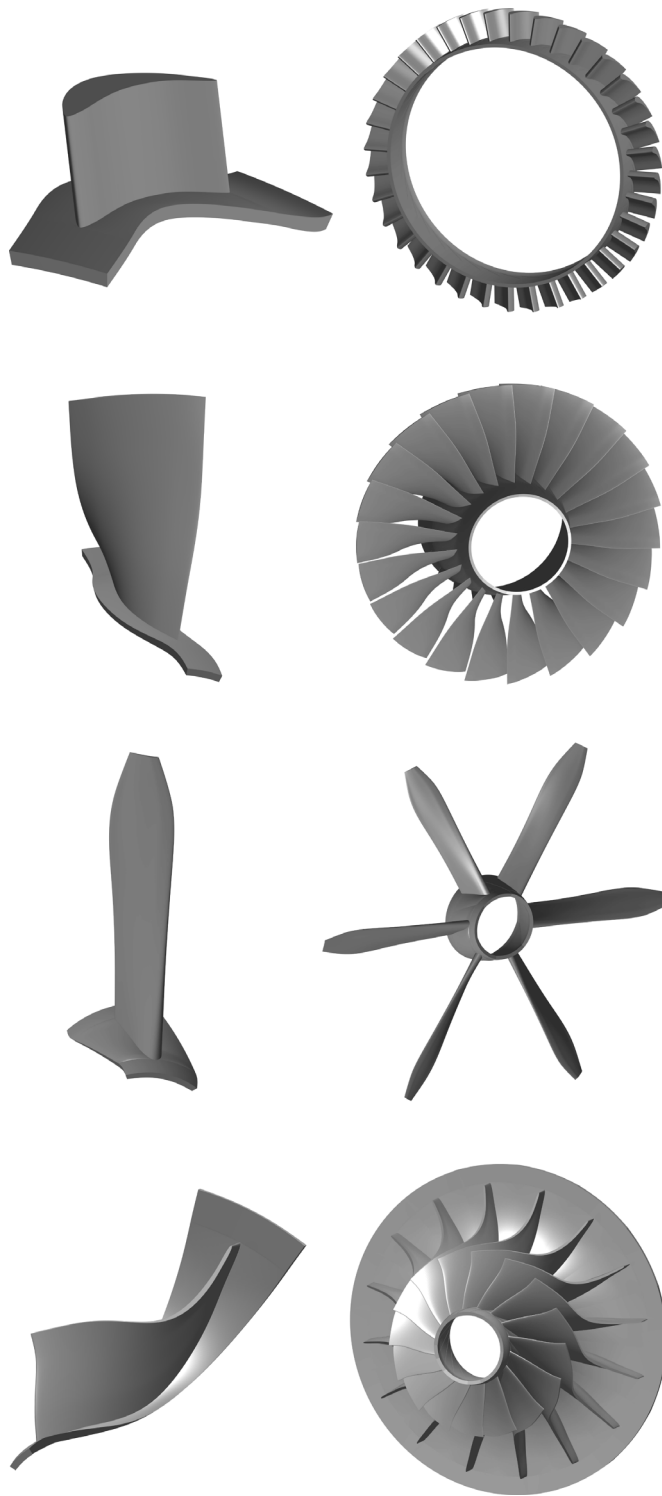


Fig. 11. Geometry of the three-dimensional test cases. From top to bottom: AACHEN stator [60], NASA R67 [61,62], XPROP propeller [63], APU rotor [64,65].

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

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

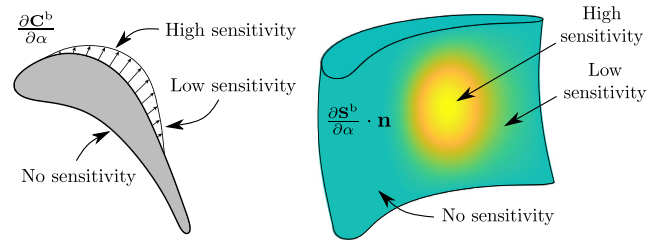


Fig. 12. Sensitivity of the blade geometry with respect to a thickness distribution control point. The sensitivity of the 2D case (left) is represented as a quiver plot. The sensitivity of the 3D case (right) is represented as a colormap of the scalar field given by the dot product of the sensitivity and the unitary vector normal to the blade surface.

Re-arranging the imaginary part of the equation leads to

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

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 an 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

Table 4
Summary of the test cases and matching results.

Name and reference	Description	Dim.	DVs	Error abs. ^a (mm)	Error rel. ^b (%)
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 turbine 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

^aDefined as the arithmetic mean deviation between the prescribed and the matched blades.

^bDefined 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).

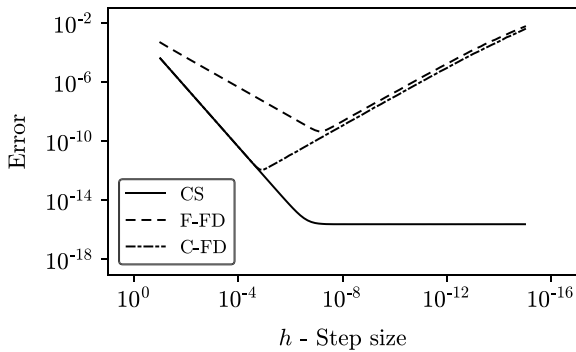


Fig. 13. Sensitivity error at different step sizes for the complex-step method (CS), forward finite differences (F-FD), and central finite differences (C-FD).

truncation error. In addition, it can be observed that the complex-step method and the central finite differences agree in interval when the truncation error dominates ($h \lesssim 10^{-6}$). This verifies that the implementation of the complex-step method is correct. Although not shown here for brevity, the authors performed similar convergence studies for all the design variables of each of the test cases summarized in Table 4 and obtained similar results.

6. Blade matching methodology

In order to optimize the performance of an existing turbomachinery blade, it is essential to find a parametric representation of its geometry, which is usually available as a set of scattered points coordinates $\mathbf{Q}_i$, with $i = 1, 2, \dots, N_Q$ obtained from a mesh, from sampling a CAD model, or from laser scan measurements. This section proposes a systematic method to find the set of design variables that best approximates the shape of a prescribed blade geometry. The method can be divided in two phases: (1) the point projection phase and (2) the geometry update phase. It is assumed that the designer starts from an initial guess for the design variables that roughly approximates the existing geometry, see Fig. 14(a).

In the point projection phase, the goal is to find the parametric values u_i , in two dimensions, or (u_i, v_i) in three dimensions, that minimize the distance with respect to each prescribed point $\mathbf{Q}_i$. The two-dimensional point projection problem can be formulated as

$$\underset{u \in \mathbb{R}}{\text{minimize}} \quad J(u) = \frac{1}{2} \|\mathbf{C}^b(u) - \mathbf{Q}_i\|^2, \quad (33)$$

subject to $0 \leq u \leq 1$,

where J is the distance between the prescribed and the parametrized point. The gradient of the objective function J can be computed analytically as

$$\nabla J = \frac{\partial J}{\partial u} = (\mathbf{C}^b(u) - \mathbf{Q}_i) \cdot \frac{\partial \mathbf{C}^b}{\partial u}. \quad (34)$$

Similarly, in three dimensions, the point projection problem is given by

$$\underset{(u,v) \in \mathbb{R}^2}{\text{minimize}} \quad J(u, v) = \frac{1}{2} \|\mathbf{S}^b(u, v) - \mathbf{Q}_i\|^2, \quad (35)$$

subject to $0 \leq u, v \leq 1$

and the gradient of the objective function can be computed according to

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial u} \\ \frac{\partial J}{\partial v} \end{bmatrix} = \begin{bmatrix} (\mathbf{S}^b(u, v) - \mathbf{Q}_i) \cdot \frac{\partial \mathbf{S}^b}{\partial u} \\ (\mathbf{S}^b(u, v) - \mathbf{Q}_i) \cdot \frac{\partial \mathbf{S}^b}{\partial v} \end{bmatrix}. \quad (36)$$

Note that the geometry of the parametrized blade *does not change* during the point projection phase. One common pitfall when solving the point projection problem is that the optimization may converge to a local minimum as illustrated in Fig. 14(b). This limitation can be addressed by solving the point projection problem from different starting points and then selecting the global minimum among the various solutions or by sampling the parametrized blade at several locations and then starting the optimization from the test point that is closest to $\mathbf{Q}_i$ [43, pp. 229–234].

In the geometry update phase, the goal is to find the set of design variables α that minimizes the deviation between the parametrized and the prescribed blades. This can be formulated as an unconstrained minimization problem where the objective function is the sum of the distances between each projected point and the corresponding prescribed point. This optimization problem is given by

$$\underset{\alpha \in \mathbb{R}^\alpha}{\text{minimize}} \quad J(\alpha) = \sum_{i=1}^{N_Q} \|\mathbf{C}^b(u_i, \alpha) - \mathbf{Q}_i\|^2 \quad (37)$$

in two dimensions and by

$$\underset{\alpha \in \mathbb{R}^\alpha}{\text{minimize}} \quad J(\alpha) = \sum_{i=1}^{N_Q} \|\mathbf{S}^b(u_i, v_i, \alpha) - \mathbf{Q}_i\|^2 \quad (38)$$

in three dimensions, where α 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 α , 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

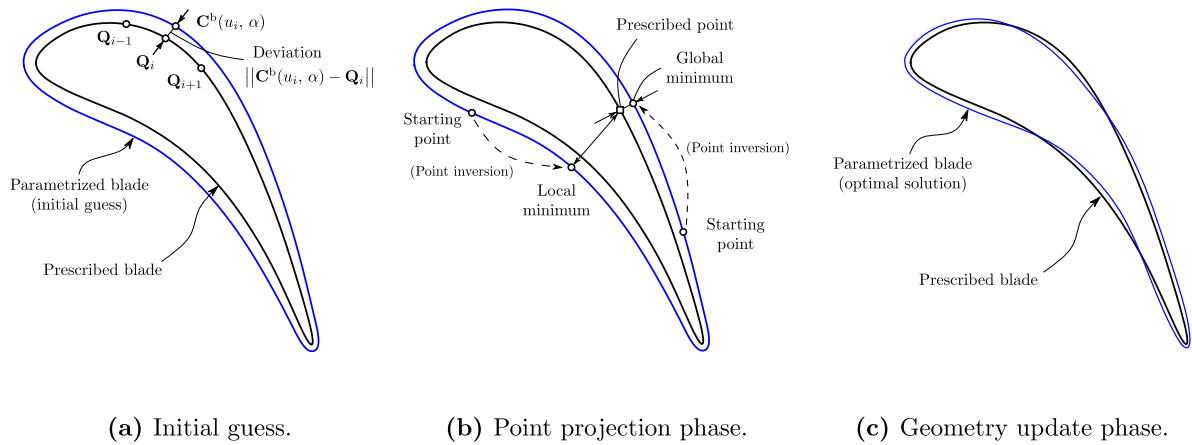


Fig. 14. Illustration of the blade matching problem in two dimensions. The deviation between the prescribed and parametrized blades after the geometry update phase was exaggerated to improve visibility.

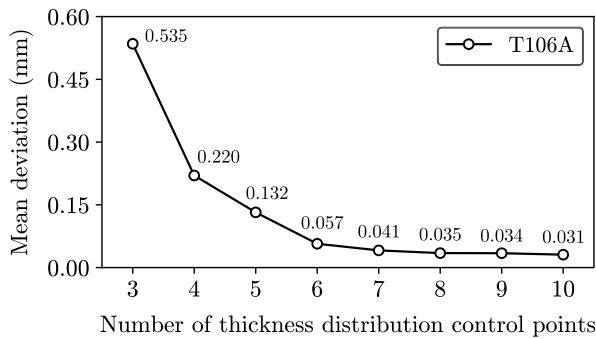


Fig. 15. Matching error as a function of the number of thickness distribution control points for the T106A test case.

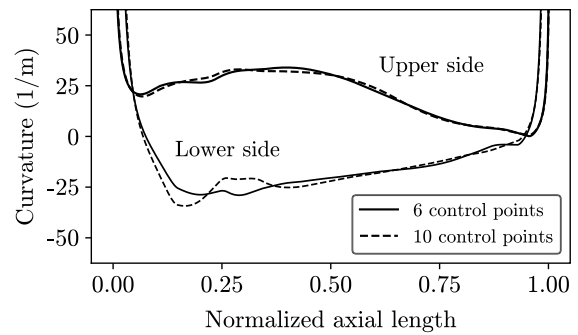


Fig. 16. Curvature distribution of the T106A profile when the matching is performed using 6 and 10 control points.

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 (≈ 0.05 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

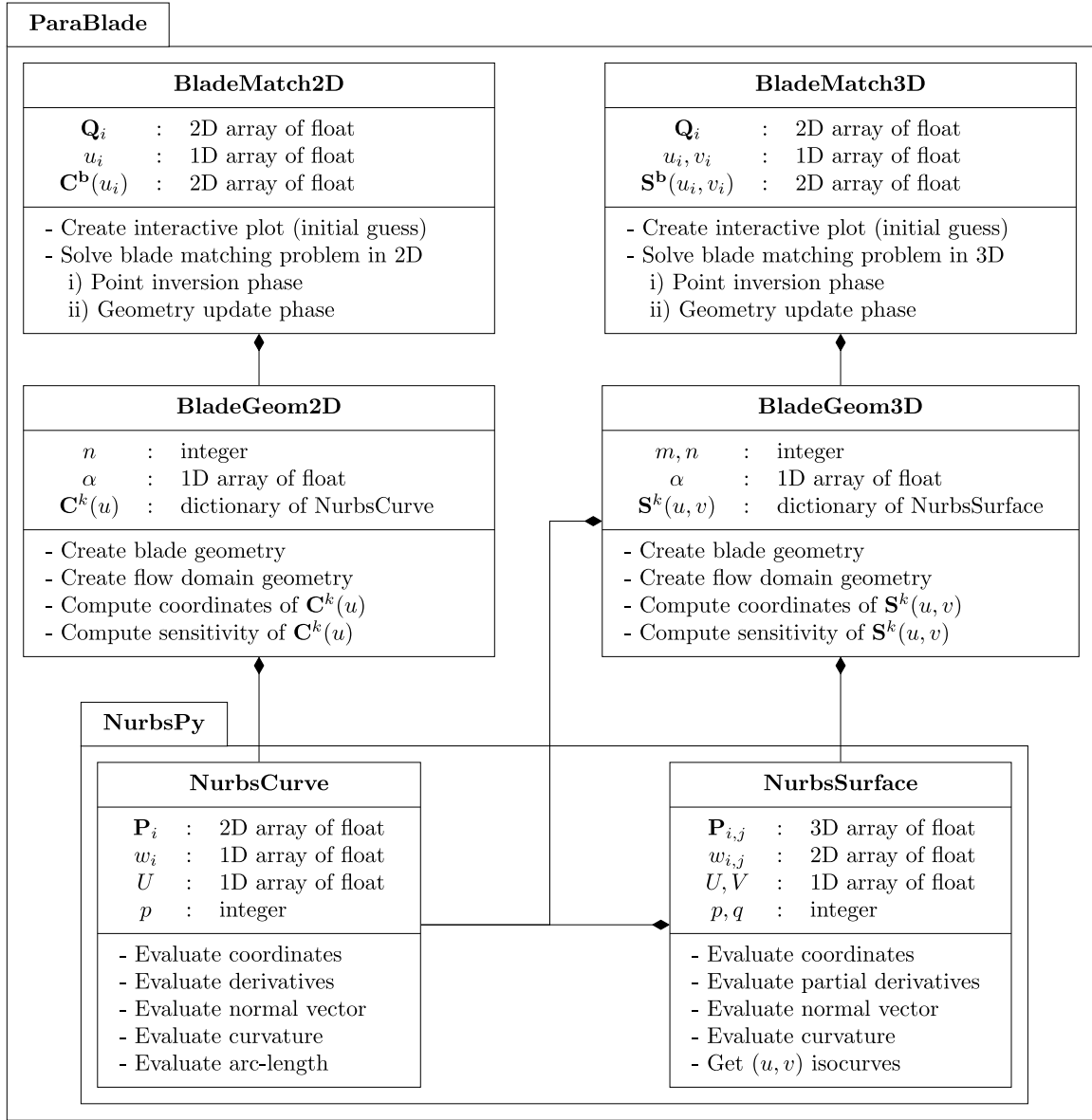


Fig. 17. Class diagram of the ParaBlade and NurbsPy packages. Each class is represented by a box with three compartments containing its name and main attributes/methods. The diamond-ended arrows (◆) represent a composition relation between two classes. This means that the class next to the diamond symbol contains one or more instances of the class at the other end of the line. For instance, the BladeMatch3D contains one instance of the BladeGeom3D class, which in turn contains several NurbsCurve and NurbsSurface objects.

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 255016) for the COPRO project, the Netherlands Organization for Scientific

Research (grant project 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 $\mathbf{C}(u)$ is given by

$$\kappa(u) = \frac{\|\ddot{\mathbf{C}}(u) \times \dot{\mathbf{C}}(u)\|}{\|\dot{\mathbf{C}}(u)\|^3}. \quad (\text{A.1})$$

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

$$\dot{\mathbf{C}}(u=0) = \left(\frac{p}{u_{p+1}}\right) \left(\frac{w_1}{w_0}\right) (\mathbf{P}_1 - \mathbf{P}_0) \quad (\text{A.2})$$

and

$$\begin{aligned}\ddot{\mathbf{C}}(u=0) = & + \frac{p(p-1)}{u_{p+1}} \left(\frac{1}{u_{p+2}} \right) \left(\frac{w_2}{w_0} \right) (\mathbf{P}_2 - \mathbf{P}_0) + \\ & - \frac{p(p-1)}{u_{p+1}} \left(\frac{1}{u_{p+1}} + \frac{1}{u_{p+2}} \right) \left(\frac{w_1}{w_0} \right) (\mathbf{P}_1 - \mathbf{P}_0) + \\ & + \frac{2p^2}{u_{p+1}^2} \left(\frac{w_1}{w_0} \right) \left(1 - \frac{w_1}{w_0} \right) (\mathbf{P}_1 - \mathbf{P}_0),\end{aligned}\quad (\text{A.3})$$

respectively, where p is the degree of the curve, u_k are the knot values, $\mathbf{P}_k$ are the control point coordinates, and w_k are the control point weights. Inserting these expressions into the curvature definition and using the fact that the vector cross product of two parallel vectors is zero we find

$$\begin{aligned}\kappa(u=0) = & \left(\frac{p-1}{p} \right) \left(\frac{u_{p+1}}{u_{p+2}} \right) \left(\frac{w_0 w_2}{w_1^2} \right) \\ & \times \frac{\|(\mathbf{P}_2 - \mathbf{P}_0) \times (\mathbf{P}_1 - \mathbf{P}_0)\|}{\|\mathbf{P}_1 - \mathbf{P}_0\|^3}.\end{aligned}\quad (\text{A.4})$$

Noting that unitary vector perpendicular to the camber line at the leading edge ($\mathbf{n}$) points from $\mathbf{P}_0$ to $\mathbf{P}_1$, see Fig. 2(d), the previous equation simplifies to

$$\kappa_{\text{in}} = \frac{1}{r_{\text{in}}} = \left(\frac{p-1}{p} \right) \left(\frac{u_{p+1}}{u_{p+2}} \right) \left(\frac{w_0 w_2}{w_1^2} \right) \frac{\|(\mathbf{P}_2 - \mathbf{P}_0) \times \mathbf{n}\|}{\|\mathbf{P}_1 - \mathbf{P}_0\|^2}.\quad (\text{A.5})$$

Solving for $\|\mathbf{P}_1 - \mathbf{P}_0\|$, the location of the control point $\mathbf{P}_1$ that guarantees that the radius of curvature at the leading edge is r_{in} is given by

$$\mathbf{P}_1 = \mathbf{P}_0 \pm \|\mathbf{P}_1 - \mathbf{P}_0\| \cdot \mathbf{n} = \mathbf{P}_0 \pm f(r_{\text{in}}) \cdot \mathbf{n},\quad (\text{A.6})$$

where the plus and minus signs correspond to the upper and lower sides of the blade, respectively, and the function $f(r_{\text{in}})$ is given by

$$f(r_{\text{in}})^2 = r_{\text{in}} \left(\frac{p-1}{p} \right) \left(\frac{u_{p+1}}{u_{p+2}} \right) \left(\frac{w_0 w_2}{w_1^2} \right) \|(\mathbf{P}_2 - \mathbf{P}_0) \times \mathbf{n}\|. \quad (\text{A.7})$$

The derivation for the trailing edge is analogous. The first and second derivatives of a NURBS curve at its *end point* are given by

$$\dot{\mathbf{C}}(u=1) = \left(\frac{p}{1-u_n} \right) \left(\frac{w_{n-1}}{w_n} \right) (\mathbf{P}_n - \mathbf{P}_{n-1})\quad (\text{A.8})$$

and

$$\begin{aligned}\ddot{\mathbf{C}}(u=1) = & + \frac{p(p-1)}{1-u_n} \left(\frac{1}{1-u_{n-1}} \right) \left(\frac{w_{n-2}}{w_n} \right) (\mathbf{P}_{n-2} - \mathbf{P}_n) + \\ & - \frac{p(p-1)}{1-u_n} \left(\frac{1}{1-u_n} + \frac{1}{1-u_{n-1}} \right) \left(\frac{w_{n-1}}{w_n} \right) (\mathbf{P}_{n-1} - \mathbf{P}_n) + \\ & + \frac{2p^2}{(1-u_n)^2} \left(\frac{w_{n-1}}{w_n} \right) \left(1 - \frac{w_{n-1}}{w_n} \right) (\mathbf{P}_{n-1} - \mathbf{P}_n),\end{aligned}\quad (\text{A.9})$$

respectively. Inserting these expressions into the curvature definition we find

$$\begin{aligned}\kappa(u=1) = & \left(\frac{p-1}{p} \right) \left(\frac{1-u_n}{1-u_{n-1}} \right) \left(\frac{w_n w_{n-2}}{w_{n-1}^2} \right) \\ & \times \frac{\|(\mathbf{P}_{n-2} - \mathbf{P}_n) \times (\mathbf{P}_{n-1} - \mathbf{P}_n)\|}{\|\mathbf{P}_{n-1} - \mathbf{P}_n\|^3}.\end{aligned}\quad (\text{A.10})$$

Noting that unitary vector perpendicular to the camber line at the trailing edge ($\mathbf{n}$) points from $\mathbf{P}_n$ to $\mathbf{P}_{n-1}$, the previous equation simplifies to

$$\begin{aligned}\kappa_{\text{out}} = & \frac{1}{r_{\text{out}}} \\ = & \left(\frac{p-1}{p} \right) \left(\frac{1-u_n}{1-u_{n-1}} \right) \left(\frac{w_n w_{n-2}}{w_{n-1}^2} \right) \frac{\|(\mathbf{P}_{n-2} - \mathbf{P}_n) \times \mathbf{n}\|}{\|\mathbf{P}_{n-1} - \mathbf{P}_n\|^2}.\end{aligned}\quad (\text{A.11})$$

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

$$\mathbf{P}_{n-1} = \mathbf{P}_n \pm \|\mathbf{P}_{n-1} - \mathbf{P}_n\| \cdot \mathbf{n} = \mathbf{P}_n \pm g(r_{\text{out}}) \cdot \mathbf{n},\quad (\text{A.12})$$

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

$$\begin{aligned}g(r_{\text{out}})^2 = & r_{\text{out}} \left(\frac{p-1}{p} \right) \left(\frac{1-u_n}{1-u_{n-1}} \right) \left(\frac{w_n w_{n-2}}{w_{n-1}^2} \right) \|(\mathbf{P}_{n-2} - \mathbf{P}_n) \times \mathbf{n}\|.\end{aligned}\quad (\text{A.13})$$

Note that this construction guarantees that the blade is G^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] Verstraete T. Toward gradient-based optimization of full gas turbines. *Mech Eng* 2019;141(03):54–5. <http://dx.doi.org/10.1115/1.2019-MAR-7>.
- [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] Banović M, Mykhaskiv O, Auriemma S, Walther A, Legrand H, Müller 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, Müller 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/10.1115/1.4029550>.
- [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;136(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.Z1FV0I>.
- [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>.

- [13] 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* 2018;72:267–77. <http://dx.doi.org/10.1016/j.ast.2017.11.015>.
- [14] Gagliardi F, Giannakoglou KC. RBF-based morphing of B-Rep models for use in aerodynamic shape optimization. *Adv Eng Softw* 2019;138:1–17. <http://dx.doi.org/10.1016/j.advengsoft.2019.102724>.
- [15] Bahamonde S, Pini M, De Servi C, Rubino A, Colonna P. Method for the preliminary fluid dynamic design of high-temperature mini-organic Rankine cycle turbines. *Trans ASME, J Eng Gas Turbines Power* 2017;139(8):1–14. <http://dx.doi.org/10.1115/1.4035841>.
- [16] Agromayor R, Nord LO. Preliminary design and optimization of axial turbines accounting for diffuser performance. *Int J Turbomach Propuls Power* 2019;4(3):1–32. <http://dx.doi.org/10.3390/ijtp4030032>.
- [17] Damle SV, Dang TQ, Reddy DR. Throughflow method for turbomachines applicable for all flow regimes. *Trans ASME, J Turbomach* 1997;119(2):256–62. <http://dx.doi.org/10.1115/1.2841108>.
- [18] Persico G, Rebay S. A penalty formulation for the throughflow modeling of turbomachinery. *Comput & Fluids* 2012;60:86–98. <http://dx.doi.org/10.1016/j.compfluid.2012.03.001>.
- [19] Dunham J. A parametric method of turbine blade profile design. In: *ASME turbo expo: Power for land, sea, and air*. 1974, p. 1–9. <http://dx.doi.org/10.1115/74-GT-119>.
- [20] Crouse JE, Gorrell WT. Computer program for aerodynamic and blading design of multistage axial-flow compressors. Tech. Rep. TP 1946, NASA Lewis Research Center; Cleveland, OH, United States; 1981, p. 1–110.
- [21] Ye ZQ. A systematic computational design system for turbine cascades, airfoil geometry and blade-to-blade analysis. *Trans ASME, J Eng Gas Turbines Power* 1984;106(3):598–605. <http://dx.doi.org/10.1115/1.3239612>.
- [22] Pritchard LJ. An eleven parameter axial turbine airfoil geometry model. In: *ASME turbo expo: Power for land, sea, and air*. 1985, p. 1–12. <http://dx.doi.org/10.1115/85-GT-219>.
- [23] Korakianitis T. Hierarchical development of three direct-design methods for two-dimensional axial-turbomachinery cascades. *Trans ASME, J Turbomach* 1993;115(2):314–24. <http://dx.doi.org/10.1115/1.2929237>.
- [24] Aungier RH. Turbine aerodynamics: Axial-flow and radial-flow turbine design and analysis. 1st ed.. ASME Press; 2006, p. 166–83. <http://dx.doi.org/10.1115/1.802418>.
- [25] Engeli M, Zollinger HJ, Allemann JC. A computer program for the design of turbomachinery blades. In: *ASME turbo expo: Power for land, sea, and air*. 1974, p. 1–10. <http://dx.doi.org/10.1115/78-GT-36>.
- [26] Casey MV. A computational geometry for the blades and internal flow channels of centrifugal compressors. *Trans ASME, J Eng Power* 1983;105(2):288–95. <http://dx.doi.org/10.1115/1.3227414>.
- [27] Goel S, Cofer I, Singh H. Turbine airfoil design optimization. In: *ASME turbo expo: Power for land, sea, and air*. 1996, p. 1–10. <http://dx.doi.org/10.1115/96-GT-158>.
- [28] Giannakoglou KC. Designing turbomachinery blades using evolutionary methods. In: *ASME turbo expo: Power for land, sea, and air*. 1999, p. 1–9. <http://dx.doi.org/10.1115/99-GT-181>.
- [29] Trigg MA, Tubby GR, Sheard AG. Automatic genetic optimization approach to two-dimensional blade profile design for steam turbines. *Trans ASME, J Turbomach* 1999;121(1):11–7. <http://dx.doi.org/10.1115/1.2841220>.
- [30] Pierret S, Van den Braembussche RA. Turbomachinery blade design using a Navier-Stokes solver and artificial neural network. *Trans ASME, J Turbomach* 1999;121(2):326–32. <http://dx.doi.org/10.1115/1.2841318>.
- [31] Pierret S, Demeulenaere A, Gouverneur B, Van den Braembussche R. Designing turbomachinery blades with the function approximation concept and the Navier-Stokes equations. In: *8th symposium on multidisciplinary analysis and optimization*. 2000, p. 1–11. <http://dx.doi.org/10.2514/6.2000-4879>.
- [32] Oyama A, Liou M-S, Obayashi S. Transonic axial-flow blade optimization: Evolutionary algorithms/three-dimensional Navier-Stokes solver. *J Propul Power* 2004;20(4):612–9. <http://dx.doi.org/10.2514/1.2290>.
- [33] Huppertz A, Flassig PM, Flassig RJ, Swoboda M. Knowledge based 2D blade design using multi-objective aerodynamic optimization and a neural network. In: *ASME turbo expo: Power for land, sea, and air*. 2007, p. 413–23. <http://dx.doi.org/10.1115/GT2007-28204>.
- [34] Verstraete T. CAD: A computer aided design and optimization tool for turbomachinery applications. In: *Proceedings of the 2nd International Conference on Engineering Optimization*. Lisbon, Portugal; 2010, p. 1–10.
- [35] Siddappaji K, Turner MG, Merchant A. General capability of parametric 3D blade design tool for turbomachinery. In: *ASME turbo expo: Power for land, sea, and air*. 2012, p. 1–14. <http://dx.doi.org/10.1115/GT2012-69756>.
- [36] Sanchez-Torreguitart I, Verstraete T, Müller L. Optimization of the LS89 axial turbine profile using a CAD and adjoint based approach. *International Trans ASME, J Turbomach, Propulsion and Power* 2018;3(3):1–20. <http://dx.doi.org/10.3390/ijtp3030020>.
- [37] Mykhaskiv O, Banović M, Auriemma S, Mohanamurthy P, Walther A, Legrand H, et al. NURBS-Based and parametric-based shape optimization with differentiated CAD kernel. *Comput-Aided Des Appl* 2018;15(6):916–26. <http://dx.doi.org/10.1080/16864360.2018.1462881>.
- [38] Miller IV PL, Oliver JH, Miller DP, Tweedt DL. BladeCAD: An interactive geometric design tool for turbomachinery blades. In: *ASME turbo expo: Power for land, sea, and air*. 1996, p. 1–8. <http://dx.doi.org/10.1115/96-GT-058>.
- [39] Gräsel J, Keskin A, Swoboda M, Przewozny H, Saxer A. A full parametric model for turbomachinery blade design and optimisation. In: *ASME international design engineering technical conferences and computers and information in engineering conference*. 2004, p. 907–14. <http://dx.doi.org/10.1115/DETC2004-57467>.
- [40] 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>.
- [41] Müller L, Verstraete T. CAD Integrated multipoint adjoint-based optimization of a turbocharger radial turbine. *Int J Turbomach Propuls Power* 2017;2(3):1–14. <http://dx.doi.org/10.3390/ijtp2030014>.
- [42] Anand N, Vitale S, Pini M, Colonna P. Assessment of FFD and CAD-based shape parametrization methods for adjoint-based turbomachinery shape optimization. In: *Proceedings of the Montreal 2018 global power and propulsion forum*. 2018, p. 1–8.
- [43] Piegel L, Tiller W. The NURBS book. 2nd ed.. Springer Science & Business Media; 2012. <http://dx.doi.org/10.1007/978-3-642-97385-7>.
- [44] Farin GE. Curves and surfaces for CAGD: A practical guide. 5th ed.. Morgan Kaufmann; 2001.
- [45] Taylor NJ. Industrial perspectives on geometry handling for aerodynamics. In: *22nd AIAA computational fluid dynamics conference*. 2015, p. 1–12. <http://dx.doi.org/10.2514/6.2015-3408>.
- [46] Xu S, Timme S, Mykhaskiv O, Müller J-D. Wing-body junction optimisation with CAD-based parametrisation including a moving intersection. *Aerosp Sci Technol* 2017;68:543–51. <http://dx.doi.org/10.1016/j.ast.2017.06.014>.
- [47] Lyness JN, Moler CB. Numerical differentiation of analytic functions. *SIAM J Numer Anal* 1967;4(2):202–10. <http://dx.doi.org/10.1137/0704019>.
- [48] Squire W, Trapp G. Using complex variables to estimate derivatives of real functions. *SIAM Rev* 1998;40(1):110–2. <http://dx.doi.org/10.1137/S003614459631241X>.
- [49] Martins JRR, Sturdza P, Alonso JJ. The complex-step derivative approximation. *ACM Trans Math Softw* 2003;29(3):245–62. <http://dx.doi.org/10.1145/838250.838251>.
- [50] Piegel L. On NURBS: A survey. *IEEE Comput Graph Appl* 1991;11(1):55–71. <http://dx.doi.org/10.1109/38.67702>.
- [51] Korakianitis T. Prescribed-curvature-distribution airfoils for the preliminary geometric design of axial-turbomachinery cascades. *Trans ASME, J Turbomach* 1993;115(2):325–33. <http://dx.doi.org/10.1115/1.2929238>.
- [52] Korakianitis T, Papagiannidis P. Improved turbine-blade design techniques using 4th-order parametric-spline segments. *Comput-Aided Des* 1993;25(5):289–99. [http://dx.doi.org/10.1016/0010-4485\(93\)90086-4](http://dx.doi.org/10.1016/0010-4485(93)90086-4).
- [53] Arts T, Lambert de Rouvroit M, Rutherford AW. Aero-Thermal investigation of a highly loaded transonic linear turbine guide vane cascade: A test case for inviscid and viscous flow computations. Tech. rep. 174, VKI Training Center for Experimental Aerodynamics; 1990, p. 1–97.
- [54] Arts T, Lambert de Rouvroit M. Aero-thermal performance of a two-dimensional highly loaded transonic turbine nozzle guide vane: A test case for inviscid and viscous flow computations. *Trans ASME, J Turbomach* 1992;114(1):147–54. <http://dx.doi.org/10.1115/1.2927978>.
- [55] Stadtmüller P, Fottner L. A test case for the numerical investigation of wake passing effects on a highly loaded LP turbine cascade blade. In: *ASME turbo expo: Power for land, sea, and air*. American Society of Mechanical Engineers Digital Collection; 2001, p. 1–8. <http://dx.doi.org/10.1115/2001-GT-0311>.
- [56] Anand N, Rubino A, Colonna P, Pini M. Adjoint-based aeroelastic design optimization using a harmonic balance method. In: *ASME turbo expo: Turbomachinery technical conference and exposition*. American Society of Mechanical Engineers Digital Collection; 2020, p. 1–12.
- [57] Fransson TH, Verdon JM. Panel discussion on standard configurations for unsteady flow through vibrating axial-flow turbomachine-cascades. In: *Unsteady aerodynamics, aeroacoustics, and aeroelasticity of turbomachines and propellers*. Springer; 1993, p. 859–89. <http://dx.doi.org/10.1007/978-1-4613-9341-2>.
- [58] Gordon WJ, Hall CA. Construction of curvilinear coordinate systems and applications to mesh generation. *Internat J Numer Methods Engrg* 1973;7(4):461–77. <http://dx.doi.org/10.1002/nme.1620070405>.
- [59] Golub GH, Welsch JH. Calculation of Gauss quadrature rules. *Math Comp* 1969;23(106):221–30. <http://dx.doi.org/10.2307/2004418>.
- [60] Stephan B, Gallus HE, Niehuis R. Experimental investigations of tip clearance flow and its influence on secondary flows in a 1-1/2 stage axial turbine. In: *ASME turbo expo: Power for land, sea, and air*. American Society of Mechanical Engineers Digital Collection; 2001, p. 1–12. <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, Eitelberg 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] Parablade v1.0. 2020, <http://dx.doi.org/10.5281/zenodo.3894778>.
- [71] Nurbpsy v1.1. 2020, <http://dx.doi.org/10.5281/zenodo.3894755>.
- [72] NumPy v1.18. 2020, <https://numpy.org/>.
- [73] Pagmo/Pygmo v2.15. 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/BF01589116>.
- [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>.