

Research Article

Libor Koudelka*

Radial–axial runner blade design using the coordinate slice technique

<https://doi.org/10.1515/eng-2024-0082>

received April 03, 2024; accepted August 12, 2024

Abstract: There is a lack of information about the geometric description of radial–axial runner blades. The article aims to fill this gap using a method that exploits modern differential geometry for the description and shape modification of the runner blade. Three-dimensional Euclidean space with a curvilinear coordinate system serves as a basic manifold object, and the technique of coordinate slice gives the blade camber surface a submanifold. The camber surface definition given in the article is suitable for interactive design and optimization of the shape in a computer program optimization loop. Geometric entities and maps can be viewed as objects and methods of a computer object-oriented in-house program. The Francis runner blade serves as an example. Complete blade design contains the camber surface wrapped up with an airfoil surface, but this is not dealt with here. Basic knowledge of differential geometry and spline theory is expected.

Keywords: runner blade, manifold, coordinate slice, curvilinear coordinate system, in-house program

Nomenclature

$(.,.)$	Open interval
$[.,.]$	Closed interval
$\langle.,.\rangle$	Scalar product
BEP	Best efficiency point
Bold letter	Vector, vector function, or matrix
C^k	Continuous up to the k -th derivative
D	Runner diameter (m)
$\mathbf{g}_u$	Tangent vector to isoparametric u -curve
H	Net head (m)

$J|\mathbf{g}|$

n

$$n_{11} = \frac{nD}{H^{0.5}}$$

$$n_{sq} = \frac{n}{H^{0.75}}$$

Q

$$Q_{11} = \frac{Q}{D^2 H^{0.5}}$$

R^3

$|\mathbf{v}|$

$w(x, y, z), w(u, v)$

$x_{,u}$

Jacobi determinant of function $\mathbf{g}$

Runner speed (min^{-1})

Double reduced runner speed (min^{-1})

Specific volumetric runner speed ($\text{min}^{-1} \text{m}^{-1/4}$)

Discharge ($\text{m}^3 \text{s}^{-1}$)

Double reduced discharge ($\text{m}^3 \text{s}^{-1}$)

Three-dimensional real vector space

Length of vector $\mathbf{v}$

Two different functions distinguished by number of variables

Partial derivative of function x with respect to variable u

1 Introduction and background

Hydraulic design of a runner can be done in two ways. Either based on geometrical parameters or based on hydrodynamic parameters. The first uses interactive (trial and error) or optimization methods, while the second is defined as inverse design. This article is a contribution to the first one.

Many articles have been published on the topic, but none of them offer an explicit geometric description of the blade geometry. Nevertheless, at least brief information can be found. Over time, as computational fluid dynamics (CFD) became more and more powerful and available, procedures gradually advanced from one dimension to three dimensions. In modern design, CFD simulation techniques play a crucial part in both the interactive (trial and error) design process of turbomachinery and its automatic optimization process.

The one-dimensional (1D) procedure is accompanied by a simpler set of parameters and applications for small turbines. Chen *et al.* [1] investigated the minimum blade passage area at the runner exit (called the *port area* there), while the meridional shape of the runner was inferred from a combination of guide vane loss analysis and experience. Also, blade inlet and outlet angles were determined by experience. The port area adjustment, together with the

* Corresponding author: Libor Koudelka, Section of Water Turbines Development, Strojírny Brno, Kuřim, Czech Republic, e-mail: l.koudelka@strojirnybrno.cz

blade outlet angle, was applied to correct the outflow angle to the draft tube, and in this way, efficiency was improved. All these are for the design operation point. The design quality is evaluated from averaged pressure differences.

The three-dimensional (3D) method is widely used, giving more accurate results. A simple application was presented by Biswakarma and Shrestha [2]. Design is based on basic information such as the Euler turbine equation and computation of blade leading and trailing angles from velocity triangles. Mathematical formulations were written in MATLAB. A 3D model of the blade is then obtained via the requirement of equal energy distribution for all streamlines and subjected to CFD simulation. Another improvement of the blade shape is then necessary since the runner's hydraulic efficiency is 59%. Ayli *et al.* [3] investigated the effect of five parameters (runner outlet diameter, inlet and outlet β -angles, lean angle, and runner speed) on four medium-specific speed runners. The effect of each separate parameter variation on performance and efficiency and finally correlation between rotational speed and flow rate are presented. This research resulted in a universal characteristic for the varied parameters. Ayancik *et al.* [4] proceeded from starting parameters (flow rate, head, and turbine-specific speed) to an initial shape that was inserted into an optimization loop. Leading and trailing angles were determined on five meridional sections. The optimization also contains a mechanical analysis and output to manufacturing. Ayancik *et al.* [5] improved the previous method. The optimization loop is complemented by preliminary design and enhanced by a sub-loop equipped with rough mesh. The result of the study based on CFD simulation is the effect of theoretical runner parameters on the design, cavitation, and efficiency. It is interesting to notice that since the blade pitch angle was not parameterized, the runner blade for low-specific speed has rather a strange shape (Figure 3). A more sophisticated procedure for the design of a large high-speed Francis turbine combining global and local optimization is given in Flores *et al.* [6]. The blade design is included in the global loop. The 3D optimization was model tested to confirm the expected results. The runner blade is described by one cubic NURBS patch generated by an in-house program. To manage a higher-order patch automatically would probably be difficult. This geometric description is rather simple and lacks the possibility of fine modification.

Other researchers concentrated on the improvement of existing runners. An even more specialized application is presented by Takahashi *et al.* [7], where the existing runner was redesigned to get an optimized unshrouded version. The main optimization parameter was the shroud tip clearance and trailing edge shape. Such runners, though having lower efficiency, are cheaper and suitable mainly for small

hydraulic power plants. A conventional approach to runner design is presented in Koudelka [8] for high-specific speed runners used especially in fountain turbines. This proves that conventional methods can give good results. Only by runner replacement the turbine reached a hydraulic efficiency of 91%. Reverse engineering for the original runner was used for the comparison of efficiency and power output between turbines equipped with old and new runners. Celbioglu and Kaplan [9] focused in more detail on reverse engineering and its methodology. The methodology was first applied to the runner design of a new turbine. ANSYS BladeGen[®] and SolidWorks[®] were utilized to obtain analyzable blade geometry from a scanned cloud of points. The geometry was then subjected to CFD analysis to verify that the geometry and performance results are the same as the actual cases. The next step was the redesign of the second runner for improved performance since the power plant utilized only 70% of its installed capacity. The redesign, performed by trial-and-error method, resulted in a smaller blade trailing angle and increased blade thickness distribution. In this way, the full capacity of water was exploited. Agromayor *et al.* [10] made the most recent attempt at the generalization of blade geometry suitable for reverse engineering and possible subsequent CFD analysis. Surfaces are described by NURBS and several tens of parameters. The parameters are unfortunately vaguely defined.

We should mention the inverse design method, which is being developed contemporaneously. It consists namely of two parts: calculation of the flow field and then the geometry of the blade. The process is iterative. Blade geometry is determined according to the previously computed flow field. Then usually, the fully 3D turbulence flow calculation is used as an indispensable tool for evaluation, optimization of the design outcome, and studying the design know-how.

A review of the 3D inverse design method and its applications can be found in Yang *et al.* [11], while the Kaplan runner blade design is presented by Krzemianowski [12]. In this case, streamlines in the volume where the blade operates described by cylindrical coordinates are computed from the velocity vector field. The shape of the blade is derived from the path line of a fluid element. Application of the method and boundary conditions are also presented. The design of the turbine including guide vanes was then subjected to CFD analysis to find the best efficiency point of 88.6%. A similar approach to the design of the Francis runner blade is presented by Krzemianowski and Steller [13] based on the same coordinate system but a different meridional shape. Moreover, the vortex lattice method was exploited to compute streamlines in meridional cross-section. The completed runner blade was then subjected to

optimization to increase efficiency. The optimization was performed in ANSYS Workbench®, resulting in 1% extra efficiency, although the absolute value is not mentioned. An interesting idea of inheritance is presented by Yin *et al.* [14]. The inheritance lies in the extraction of the blade load distribution from a runner with good weighted prototype efficiency (93.84%) and utilizing its design parameters in other designs with similar specific speeds. The method avoids extra CFD work, but the initialization by a quality profile is necessary. The new pump turbine had a weighted efficiency of 94% in pump mode. Ma *et al.* [15] optimized channel geometry, blade loading, and blade stacking to improve the range of operation and increase efficiency. The inverse method was exploited for a high-specific speed turbine runner to get also better pressure distribution on blades since the original ones had cracked after a few years. The latest paper on this topic is by Zanneti *et al.* [16]. They present and discuss the most interesting design solutions so far documented with the main aim to illustrate the results achieved in the field of hydraulic turbines. The basic theory of the inverse design method is presented and analyzed with a focus on the suppression of secondary flows. Then, recent applications of the method to pump turbines and Francis turbines are reported.

Because the main tool for blade construction is a differentiable manifold and its coordinate slice, we provide some basic information on this topic.

1.1 Euclidean space and coordinate slice

We briefly mention the notion of Euclidean space, viewed as a differentiable manifold. Space R^3 viewed as the Euclidean space, is the Hausdorff topological space with only one identity, diffeomorphic map (*chart*) onto its copy, where *diffeomorphism* is a *bijection* of class C^k . Then, it is a C^∞ differentiable manifold. Let M^3 be a subset of space R^3 , where the camber surface will be placed, and $\mu: U \subset M^3 \rightarrow R^3$ be a diffeomorphism called *chart*. Then, μ is the system of curvilinear coordinates. If ν is another chart: $V \subset M^3 \rightarrow R^3$, it is another system of curvilinear coordinates. Compositions $\mu \circ \nu^{-1}$ and $\nu \circ \mu^{-1}$ are transition maps. If both transition maps are C^k , then M^3 is called a C^k differentiable manifold of dimension 3 [17] (Figure 1).

Coordinate slice S of dimension 2 in 3D manifold M^3 in a neighborhood $U \in M^3$ with coordinates u, v , and w is a set $S = \{\mathbf{p} \in U: w(\mathbf{p}) = \text{const.}\}$

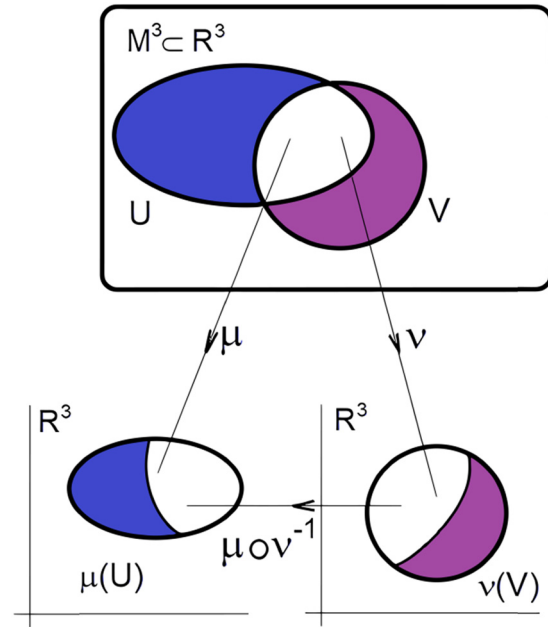


Figure 1: Scheme of manifold mapping.

2 Problem description

From the designer's point of view, the runner of a radial-axial turbine can be viewed as a ring with a meridional cross-section in the shape of a curvilinear rectangle revolved around the axis of runner rotation. The sides of the rectangle are given by inlet, outlet, hub, and shroud curve (Figure 2). Camber surface was added to clarify the geometry. It is not used for meshing. Moreover, the periodicity of the blading is specified during the runner design. Meshing programs are so powerful that it is enough to upload the meridional curves of the rectangle mentioned above, the blade surface, and specify the number of blades to obtain meshed volume (Figure 3), where there is such a meshed volume prepared in Numeca Autogrid®. Now, we can define the problem itself.

2.1 General characteristics

At this point, we can claim that the most reasonable solution for camber surface definition is a parametric surface, say $\mathbf{s}(u, v)$, with the following properties:

1. $\mathbf{s}$ is smooth enough to enable an efficient transformation of fluid energy. $\mathbf{s} \in C^k$, $k > 0$.
2. u -curves connect the leading and trailing edges, while v -curves connect the hub and shroud.
3. Meridional curves of the hub and shroud and the curves of the inlet and outlet edge are at least C^1 .

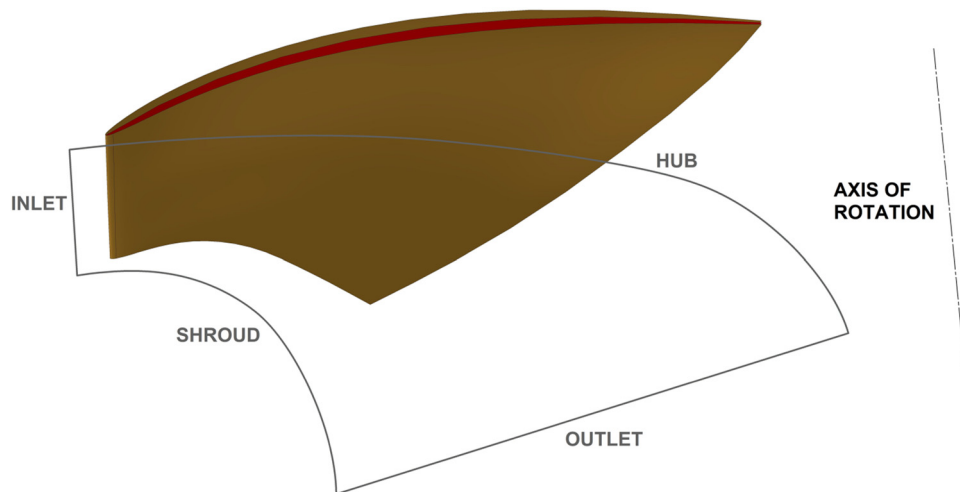


Figure 2: Basic geometry for Numeca Autogrid® mesher with camber surface (red).

4. The leading and trailing angles are specified at the hub and shroud meridional curves and change smoothly along the inlet and outlet edges.
5. If one of these angles is altered, the surface must change smoothly near the leading or trailing edge so that an unwanted wave or even oscillation does not occur.
6. The easiest and most efficient way to specify the blade hub and shroud curve is to map a plane curve into the corresponding surface.
7. Pitch angles for the hub and shroud curve can be easily specified. Blade pitch angles define the runner's specific speed and vice versa.
8. The geometric method and object should comply with the geometric nature of the runner.

Strictly speaking, we are to prepare a meaningful camber surface for an airfoil-shaped hole in the volume of the runner. The situation is depicted in Figures 2 and 3.

2.2 Leading and trailing angle

These are usually referred to as β_1 and β_2 . The geometric meaning is shown in Figure 4. The runner is considered to rotate CW in the turbine mode. To introduce these angles, it is necessary to manipulate the camber surface at the leading and trailing edges. This is done in the $[u, v, w]$ space using function $w(u, v)$ on the slice and mapped in the manifold as a camber surface. The procedure is described in the following example. Analytically, the angles are defined

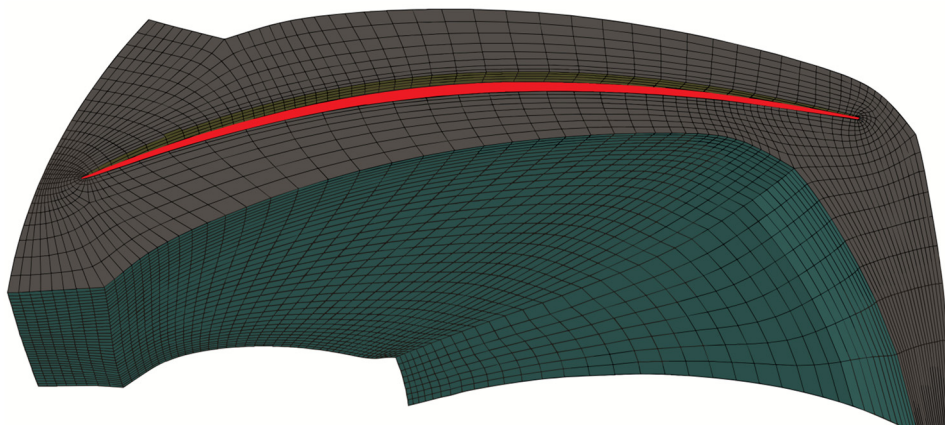


Figure 3: Example of a meshed periodic runner volume with camber surface (red).

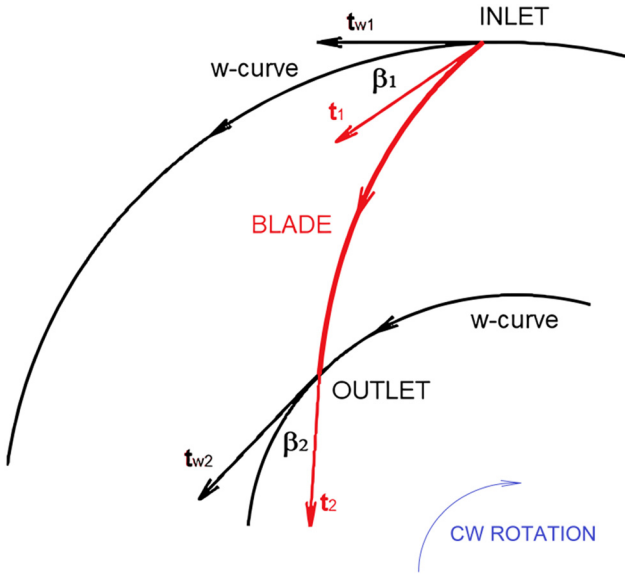


Figure 4: Definition of leading and trailing angles.

using the scalar product of tangent vectors to the camber surface and a w -curve by $\cos(\beta_i) = \frac{\langle \mathbf{t}_i, \mathbf{t}_{wi} \rangle}{\|\mathbf{t}_i\| \|\mathbf{t}_{wi}\|}$, $i = 1, 2$, by Figure 4. Vector directions have nothing to do with a sense of rotation, only with the blade geometry.

3 Method

3.1 Manifold

We work with a 3D manifold $M^3 \in R^3$ given by a system of curvilinear coordinates

$$\mathbf{g}(u, v, w) = \begin{bmatrix} x(u, v, w) \\ y(u, v, w) \\ z(u, v, w) \end{bmatrix}, (u, v, w) \in A, \quad (1)$$

$$A = [u_1, u_2] \times [v_1, v_2] \times [w_1, w_2],$$

such that $\mathbf{g}(u, v, w) \in C^n$, $n \geq 2$ is injective and Jacobi determinant

$$J[\mathbf{g}] = \begin{vmatrix} X_{,u} & X_{,v} & X_{,w} \\ Y_{,u} & Y_{,v} & Y_{,w} \\ Z_{,u} & Z_{,v} & Z_{,w} \end{vmatrix} \neq 0, (u, v, w) \in A. \quad (2)$$

Then, there is a unique chart $\mu: M^3 \rightarrow A$ given by an inverse of $\mathbf{g}(u, v, w)$

$$\mu = \begin{bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{bmatrix}, (x, y, z) \in M. \quad (3)$$

According to Bishop and Goldberg [17], M^3 is a submanifold embedded into R^3 with μ as the only chart. In the words of Section 1.1, A is a subset of the copy of R^3 . To tailor the method to our purpose, we can imbed M^3 in a quite special way as a cylindrical curvilinear coordinate system. This can be done by rotating a plane curvilinear coordinate patch by a specific angle (Figure 5). The plane patch deserves more attention since it naturally reflects the runner's meridional curves, namely the hub, shroud, inlet, and outlet. Rotation by an angle less than π is a diffeomorphic operation.

3.2 Coordinate slice

Now, we refine the definition from Section 1.1 and tailor it to our purpose by setting $U = A$, and we can go even further. We can consider the camber surface in A as a submanifold S embedded by the inclusion map and utilize Proposition 1.4.1 of [17], p. 42, namely its constructive proof. In principle, the proposition says that if S is a submanifold, then there exists a coordinate system in A such that S is a coordinate slice in A . The proof even gives a recipe for its

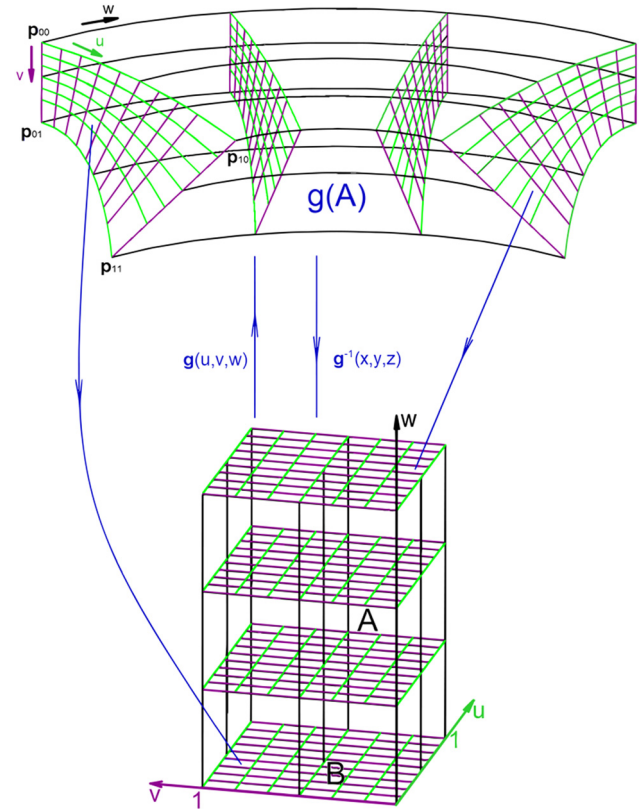


Figure 5: Scheme of the manifold with mapping.

construction. For our purpose, the slice is one dimension less than the sliced manifold. A simple example of the situation can be seen in Figure 6. The submanifold itself is used for the definition of a new coordinate system $[y^1, y^2]$ as follows:

$$\begin{aligned} y^1 &= x^1, \\ y^2 &= x^2 - f(x^1). \end{aligned}$$

S in Figure 6 is a coordinate slice since $y^2 = 0$ on S . A is mapped identically onto its copy and S onto the *projection* of A in the x^1 -axis. We used the designation of the coordinates from the proposition mentioned above.

Using this technique in 3D, we slice the 3D set A in a more general way. Let parametric surfaces $g(u, v_1, w)$, $g(u, v_2, w)$ define the hub and shroud of a runner while $g(u_1, v, w)$, $g(u_2, v, w)$ define the inlet and outlet surfaces, respectively. Now, the camber surface can be represented by a coordinate slice S given by an injective function:

$$w = w(u, v) : B \rightarrow R$$

such that $\text{rank}[w_{,u}, w_{,v}] \neq 0$, $w(u, v) \in C^n$, $n \geq 1$, $(u, v) \in B$, $B = [u_1, u_2] \times [v_1, v_2]$ (Figure 7).

The corresponding chart of the slice into A is

$$\sigma = \begin{bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{bmatrix}. \quad (4)$$

To prove that σ gives a coordinate slice, it is enough to define a coordinate system $[y^1, y^2, y^3]$, again by Proposition 1.4.1 of [17], p. 42.

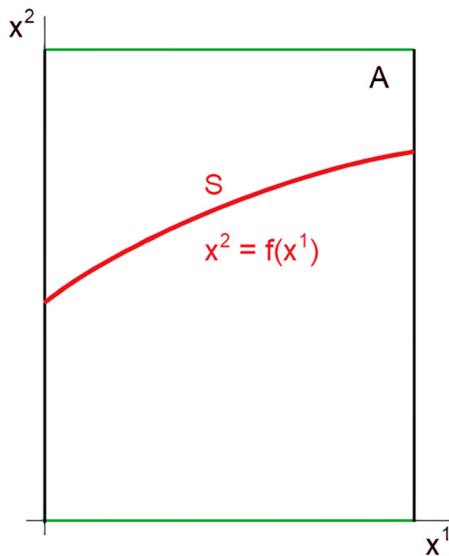


Figure 6: Simple 1D coordinate slice.

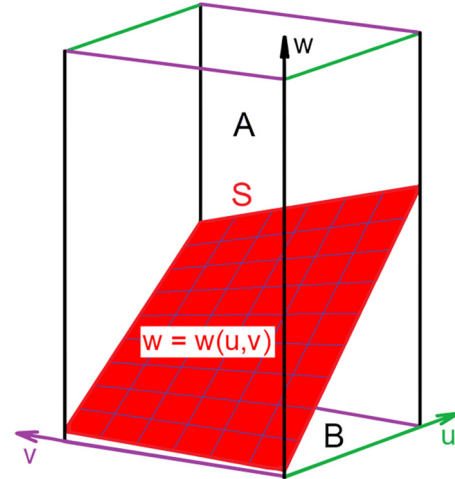


Figure 7: 2D coordinate slice.

$$\begin{aligned} y^1 &= u, \\ y^2 &= v, \\ y^3 &= w - w(u, v). \end{aligned}$$

Rem. Here, $w(x, y, z)$ and $w(u, v)$ represent two different functions distinguished by the number of variables.

This represents a coordinate system in M^3 , and the points of S are those where $y^3 = 0$. This also means that S is given by

$$\begin{aligned} s(u, v) &= \begin{bmatrix} x(u, v, w(u, v)) \\ y(u, v, w(u, v)) \\ z(u, v, w(u, v)) \end{bmatrix} = \begin{bmatrix} s^1(u, v) \\ s^2(u, v) \\ s^3(u, v) \end{bmatrix}, \quad (5) \\ (u, v) &\in B, \text{rank } J[s] = 2, \end{aligned}$$

which is a submanifold of M^3 . The slice is well defined, and its use for camber surface description is correct. In other words, since each curvilinear coordinate system is diffeomorphic with a Cartesian coordinate system, the slice under the conditions given above is well defined.

4 Simple example

The same notation as in Section 3 will be used to obtain the correspondence between both parts. Continuity C^n is meant for $n \geq 1$.

4.1 Coons patch

This method for surface description is based on the framework of its boundary curves [18]. The surface patch $P \subset R^3$ is defined as

$$\mathbf{p}(u, v) = [1 - u] \begin{bmatrix} \mathbf{p}(0, v) \\ \mathbf{p}(1, v) \end{bmatrix} + [p(u, 0)\mathbf{p}(u, 1)] \begin{bmatrix} 1 - v \\ v \end{bmatrix} - [1 - u] \begin{bmatrix} \mathbf{p}_{00} & \mathbf{p}_{01} \\ \mathbf{p}_{10} & \mathbf{p}_{11} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}, \quad (6)$$

$\mathbf{p}: B \rightarrow P, B = [0, 1] \times [0, 1]$.

The injectivity of $\mathbf{p}$ is an additional requirement to the definition given in [18].

Considering Figure 8, we can write Equation (6) in components as follows:

$$x(u, v) = [1 - u] \begin{bmatrix} x(0, v) \\ x(1, v) \end{bmatrix} + [x(u, 0)x(u, 1)] \begin{bmatrix} 1 - v \\ v \end{bmatrix} - [1 - u] \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}, \quad (7)$$

$$z(u, v) = [1 - u] \begin{bmatrix} z(0, v) \\ z(1, v) \end{bmatrix} + [z(u, 0)z(u, 1)] \begin{bmatrix} 1 - v \\ v \end{bmatrix} - [1 - u] \begin{bmatrix} z_{00} & z_{01} \\ z_{10} & z_{11} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \quad (8)$$

For future simplification of Equation (11) and others, we designate

$$f(u, v) = x(u, v) = [1 - u] \begin{bmatrix} x(0, v) \\ x(1, v) \end{bmatrix} + [x(u, 0)x(u, 1)] \begin{bmatrix} 1 - v \\ v \end{bmatrix} - [1 - u] \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \begin{bmatrix} 1 - v \\ v \end{bmatrix}. \quad (9)$$

4.2 Manifold

It is introduced as a curvilinear coordinate system similar to a cylindrical one, where u -curves are given by the hub and shroud meridian, v -curves by inlet and outlet edge, and w -curves are circular arcs with center on runner axis and in interval $[0, \pi]$. With this idea in mind, it is necessary to

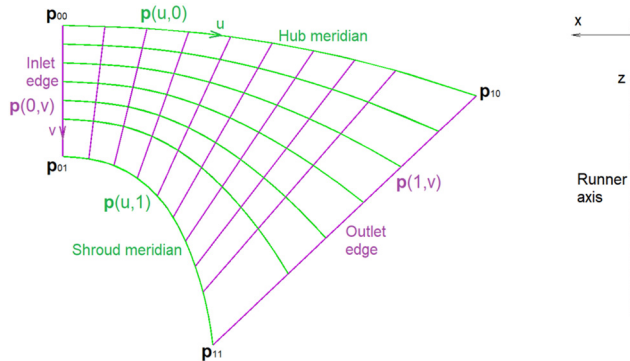


Figure 8: Patched portion of the zx -plane.

proceed from the basic data for runner design. These are curves defining hub and shroud meridian, and inlet and outlet edge. Figure 8 shows the basic net of the u - and v -curves in the zx -plane represented by the corresponding planar Coons patch. We call it the *basic patch* $\mathbf{p}(u, v)$.

$$\mathbf{p}(u, v) : B \rightarrow R^2, \mathbf{p} \in C^n,$$

$$\mathbf{p}(u, v) = \begin{bmatrix} x = f(u, v) \\ z = z(u, v) \end{bmatrix}, J|\mathbf{p}| \neq 0. \quad (10)$$

Now, the *basic patch* is used for the construction of manifold M as a curvilinear coordinate system by

$$\mathbf{g}(u, v, w) = \begin{bmatrix} f(u, v) \cos(w) \\ f(u, v) \sin(w) \\ z(u, v) \end{bmatrix} = \begin{bmatrix} x(u, v, w) \\ y(u, v, w) \\ z(u, v) \end{bmatrix}, \quad (11)$$

$$(u, v, w) \in A, A = B \times [0, \pi],$$

where w is the angular coordinate. Sets A and B are marked in Figure 7.

$$J|\mathbf{g}| = \begin{vmatrix} x_{,u} & x_{,v} & x_{,w} \\ y_{,u} & y_{,v} & y_{,w} \\ z_{,u} & z_{,v} & 0 \end{vmatrix} \neq 0, \quad (12)$$

on A . This is derived from Equations (10) and (11).

In correspondence with Equation (2), this ensures the existence and uniqueness of $\mathbf{g}^{-1}(x, y, z) \in C^n$. The manifold is mapped onto A by

$$\mathbf{g}^{-1}(x, y, z) = \begin{bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{bmatrix}. \quad (13)$$

This is the only map. It is a C^n manifold of dimension 3 with a boundary, where n depends on the continuity of the *basic patch* boundary curves. It can be observed also in Figure 5 that $\mathbf{g}$ is injective. The isoparametric curves are not tangent to each other.

Rem. Tangent space at a point has the basis

$$\mathbf{g}_u = \begin{bmatrix} f_{,u} \cos(w) \\ f_{,u} \sin(w) \\ z_{,u} \end{bmatrix}, \mathbf{g}_v = \begin{bmatrix} f_{,v} \cos(w) \\ f_{,v} \sin(w) \\ z_{,v} \end{bmatrix}, \quad (14)$$

$$\mathbf{g}_w = \begin{bmatrix} -f \sin(w) \\ f \cos(w) \\ 0 \end{bmatrix}.$$

4.3 Coordinate slice

We define the coordinate slice in the same manner as described in Section 3.2. Then, Equation (5) has the following form:

$$\mathbf{s}(u, v) = \begin{bmatrix} f(u, v) \cos(w(u, v)) \\ f(u, v) \sin(w(u, v)) \\ z(u, v) \end{bmatrix}. \quad (15)$$

Thus, $\mathbf{s}(u, v)$ is a regular surface and can be considered as embedded in M by inclusion. This situation is shown in Figure 9.

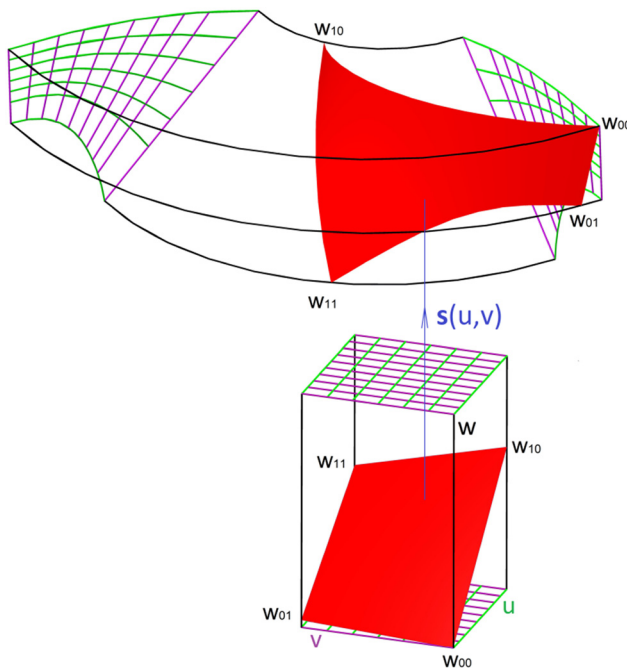


Figure 9: Camber surface as a coordinate slice.

4.4 Manipulating leading and trailing angle

We work in A with isoparametric curves $w(u, 0)$ and $w(u, 1)$. They connect vertices w_{00} , w_{10} , and w_{01} , w_{11} . The curves can be defined as cubic polynomials with boundary conditions listed in Table 1. Coefficients of the polynomials are easily calculated from boundary conditions

$$\begin{aligned} \beta_{ij} &\in (0, \Pi/2), \varphi_{ij} = \Pi/2 - \beta_{ij}, w(i, j) = w_{ij}, \\ w'(i, j) &= tg(\varphi_{ij}), i, j = 0, 1. \end{aligned} \quad (16)$$

The meaning of these angles is shown in Figure 10. These curves are then mapped in S . Because the mapping

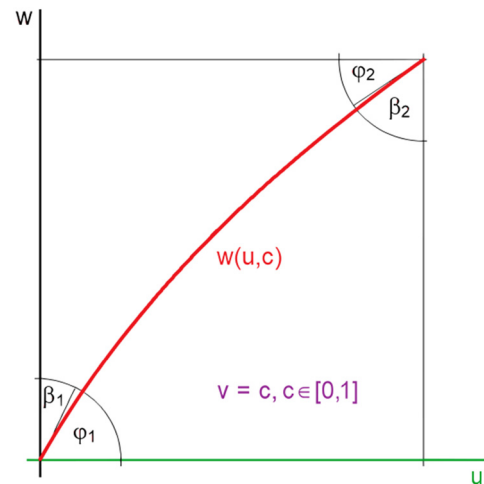


Figure 10: Section of the slice in A at $v = \text{const.}$ with definition of the corresponding leading and trailing angles.

is not conformal, it is necessary to compute real β_1 and β_2 angles, as defined in Section 2.2, and step-by-step change φ_{ij} to obtain the required values. This can be performed in the cycle of a computer language program. Function $w(u, v)$ to be inserted into Equation (12) is

$$w(u, v) = w(u, 1)v + w(u, 0)(1 - v). \quad (17)$$

Figure 11 shows the same camber surface as in Figure 9 but with modified φ_{ij} and corresponding β_{ij} . The shape modification is rather excessive, to be evident.

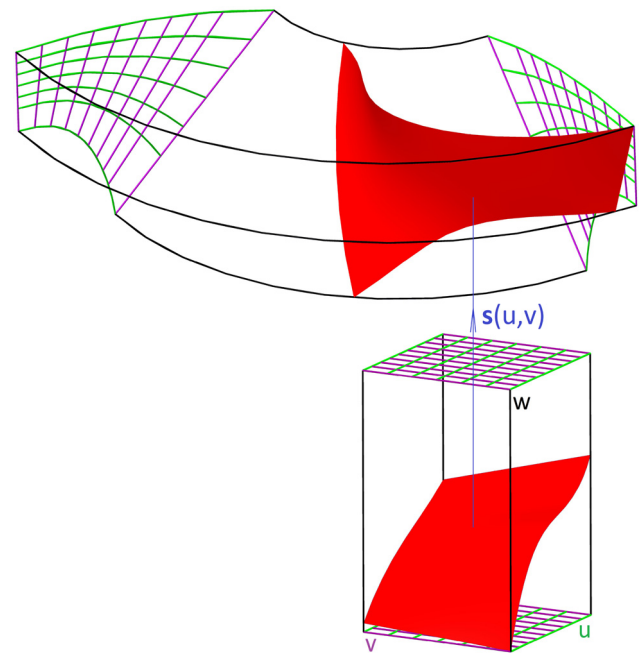


Figure 11: Camber surface with modified β -angles.

Table 1: Boundary conditions

Curve	Position	Angle	Leading and trailing angle			
$w(u, 0)$	w_{00} w_{10}	φ_{00} φ_{10}	β_{10}	β_{20}		
$w(u, 1)$	w_{01} w_{11}	φ_{01} φ_{11}	β_{11}	β_{21}		

Now, having explained the idea and basic strategy, we can approach generalization.

5 Generalization

Since Equation (15) together with sine and cosine functions on $[0, \pi]$ is C^1 and injective, we claim that any C^1 slice in space (u, v, w) is mapped on a C^1 camber surface. Now, the idea is simple. We can patch the slice consisting of its boundary curves and as many as necessary $v = \text{const.}$ curves. The patch including curves with properly set β -angles is then mapped as the camber surface. In this way, the hydraulic designer can specify a blade with pitch and β -angles on those $v = \text{const.}$ meridional curves.

We keep the idea of Section 4.4 where camber curves were defined as cubic polynomials. With this in mind, particularly useful is the bicubic blending. Figure 12 shows the method that offers to set four meridional curves, which require three patches. Detailed patching procedure of the coordinate slice in set A is given in [18], p. 231–234. This also shows that the method is well-defined. The table with boundary conditions would be similar to Table 1 but with four rows. This enables finer shape modification like bent of the trailing edge (black curves in the slice and camber surface).

If necessary, the camber surface can be fitted as a NURBS surface (Figure 13).

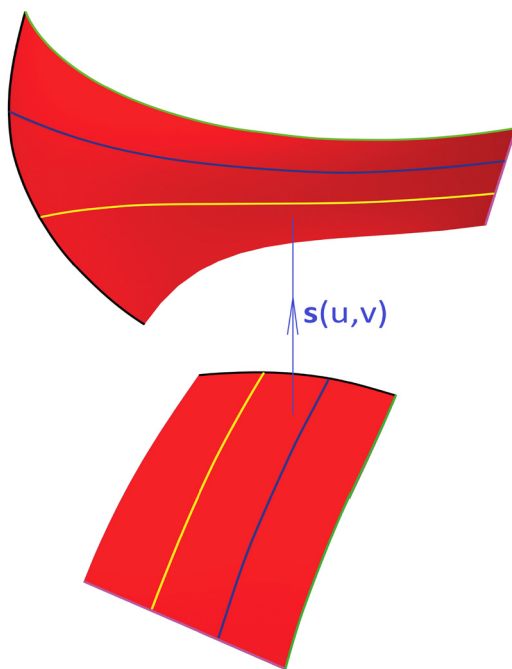


Figure 12: Bicubic blended coordinate slice in R^3 and corresponding camber surface.

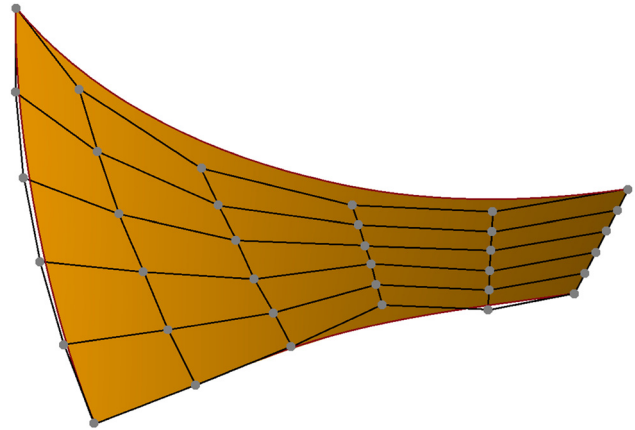


Figure 13: Quintic NURBS representation of the camber surface.

6 Practical example and physical meaning

The above method was used for the design of the high-specific speed Francis runner presented in Koudelka [8]. A detailed design process is described there. Nevertheless, basic hydraulic and geometric parameters are listed in Table 2 and depicted in Figure 14. The data in Table 2 are inferred from those given in Koudelka [8].

Since the runner for such speed is higher, there must be six $v = \text{const.}$ Curves in R^3 to sufficiently describe the camber surface (Figure 15).

To describe the situation thoroughly, Figure 16 shows the camber surface (red) inside the manifold boundaries formed by the hub and shroud (green) together with inlet and outlet (violet) surfaces.

The basic physical meaning of the blade is to take effectively the static component of the water pressure, transform it into the runner torque, and subsequently produce turbo-generator power. The process of draining static

Table 2: Hydraulic parameters

Parameter	Value
Runner diameter D	0.35 m
Number of blades	11
Runner speed n	470 min^{-1}
Number of guide vanes	10
Net head H	4.6 m
Rated discharge Q	$0.32 \text{ m}^3 \text{ s}^{-1}$
Q_{11} at BEP	1.23
n_{11} at BEP	75.1 min^{-1}
n_{sq} at BEP	$84.7 \text{ min}^{-1} \text{ m}^{-1/4}$
Hydraulic efficiency at BEP	91%

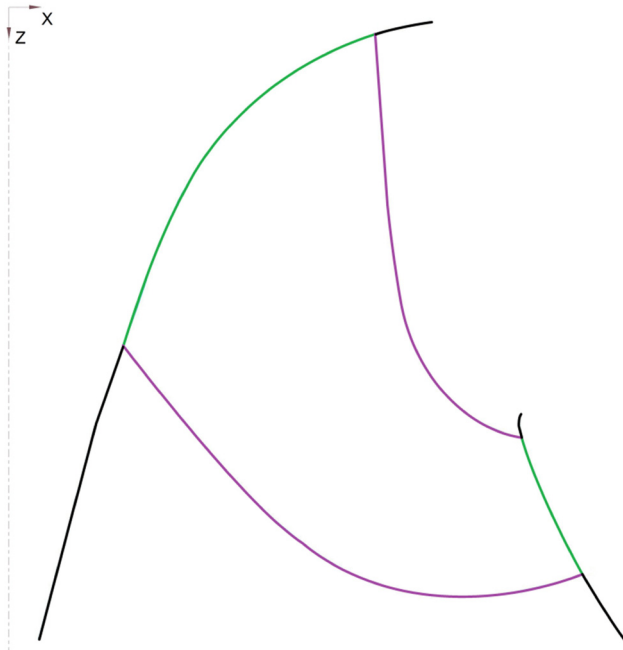


Figure 14: Runner meridian with borders of the basic patch.

pressure is nicely seen in Figure 17. Pressure decreases from the leading to the trailing edge as the blade sucks it gradually.

This process is the most effective when flow splits just on the leading edge, also indicated by the red pressure strip in Figure 18. This also indicates the best efficiency point.

A photograph of the resulting runner in steel can be seen in [8].

7 Discussion and conclusions

A straightforward step toward an object-oriented programming language is possible. Objects include surfaces, curves, vectors, *etc.*, with methods for manipulation, description, and display. The runner is then inserted into a turbine model, and this virtual prototype is subjected directly to CFD analysis or an optimization loop.

7.1 Why the coordinate slice?

The question may arise as to why we do not define a surface patch such as NURBS in the bounds of hub, shroud, inlet, and outlet surfaces [19]. Here are some difficulties coming from the nature of this technique:

1. It is difficult to insert such a patch inside the runner volume with specified camber surface pitch angles, namely positions

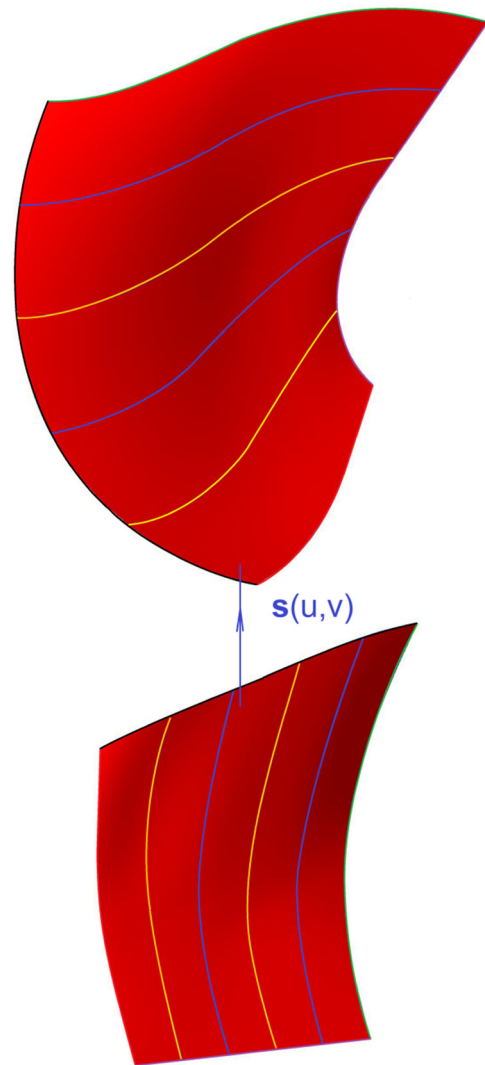


Figure 15: Bicubic blended coordinate slice in R^3 and corresponding camber surface with six curves.

of control points, since each of them has three degrees of freedom. Moreover, the surface would consist of more than one patch or a patch of a higher degree than three, as can be seen from Section 5.

2. The danger of local change of the surface, namely when a higher order (rational) polynomial is used. This can result in unwanted waves and does not comply with items 1 and 5 (Section 2.1).
3. Simple modification of β -angles by the control polygons in space is questionable. That is why Flores *et al.* [6] use the simplest reasonable NURBS patch for the purpose, which is the cubic patch.
4. The subset of R^3 defining the runner is described by a cylindrical coordinate system. This follows from the fact that the runner rotates around its axis, here coincident with the z -axis. If this approach is not applied, then the

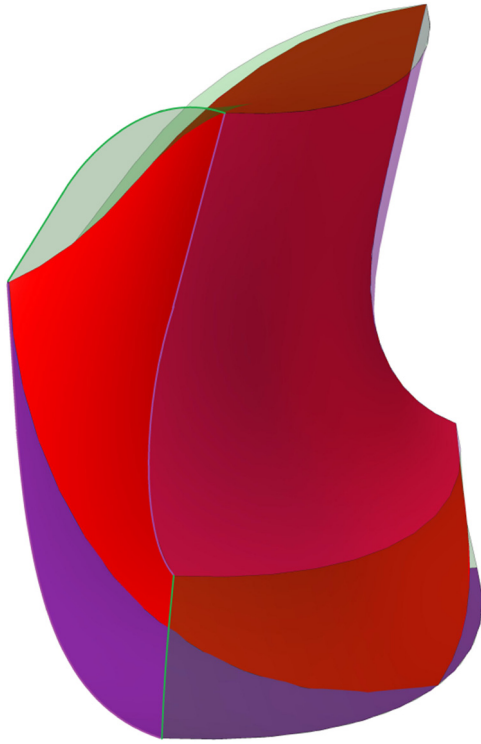


Figure 16: Blade in the boundaries.

problem of pitch angle specification arises. These angles define runner speed and vice versa. A problem with a deformed blade may emerge like in [5].

5. Techniques based on *classic* patches (NURBS can be considered to be their generalization) are suitable,

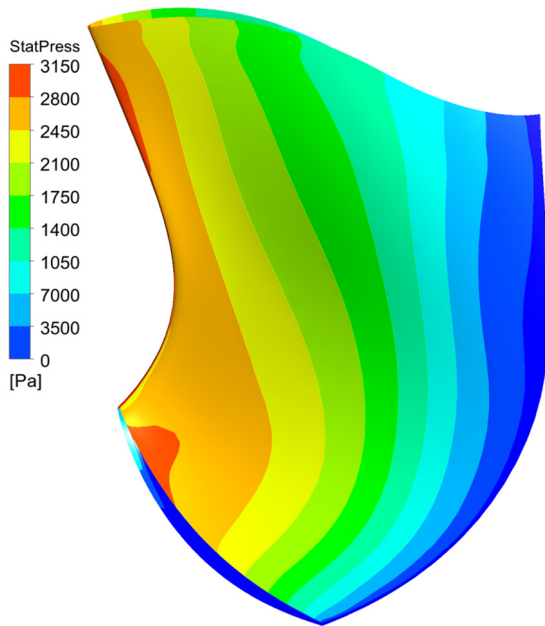


Figure 17: Pressure field on the blade pressure side at BEP.

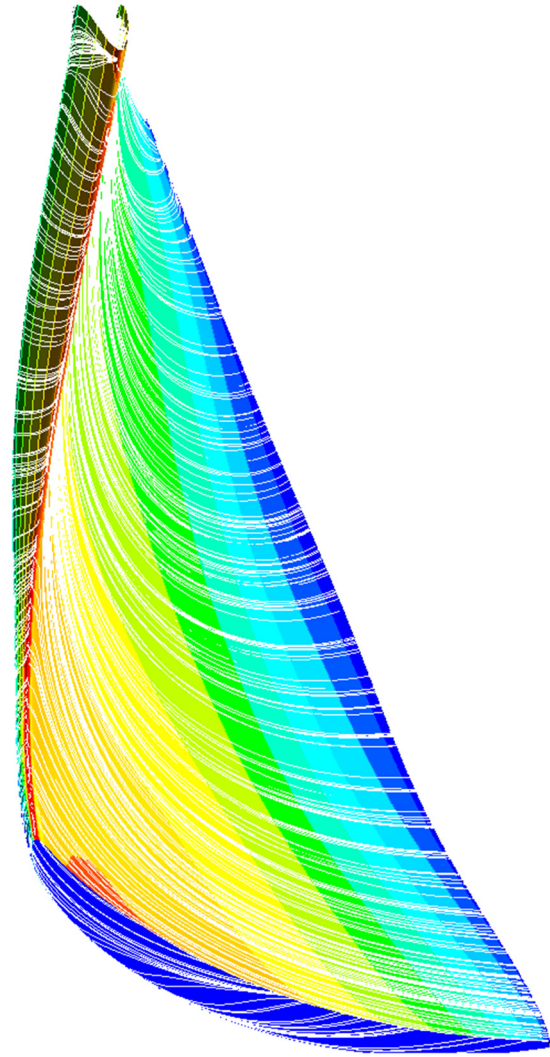


Figure 18: Flow lines splitting at the leading edge at the BEP.

namely for free-form modeling. Reasonable utilization in blade design requires a patch with a small number of control points to handle the shape either in an interactive or optimization loop process. That is why only cubic patch is used in [6]. Usage of patches of higher degree may lead to unwanted oscillations. See also Figure 13, where the NURBS control polygon was generated just after design in the way of the coordinate slice. The coordinate slice is easily defined, including the construction of the planar curve mapped as a camber curve and easy mapping into the manifold, which complies with runner geometry.

Conversely, the coordinate slice technique respects the nature of the blade geometry. One can, in a natural geometric way, define all important blade parameters as follows:

1. Meridional shape of hub, shroud, inlet, and outlet edge by corresponding (spline) curves.
2. Pitch, leading, and trailing angles by straight assignment of the value for specific camber curves.
3. Lean angle by varying the w -parameter along the blade and, in this way, also the curved shape of the leading and trailing edge and blade passage area at the runner exit.
4. Since the camber surface is regular, it is possible to modify it in the normal direction reasonably.
5. The *inheritance method* mentioned in [13] is obvious here. This time geometric parameters are inherited.

Rem.

The method is also suitable for low-specific speed runners, as indicated in Section 5. To the author's knowledge, this method may not be useful for the inverse problem design.

7.2 Further development

The next step to improve the presented method lies in conformal mapping of the slice isoparametric curves. This leads to avoiding the numerical computation of meridional curves that have required leading and trailing angles.

Funding information: Author states no funding involved.

Author contributions: The author confirms the sole responsibility for the conception of the study, presented results and manuscript preparation.

Conflict of interest: Author L.K. is an employee of Strojírny Brno.

Data availability statement: The data generated during and/or analysed during the current study are available from the author on request.

References

- [1] Chen Z, Singh PM, Choi Y. Francis turbine blade design on the basis of port area and loss analysis. *Energies*. 2016;9:164. doi: 10.3390/en9030164.
- [2] Biswakarma BB, Shrestha R. Mathematical modeling for the design of Francis runner. *Proceedings of IOE Graduate Conference*. vol. 5, 2017. p. 67–74.
- [3] Ayli E, Celebioglu K, Aradag S. Determining and generalization of the effects of design parameters on Francis turbine runner performance. *Eng Appl Comput Fluid Mech*. 2016;10:545–64. doi: 10.1080/19942060.2016.1213664.
- [4] Ayancik F, Aradag U, Ozkaya E, Celebioglu K, Unver O, Aradag S. Hydroturbine runner design and manufacturing. *Int J Mater Mech Manuf*. 2013;1:162–5. doi: 10.7763/IJMMM.2013.V1.35.
- [5] Ayancik F, Celebioglu K, Aradag S. Parametrical and theoretical design of a Francis turbine runner with help of computational fluid dynamics. *International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics*. Vol. 10, 2014. p. 775–80. doi: 10.13140/2.1.3604.7683.
- [6] Flores E, Bornard L, Tomas L, Liu J, Couston M. Design of large Francis turbine using optimal methods. *IOP Conf Series: Earth Environ Sci*. 2012;15:022023. doi: 10.1088/1755-1315/15/2/022023
- [7] Takahashi W, Shinji M, Liu ZH, Miyagawa K, Hayashi Y. Design optimization of medium specific speed Francis turbines with unshrouded runners. *IOP Conf Series: Earth Environ Sci*. 2019;240:1–12. doi: 10.1088/1755-1315/240/2/022060.
- [8] Koudelka L. Do not be afraid of small high-speed Francis turbines. *de Gruyter J Mech Eng*. 2018;3:111–28. doi: 10.2478/scjme.2018.0030.
- [9] Celebioglu K, Kaplan A. Development and implementation of a methodology for reverse design of Francis turbine runners. *Pamukale Univ J Eng Sci*. 2019;25(4):430–9. doi: 10.5505/pajes.2018.43959.
- [10] Agromayor R, Anand N, Muller JD, Pini M, Nord LO. A unified geometry parametrization method for turbomachinery blades. *Comput Des*. 2021;2021:133. doi: 10.1016/j.cad.2020.102987.
- [11] Yang W, Liu B, Xiao R. Three-dimensional inverse design method for hydraulic machinery. *Energies*. 2019;12:1–19. doi: 10.3390/en12173210
- [12] Krzemianowski Z. Engineering design of low-head Kaplan hydraulic turbine blades using the inverse problem method. *Bull Pol Acad Sci: Tech Sciences*. 2019;67(6):1133–47. doi: 10.24425/bpasts.2019.130888.
- [13] Krzemianowski Z, Steller J. High specific speed Francis turbine for small hydro purposes – design methodology based on solving the inverse problem in fluid mechanics and the cavitation test experience. *Renew Energy*. 2021;169:1210–28. doi: 10.1016/j.renene.2021.01.095.
- [14] Yin J, Li J, Wang D, Wei X. A simple inverse design method for pump turbine. *IOP Conf Series: Earth Environ Sci*. 2014;22(1):012030. doi: 10.1088/1755-1315/22/1/012030.
- [15] Ma Z, Zhu B, Shangguan Y. Comprehensive hydraulic improvement and parametric analysis of a Francis turbine runner. *Energies*. 2019;12:1–20. doi: 10.3390/en12020307.
- [16] Zannetti G, Siviero M, Savazzini G, Santolini A. Application of the 3D inverse design method in reversible pump turbines and francis turbines. *Water*. 2023;15:2271. doi: 10.3390/w15122271.
- [17] Bishop RL, Goldberg SI. *Tensor analysis on manifolds*. Vol. 1968. USA: Dover Publications; 1968. p. 40–55.
- [18] Faux ID, Pratt MJ. *Computational geometry for design and manufacture*. Vol. 1980. UK: Ellis Horwood; 1980. p. 231–4.
- [19] Piegl L, Tiller W. *The NURBS book*. Germany: Springer; 1997. p. 81–138.