Design optimization of a 2D blade by means of milling tool path

Christian Vessaz, Christophe Tournier, Cécile Munch, François Avellan

Design optimization of a 2D blade by means of milling tool path

Christian Vessaz[a], Christophe Tournier[b,*], Cécile Münch[c], François Avellan[a]

[a] *EPFL, École polytechnique fédérale de Lausanne, Laboratory for Hydraulic Machines, avenue de Cour 33 bis, 1007 Lausanne, Switzerland*
[b] *LURPA, ENS Cachan, Université Paris Sud 11, 61 avenue du Président Wilson, 94235 Cachan, France*
[c] *HES SO Valais, Haute école spécialisée de Suisse occidentale, Institut Systèmes industriels, Route du Rawyl 47, 1950 Sion, Switzerland*

Abstract

In a conventional design and manufacturing process, turbine blades are modeled based on reverse engineering or on parametric modeling with Computer Fluids Dynamics (CFD) optimization. Then, only raises the question of the manufacturing of the blades. As the design does not take into account machining constraints and especially tool path computation issues in flank milling, the actual performance of the machined blade could not be optimal. In this paper, a new approach is used for the design and manufacture of turbine blades in order to ensure that the simulated machined surface produces the expected hydraulic properties. This consists in the modeling of a continuous tool path based on numerical simulation rather than the blade surface itself. Consequently, this paper aims at defining the steps of the proposed design approach including geometrical modeling, mesh generation, CFD simulation and genetic optimization. The method is applied on an isolated blade profile in a uniform water flow and results are compared to the conventional design process.

Keywords: Design for manufacturing, Tool path, Flank milling, Genetic optimization, Blades

*Corresponding author Tel.: +33 1 47 40 29 96; Fax: +33 1 47 40 22 11
Email address: christophe.tournier@ens-cachan.fr (François Avellan)

1. Introduction

The design and machining process of blade surfaces consists in three steps including Computer-Aided Design (CAD) modeling with CFD optimization of the blade, tool path computation in the Computer-Aided Manufacturing (CAM) software and machining of the part on a machine tool. In this process, the CAD model is the reference model used by any other applications and it usually relies on parametric curves and surfaces with geometrical continuity properties.

From a geometrical point of view, blade modeling is mostly performed by the interpolation of a succession of 2D profiles along the spanwise direction. These 2D profiles are computed whether by the interpolation of a set of sampling points computed by CFD software or directly by the use of parametric curves. The main objectives of the parametric model are to ensure continuity between the curves defining the 2D contour and to reduce the number of geometrical parameters.

However, from a digital mock-up point of view, the degree of polynomial curves and continuity between curves in the parametric model does not matter. Indeed, the geometry of the CAD model is approximated by all the CAX applications based on this model: CFD and Finite Element Analysis (FEA), which generate meshes, visualization using tessellation and even tool path computation, which discretizes surfaces into curves and curves into points as developed hereafter.

During the CAM stage, a set of tool postures (tool position and tool axis orientation) is computed according to a machining strategy [1]. Depending on the design surface, machining is performed in 3 or 5-axis and in flank or end milling. Usual CAM system algorithms for tool path computation rely on the linear format, which is common in the machine tool Numerical Controllers (NC). The surface to be machined is discretized into a set of curves and each curves is discretized into polylines generating geometrical deviations compared to the design. Furthermore, tool paths contain tangency discontinuities which leads to slowdowns during machining and marks on the part due to chip section vari-

ations during machining. Post-processors may convert linear tool paths into polynomial curves. The polynomial trajectory is thus called off-line polynomial trajectory [2] in opposition to on-line polynomial trajectory that corresponds to polynomial trajectory calculated in real-time by the NC unit [3]. In both methods, the polyline is interpolated by polynomial curves such as Bspline curves according to a geometrical tolerance specified by the user. Consequently, in the classical design approach, hydraulic optimization of the blade geometry can provide an optimal design X_d^*, which probably could not be machined without geometrical deviations.

To quickly design blades taking into account their machinability, a new paradigm consists of placing the machining tool path on the heart of the design process [4]. A continuous polynomial tool path is computed so that the envelop of the tool movement, not the CAD model, is optimum with respect to the hydraulic performances. Then, the optimization validates the surface geometry resulting from a kinematical simulation of the machining, computed with the Nbuffer method, and CAD model is the 3D representation of the simulation. Manufacturing should be faster, geometric quality and perceived quality are improved due to the use of a native polynomial tool path and hydraulic performance should better meet expectations. Therefore, the tool path is modeled as a native set of Bspline curves and considered as the reference model. CAD model and CFD/FEA models are then built on these curves through machining simulation (Table 1).

Table 1: Geometry modeling for the different approaches

Method	**CAD**	**FEA/CFD**	**CAM**	**Post-Pro**	**CNC**
Conventional	**Curves**	Mesh	Points	Points	Points
On line	**Curves**	Mesh	Points	Points	Curves
Off line	**Curves**	Mesh	Points	Curves	Curves
Proposed	Mesh	Mesh	**Curves**	Curves	Curves

In the proposed approach, the optimal tool path X_t^* is necessarily machinable, from a kinematical point of view, but the hydraulic performance will be different from that obtained with the conventional approach X_d^*. Indeed, there is an infinite geometrical solutions to define the blade surface and only one hydraulic optimal solution. As soon as the parametric models to define the design or the tool path are stated, the range of solutions is reduced as well as the probability of finding the ideal solution. Moreover, as both approaches generate different geometric models, they do not necessarily share the same solution space. Thus, numerical investigations has to be performed to compare hydraulic performances of both approaches.

The real challenge would consist to compare both approaches in the case of 5-axis flank milling of 3D blades. Indeed, this process presents a high removal material rate a better surface roughness [5]. This process is now widely investigated for the machining of slender complex parts like impellers or turbine blades but it generates undercut and overcut as blades are non developable ruled surfaces or free-form surfaces. Extensive works have been carried out to reduce overcuts and undercuts with cylindrical cutters [6,7,8], conical cutters [9,10] and barrel cutters [11,12]. Still, geometrical deviations are not removed and futhermore, the link between those deviations and the hydraulic performances of the blades has not been investigated.

However, full 3D modeling for twisted blades made of non developable surfaces requires a lot of tests and expensive computational time to setup the process and the evidence of the success of the proposed approach is not guaranteed. The purpose of this paper is thus to set-up and compare the classical and proposed paradigm to design hydraulic profiles made with developable ruled surfaces. Nevertheless, the use of polynomial curves such as Bezier curves to model the designed profile and the tool path makes this test case relevant. Indeed, the ideal tool path for machining the profile is an offset curve of the profile which equation is known as a rational function. As it is impossible to model the offset curve of a polynomial curve by any polynomial curve, we therefore are dealing with a case which is identical to the machining of non-developable ruled

surfaces where X_d^* and X_t^* solutions may be close but will never be identical.

The paper is organized as follows: in section 2, a parametric model is proposed based on a literature review of 2D profile parameterization. Then, the optimization process is presented in section 3, which includes geometrical modeling for both approaches, the CFD simulation and the genetic optimization loop. In section 4, numerical investigations are performed on a single blade in order to compare both approaches.

2. 2D blades modeling

Two main approaches to design blades are found in the literature: the Shape Inverse Design (SID) and the Global Shape Optimization (GSO) [13,14]. In the SID methods, the designer usually starts from an initial blade geometry and performance and inputs the desired modifications to the performance. SID methods need few iterations to generate a new shape that duplicates the desired surface flow parameters. However, this method requires an initial blade geometry computed using a direct approach. In opposition, GSO methods consist in modifying the geometry of the blades until the flow performance is achieved. Methods based on GSO have been developed more recently due to the required computational effort to test and optimize a set of parameters.

2D profiles are computed either by the interpolation of a set of sampling points computed by CFD software [15] or directly by the use of parametric curves. Two main variants exist to define the parametric curves [16]. The *Camber-Curve + Thickness* technic and the *Direct Profiling* technic. The camber-curve defines a skeleton and the thickness distribution defines the sides of the profile [17]. This kind of modeling is not well appropriate to define a 2D profile in parametric CAD systems if implicit formulation of the thickness law or of the blade turning angle are used as in [18,19]. A typical example is the definition of NACA 2D profiles, which are imported as sampling points in CAD systems and then re-interpolated.

In the method proposed by Koini et al. [20] the camber line is described

by a NURBS curve of second degree with three control points (rational Bezier curve). The suction and pressure sides are modeled as Bspline curves defined by control points or interpolation points. The position of these points is defined by a curvilinear abscissa and a normal distance to the camber line.

Different works have been published using the direct profile modeling [13,21,24]. The main objective is to ensure continuity between the curves defining the 2D contour and to reduce the number of geometrical parameters. Pritchard [21] proposed a geometrical model based on polynomial curves in 1985. The 2D profile consists in five concatenated curves connected in tangency (C^1). The advantage of this model is the reduced number of parameters (9).

The model proposed by Pierret et al. [22] consists in three Bezier curves and a circle to design the trailing edge. The connection between the pressure and the suction side at the leading edge ensures a continuous curvature.

Anders et al. [23] propose to model the 2D profile by using two Bezier curves of degree 5 for the suction and pressure sides and circles or ellipses to describe the leading and trailing edges. Tangency continuity is ensured at the junction but not curvature continuity. This model leads to 20 parameters.

Other methods use only polynomial curves to define the 2D profile. The method proposed by Buche et al. [24] consists in modeling the profile with four polynomial Bezier curves of degree 5 connected in curvature. 16 of the 48 required parameters are set by imposing curvature continuity at the junctions. The remaining 32 parameters are translated into engineering parameters and some of them are set to default values leading finally to 19 variable parameters.

In [25], Giannakoglou used two Bezier curves of degree 5 to model the pressure side and the suction side as well as the leading (LE) and trailing (TE) edges. Both curves shared the same starting and ending points at the LE and TE and ensure tangency continuity at the LE. This parameterization yields to 14 parameters.

Goel [26] also used two Bezier curves to model the pressure side and the suction side as well as the leading and trailing edges of thick airfoils for high pressure turbines.

Table 2: Number of design parameters

Method	**Number of parameters**
Pritchard 1985	9
Giannakoglou 2002	14
Goel 2009	14
Pierret et al 1999	15
Dennis et al 1999	17
Buche et al. 2003	19
Anders et al 2008	20
Korakianitis et al. 1993	35
Koini et al. 2009	Unspecified

The method proposed by Ghaly et al. [15] consists in approximating an initial 2D profile by a single Nurbs curve and optimizing the design by moving the Nurbs control points. The degree of the curve is not specified.

In [13], the profiles of suction and pressure surfaces are modeled as two splines of degree 4. The leading edge is treated separately as a line + thickness distribution. The construction ensures a third derivative continuity with the suction and pressure splines.

Many combinations of curves are presented to model 2D profiles of turbomachines. Among the criteria for discriminating methods, the order of continuity at the connection between the curves and the number of design parameters is considered (Table 2). Indeed, fewer parameters lead to shorter computation time but reduce the variety of possible blade shapes.

In the framework of the optimization of hydraulic turbines or impellers, the method proposed by Koini et al. [20] is adopted to design the blades with a camber line and a thickness law to define the suction and pressure sides. The first advantage is the use of a camber line, which shape is directly linked to hydraulic parameters such as inlet and outlet angles (β_1 and $\beta_{\bar{1}}$). This method also presents the advantage of an explicit parametric modeling, which is suited to model the tool path as polynomial curves as well as the design in CAD

softwares. The camber line is defined as a Bezier curve $C_L(u)$ (Eq. 1) of degree 2 ($m = 2$) defined by three control points, that is to say 6 parameters.

$$C_L(u) = \sum_{i=0}^{m} B_{im}(u) \cdot CL_i \text{ with } u \in [0, 1] \tag{1}$$

The first control point CL_0 is located on the leading edge and the last one CL_2 on the trailing edge (Fig. 1). The middle point CL_1 is located at the intersection between the lines defined with the inlet and outlet angles β_1 and $\beta_{\bar{1}}$. If a point of the camber line is considered as fixed, only 4 parameters are required to define it (3 angles and 1 length).

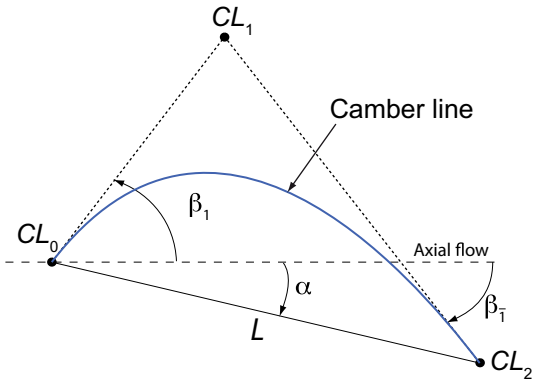

Figure 1: Camber line modeling with Bezier curve

Suction and pressure sides are defined by two Bezier curves of degree 4, $S(t)$ and $P(t)$, connected at the leading edge. Each control point S_i or P_i of the suction and pressure sides (except for $i = 3$) is defined by the abscissa u on the camber line and the distance d (Fig. 2).

$$S_i = C_L(u_{si}) + d_{si} \cdot \boldsymbol{n} \tag{2}$$

$$P_i = C_L(u_{pi}) + d_{pi} \cdot \boldsymbol{n} \tag{3}$$

The trailing edge, which is defined by the segment S_4P_4, is sharp to model real hydraulic blades. The full model is entirely defined by a maximum of 24

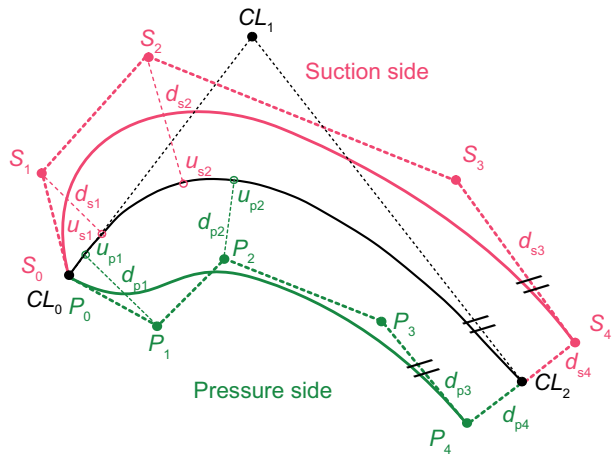

Figure 2: Proposed model

parameters. However, the start points $(S_0; P_0)$ and the end points $(S_4; P_4)$ of both suction and pressure curves are located at the beginning and at the end of the camber line. This condition ensures a G^0 continuity at the leading edge. This leads to:

$$u_{s0} = u_{p0} = 0 \qquad \text{and} \qquad d_{s0} = d_{p0} = 0 \tag{4}$$

$$u_{s4} = u_{p4} = 1 \tag{5}$$

Furthermore, the points $(S_3; P_3)$ are located on lines parallel to the tangent to the camber line at CL_2 to respect outlet angle along the profile. They are defined by two parameters:

$$d_{s3} = S_3 S_4 \tag{6}$$

$$d_{p3} = P_3 P_4 \tag{7}$$

Consequently, the maximum number of parameters is equal to 16. Depending on the continuity at the connection between suction and pressure sides at the

leading edge, the number of parameters decreases as mentioned in Table 3. In the following, the suction side is considered as the anchor curve to define the pressure side.

Table 3: Total number of parameters vs continuity at the leading edge

Continuity	**Suction side**	**Pressure side**	**Total**
G^0	[0 u_{s1} u_{s2} x 1] [0 d_{s1} d_{s2} d_{s3} d_{s4}]	[0 u_{p1} u_{p2} x 1] [0 d_{p1} d_{p2} d_{p3} d_{p4}]	16
G^1	[0 0 u_{s2} x 1] [0 d_{s1} d_{s2} d_{s3} d_{s4}]	[0 0 u_{p2} x 1] [0 d_{p1} d_{p2} d_{p3} d_{p4}]	14
C^1	[0 0 u_{s2} x 1] [0 d_{s1} d_{s2} d_{s3} d_{s4}]	[0 0 u_{p2} x 1] [0 x d_{p2} d_{p3} d_{p4}]	13
G^2	[0 0 u_{s2} x 1] [0 d_{s1} d_{s2} d_{s3} d_{s4}]	[0 0 x x 1] [0 d_{p1} k_2 d_{p3} d_{p4}]	13
C^2	[0 0 u_{s2} x 1] [0 d_{s1} d_{s2} d_{s3} d_{s4}]	[0 0 x x 1] [0 x x d_{p3} d_{p4}]	11

G^1 continuity at the leading edge is achieved if there exists a scalar $k_1 > 0$ so that $\dot{P}(0) = k_1 \cdot \dot{S}(0)$. The first derivative of a Bezier curve $Q(u)$ of degree m:

$$Q(u) = \sum_{i=0}^{m} B_{im}(u) \cdot Q_i \text{ with } u \in [0, 1] \tag{8}$$

is given for $u = 0$ by:

$$\frac{dQ(0)}{du} = m \cdot (Q_1 - Q_0) \tag{9}$$

which leads to the definition of the point P_1

$$P_1 = S_0 + k_1 \cdot (S_0 - S_1) \tag{10}$$

with

$$k_1 = \frac{d_{p1}}{d_{s1}} \tag{11}$$

If $k_1 = 1$, i.e. $d_{p1} = d_{s1}$, C^1 continuity is achieved.

G^2 continuity at the leading edge is achieved if there exists a scalar $k_2 > 0$ so that $\ddot{P}(0) = k_2 \cdot \ddot{S}(0)$. The second derivative of a Bezier curve $Q(u)$ of degree m for $u = 0$ is given by

$$\frac{d^2Q(0)}{du^2} = m(m-1) \cdot (Q_0 - 2Q_1 + Q_2) \tag{12}$$

which leads to the definition of the point P_2

$$P_2 = (1 + 2k_1 + 2k_2) \cdot S_0 - (2k_1 + k_2) \cdot S_1 + k_2 \cdot S_2 \tag{13}$$

If $k_1 = 1$ and $k_2 = 1$, C^2 continuity is achieved. Thus, if G^2 or C^2 continuity is prescribed, parameters u_{p2} and d_{p2} are replaced by k_2.

3. Optimization process

The proposed method uses an optimization loop containing three separate blocks (Fig. 3): geometrical modeling, CFD simulation and genetic optimization.

3.1. Geometrical modeling

The surface profile is modeled as described in the previous paragraph, i.e. using two Bezier curves for suction and pressure sides. In the example depicted in Fig. 4, two Bezier curves of degree 4 are used to model the suction and pressure sides with a curvature continuity C^2 at the leading edge. The trailing edge is sharp and modeled as a straight line. This parameterization leads to 11 parameters (Table 4).

The tool path is modeled with the same parameterization than the design approach, that is to say, by means of two Bezier curves of degree 4 with G^n/C^n continuity at the leading edge (Fig. 5, Table 5). The degree of the Bezier curves has been chosen to be consistent with the maximum degree of polynomial curves

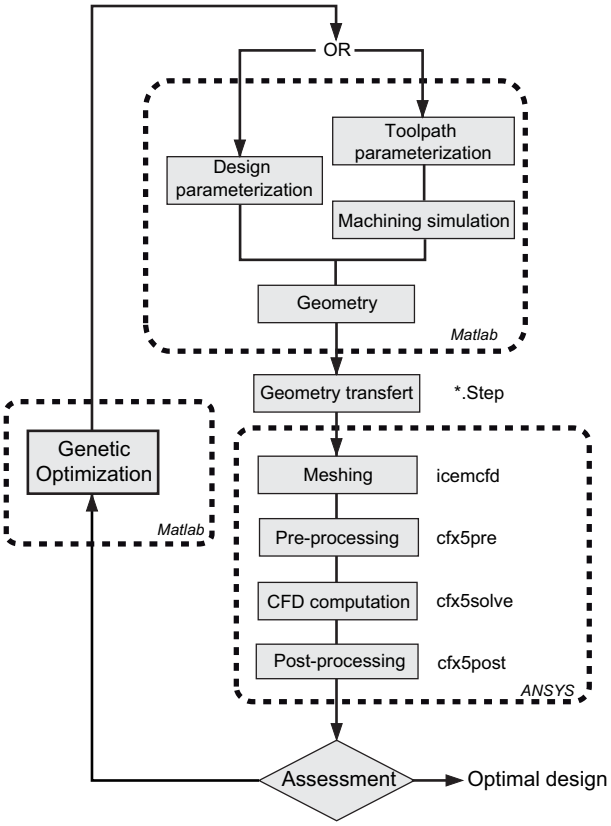

Figure 3: Optimization process

Table 4: C^2 example design parameters

Camber line		**Suction side**		**Pressure side**	
L	0.1 m	u_{s2}	0.3	d_{p3}	0.025 m
α	30°	d_{s1}	0.005 m	d_{p4}	0.002 m
β_1	0°	d_{s2}	0.015 m		
$\beta_{\bar{1}}$	60°	d_{s3}	0.045 m		
		d_{s4}	0.002 m		

a numerical controller could interpolate, which is equal to 5 for a Siemens 840D [27]. In this way, the proposed design process ensures a full polynomial model from design to manufacturing without any geometric approximation.

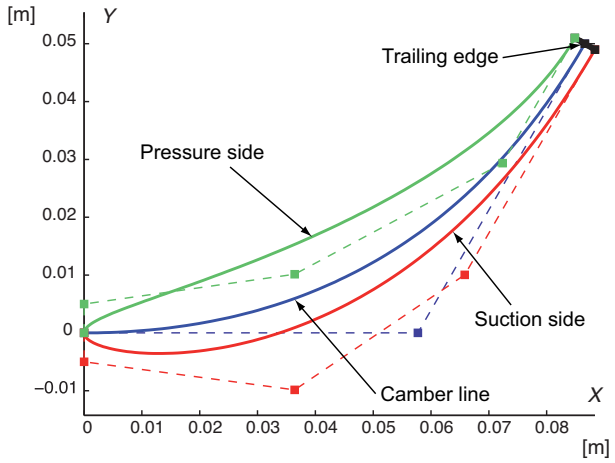

Figure 4: C^2 design example

The first control point of both curves at the leading edge is the offset of the origin point of the camber line CL_0 along its tangent vector with a magnitude equal to the tool radius.

$$S_0 = P_0 = CL_0 + R \cdot \boldsymbol{t} \tag{14}$$

The last control point of both curves at the trailing edge is the offset of the last point of the camber line CL_2 along its normal vector with a magnitude greater or equal to the tool radius.

$$S_4 = CL_2 + (R + d_{s4}) \cdot \boldsymbol{n} \tag{15}$$

$$P_4 = CL_2 - (R + d_{p4}) \cdot \boldsymbol{n} \tag{16}$$

The points describing the blade surface are generated through a kinematical machining simulation. They are defined as points of the envelope surface of the tool movement computed by a N-buffer algorithm [28] (Fig. 6). This consists in computing intersections between the straight lines of the N-buffer with the cylindrical tool. Those points are then used to build the mesh during CFD simulation.

Table 5: G^1 example tool path parameters

Camber line		**Suction side**		**Pressure side**	
L	0.1 m	u_{s2}	0.2	u_{p2}	0.25
α	30°	d_{s1}	0.013 m	d_{p1}	0.012 m
β_1	0°	d_{s2}	0.019 m	d_{p2}	0.014 m
$\beta_{\overline{1}}$	60°	d_{s3}	0.020 m	d_{p3}	0.025 m
		d_{s4}	0.007 m	d_{p4}	0.007 m

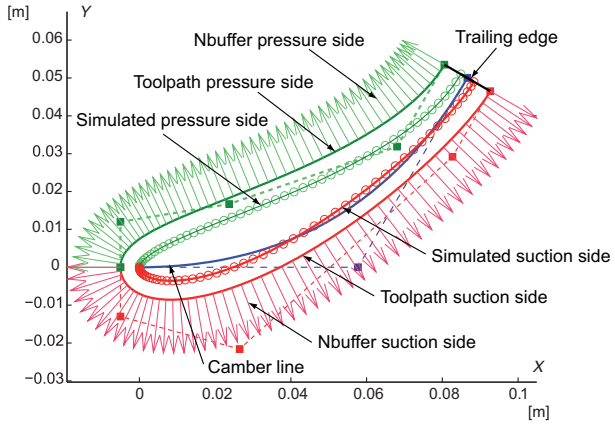

Figure 5: G^1 tool path example

3.2. CFD simulation

The quality of the mesh is an important issue to ensure the quality of the results. In order to obtain accurate results and a robust automatic mesh generation during the optimization process, a hybrid (structured/unstructured) mesh is used as shown in Fig. 7.

The separation between the two mesh types is performed by the convex envelope of a profile's offset. This offset is taken with a magnitude equal to 10 percent of the chord (L) for the suction and pressure sides and 20 percent for the trailing edge. The convex envelope is used to ensure the robustness of the automatic mesh generation in case of a high curvature blade.

The structured mesh is located between the profile and the offset (O-mesh).

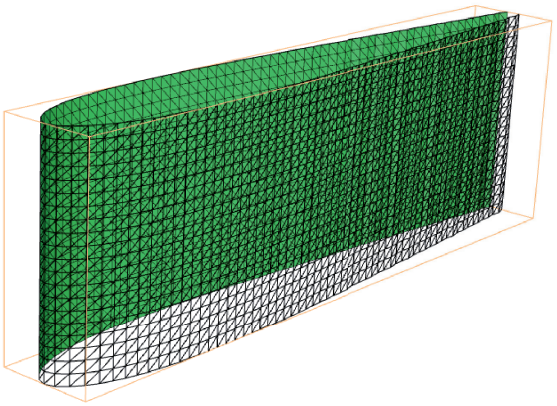

Figure 6: Blade generated with N-buffer simulation

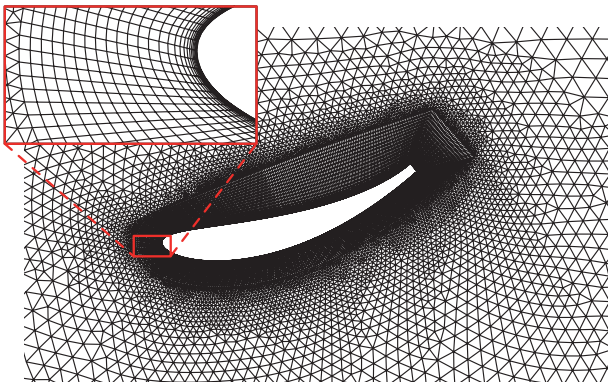

Figure 7: Hybrid mesh

In this way, the boundary layer effects are well captured due to the better accuracy of this kind of mesh. The unstructured mesh is located between the offset and the boundaries of the fluid domain. Therefore, topological constraints do not depend on the relative position of the profile regarding to the boundaries.

The mesh generation is done during the optimization within ANSYS ICEM through a replay script-file written in Tcl/Tk language. The total number of mesh nodes is approximately 100'000 due to the different profiles shapes.

The quality of the mesh is checked for the optimal blades at each iteration.

The quality criterions for a mesh cell are a minimum angle greater than 15 degrees and a relative Jacobian determinant greater than 0.4 [29].

In the present study, the water is assumed as incompressible at constant temperature. The motion of the water is governed by the Reynolds Averaged Navier Stokes (RANS) equations (17) and (18), where p is the static pressure, ν is the kinematic viscosity and $X_i = [X, Y, Z]$ the components of the Cartesian coordinate system. In these equations, the velocity and pressure are split into a mean value and a fluctuating part (19). The Reynolds stresses term defined as $\rho\overline{C_i' C_j'}$ is modelled by the Shear Stress Transport (SST) turbulence model, which is described by Menter et al. [30].

$$\frac{\partial \overline{C}_i}{\partial X_i} = 0 \qquad \text{and} \qquad \frac{\partial C_i'}{\partial X_i} = 0 \tag{17}$$

$$\frac{\partial \overline{C}_i}{\partial t} + \overline{C}_j \frac{\partial \overline{C}_i}{\partial X_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial X_i} + \nu \frac{\partial^2 \overline{C}_i}{\partial X_j^2} - \frac{\partial \overline{C_i' C_j'}}{\partial X_j}. \tag{18}$$

$$C_i = \overline{C_i} + C_i' \qquad \text{and} \qquad p = \overline{p} + p' \tag{19}$$

The dimensions of the computational domain as well as the different boundary conditions are illustrated in Fig. 8.

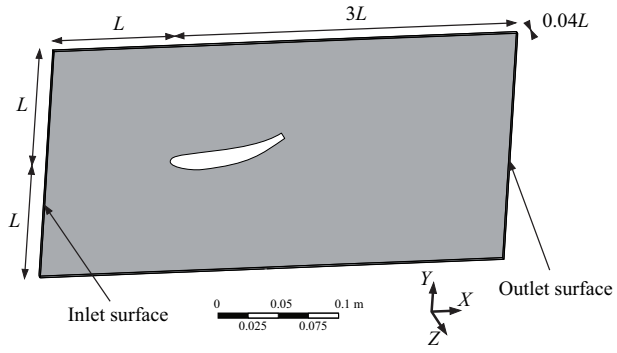

Figure 8: Computational domain

At the inlet surface, a uniform reference velocity C_0 is applied and decomposed as follow: $C_X = 10\ \mathrm{m\,s^{-1}}$, $C_Y = 0\ \mathrm{m\,s^{-1}}$ and $C_Z = 0\ \mathrm{m\,s^{-1}}$. Moreover,

a turbulent intensity of 5% percent is chosen for the inlet, which corresponds to a medium turbulent intensity. On the outlet surface, a zero average static pressure is imposed. The suction side, pressure side and trailing edge surfaces of the blade are set as no-slip smooth walls. Therefore, the velocity on these surfaces is imposed equal to zero. On the top, bottom and sides surfaces a symmetric boundary conditions is applied [31].

In order to improve the convergence of the computation, the initial conditions for the velocity and pressure are taken from the result of a simulation, which corresponds to the middle of the parameters range (Table 6).

The software used for the different numerical flow simulation is the commercial code ANSYS-CFX Release 12.1. The geometry of the computational domain is imported from ANSYS-ICEM. Then, the selected numerical setup for the case study is applied in ANSYS-CFX-Pre. Afterwards, the numerical simulation is performed by ANSYS-CFX-Solver.

This solver uses a finite volume discretization of the equations. Consequently, the continuous value of velocity and pressure of the flow field are discretized on each mesh node in order to solve implicitly a system of algebraic equations. The stop criterion for the computation is a maximum residual of $5 \cdot 10^{-6}$ for each unknown variable or a maximum number of 300 iterations.

The results are analyzed with ANSYS-CFX-Post where the interesting values for the optimization process are extracted. In the present study, the extracted values are: the drag force F_X, the lift force F_Y, the maximum velocity value in the computational domain C_{max} and the non dimensional wall distance y^+.

The maximum velocity value in the computational domain is used to evaluate the specific energy losses in the computational domain. Indeed, as these losses are proportional to $\frac{C^2}{2}$, the minimization of C_{max} corresponds to the minimization of the losses.

The non dimensional wall distance value y^+ represents the ability of the mesh to take into account the physic of the flow (Eq. 20). This value is proportional to the height of the mesh elements in contact with the blade and to the friction

velocity u_τ. After each computation, a maximum y^+ value below 20 is checked on the profile to simulate the boundary layer behavior and verify the mesh quality.

$$y^+ = \frac{u_\tau \cdot y}{\nu} \qquad \text{with} \qquad u_\tau = \sqrt{\frac{\tau_p}{\rho}} \tag{20}$$

3.3. Genetic optimization

The genetic algorithms are based on the principles of natural genetics and evolution. At each step of the optimization process, the algorithm computes the objectives for each individual. Then, the population is sorted and scaled in order to select the individuals, which will be used to produce the next population. This population is either produced by the mutation of a selected individual (18%) or by the crossover of two selected individuals (80%) and with the two best individuals from the previous iteration (2%) [33].

In the present analysis, the multi-objective optimization process is performed with the global optimization toolbox from Matlab. The size of this population is set to 100 different profiles and the maximum number of iterations is limited to 50, which leads to approximately 5'000 computations per optimization run.

The design and tool path parameters are limited by the parameters range given in Table 6. Moreover, linear constraints are applied inside the optimization process in order to produce only feasible profile shape (Eq. 21).

$$\beta_1 \leq \alpha \qquad \text{and} \qquad \alpha \leq \beta_{\bar{1}} \tag{21}$$

In the case of tool path optimization, the curvature radius R_κ of both suction and pressure sides has to be greater than the tool radius R to prevent the generation of a loop in the offset profile (Eq. 22).

$$R_\kappa = \frac{\|\dddot{Q}(u)\|^3}{\|\dot{Q}(u) \times \ddot{Q}(u)\|} \qquad \text{and} \qquad R_\kappa \geq R \tag{22}$$

As the purpose of this paper is to compare the design and tool path approaches in order to generate a blade, only two objectives are arbitrarily chosen for the present study: maximize F_Y and minimize C_{max}.

To decrease the time of the optimization process, the genetic algorithm is used in parallel. Therefore, the simulations of the population are sent to a computational grid to compute several individuals in the same time.

4. Numerical investigations

Numerical investigations have been conducted for eight different cases, gathering the two methods and four types of continuity at the leading edge between suction and pressure sides. The range of the geometrical parameters is given in Table 6.

Table 6: Range of parameters

Parameter	**Min**	**Max**	**Design**	**tool path**
α	$-5°$	$25°$	x	x
β_1	$-20°$	$30°$	x	x
$\beta_{\overline{1}}$	$0°$	$50°$	x	x
u_{s2}	0.01	0.35	x	x
d_{s1}	0.005 m	0.008 m	x	o
d_{s1}	0.008 m	0.011 m	o	x
d_{s2}	0.005 m	0.015 m	x	x
d_{s3}	0.02 m	0.05 m	x	x
d_{s4}	0.002 m	0.002 m	x	x
u_{p2}	0.01	0.35	x	x
d_{p1}	0.005 m	0.008 m	x	o
d_{p1}	0.008 m	0.011 m	o	x
d_{p2}	0.005 m	0.015 m	x	x
d_{p3}	0.02 m	0.05 m	x	x
d_{p4}	0.002 m	0.002 m	x	x
R	0.005 m	0.005 m	o	x

For each case study, ten Intel Xeon CPU at 3.00 GHz with eight cores and 8 GB of memory are used in parallel during the genetic optimization and each numerical simulation is carried out on four cores. The mean computing time for the simulation of an individual is ten minutes, which leads to an averaged

time of one hour per iteration.

The objectives are maximization of the lift F_Y and minimization of the maximal velocity C_{max}. The resulting values for the eight cases are gathered in Table 7. Moreover, lift coefficients C_L and pressure coefficient $C_{p,min}$ have been calculated to non-dimensionalize the results.

$$C_L = \frac{F_Y}{\rho \cdot \frac{C_0^2}{2} \cdot 0.04L^2} \qquad \text{and} \qquad C_{p,min} = 1 - \frac{C_{max}^2}{C_0^2} \tag{23}$$

The magnitude of lift F_Y and maximum velocity C_{max} are approximately the same for both methods and for the different levels of continuities (Table 7). However, in the C^2 case, the maximum lift is lower by about 15% for both design and tool path approaches. This is because the C^2 continuity constraint at the leading edge blocks the first three control points of the pressure side (P_0, P_1, P_2), thereby reducing the degrees of freedom for the generation of the blades.

The lift difference between the two methods is always lower than 4%. This is not significant because it has the same order of magnitude as the amplitude of lift fluctuation (± 5 N). Indeed, the high curvature profiles generate vortices shedding, which induces a fluctuating lift value during the computation despite the use of a steady solver. Moreover, the differences in the optimization results are likely well within the errors introduced by the RANS model for the complex flows over such blades.

The maximal velocity C_{max} (or the pressure coefficient $C_{p,min}$) is always greater for the tool path method because the curvature radius at the leading edge is imposed by the choice of parameters d_{s1} and d_{s2}. Indeed, if these parameters are too small, the radius of curvature of the generated tool path is lower than the tool radius and the N-buffer simulation is not allowed. The choice of the range of these parameters (Table 6) may have been too conservative resulting in more flattened edges, generating slightly higher speed along the profile.

Since there are two different objectives in the optimization, the Pareto fronts

Table 7: Optimisation results

Case	**Max** $\lvert F_Y \rvert$ [N]	**Max** C_L [-]	**Min** C_{max} [$\mathrm{m\,s^{-1}}$]	**Min** $\lvert C_{p,min} \rvert$ [-]
G^1 Design	84.14	4.20	12.15	0.48
G^1 Tool path	88.16	4.41	12.25	0.50
C^1 Design	87.00	4.35	11.66	0.36
C^1 Tool path	84.30	4.21	12.08	0.46
G^2 Design	88.00	4.40	11.73	0.38
G^2 Tool path	84.74	4.23	12.49	0.56
C^2 Design	78.02	3.90	11.73	0.37
C^2 Tool path	76.40	3.82	12.80	0.64

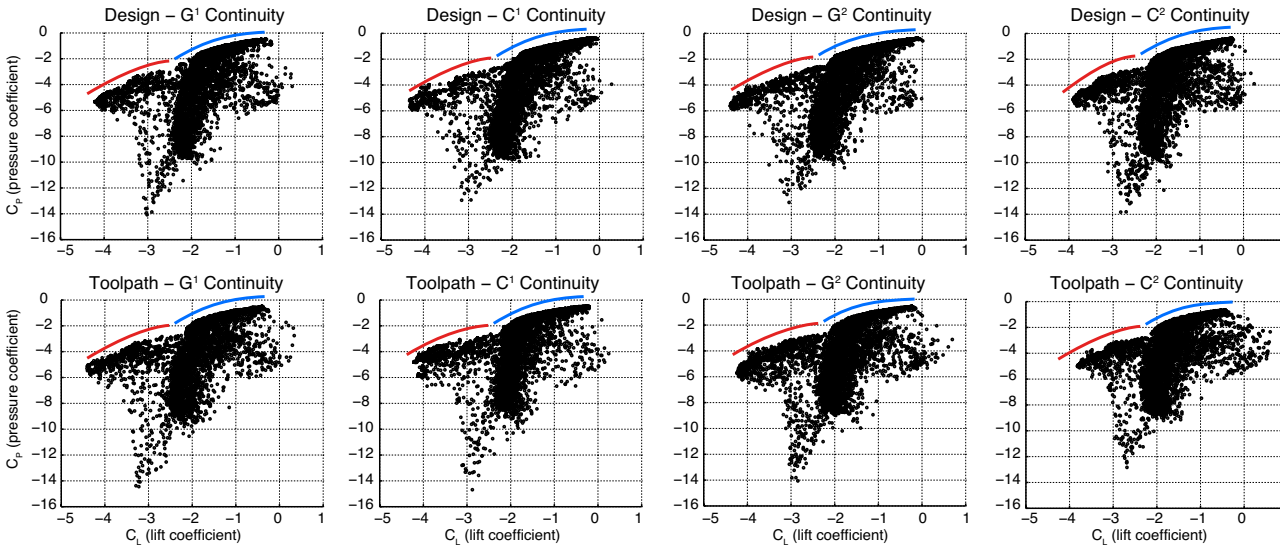

Figure 9: Pareto fronts

for the eight cases have been plotted on Fig. 9. It shows that the Pareto fronts resulting from the optimization of the two criterions C_L and $C_{p,min}$ have the same shape and are divided into two zones.

The Pareto fronts for the G^1 case with design and tool path approaches have been illustrated in Fig. 10 and Fig. 11. Six points on each Pareto front have been chosen and their hydraulic properties have been highlighted. These points are as near as possible for both cases but can't be exactly identical. The geometry of the blades corresponding to the selected hydraulic properties are presented respectiviely in Fig. 12 and Fig. 13. Results show that there are two different families of blade geometry, which depends strongly on the angles that define the camber line (α, β_1 and $\beta_{\overline{1}}$). The geometries created by the conventional approach and the proposed one are similar. Slender blades generate slow velocity profiles whereas curved blades generate high lift values which is coherent with the theory. Both approaches lead to this behavior, which proves that the proposed approach is consistent.

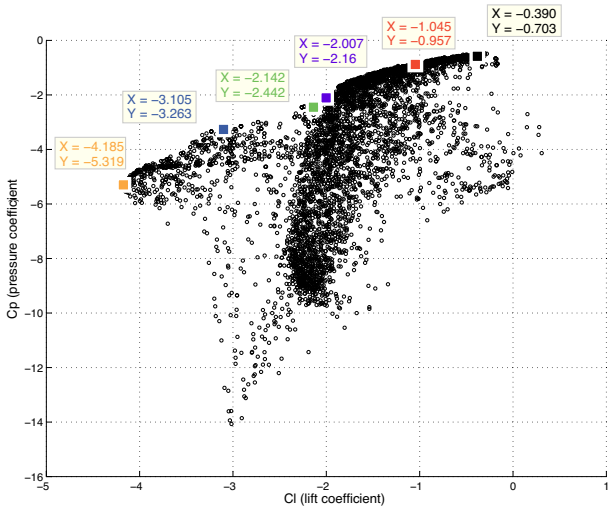

Figure 10: Pareto front ; design G^1 continuity

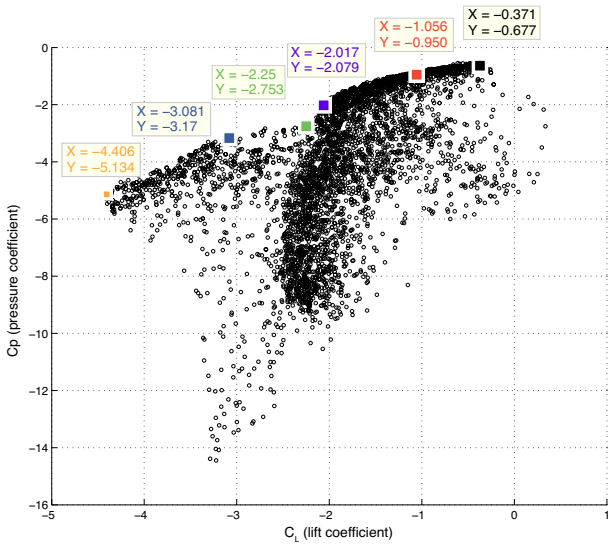

Figure 11: Pareto front ; tool path G^1 continuity

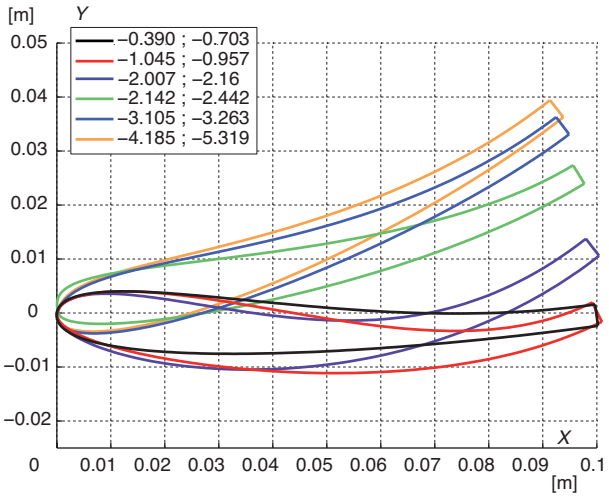

Figure 12: Geometries on Pareto front ; design G^1 continuity

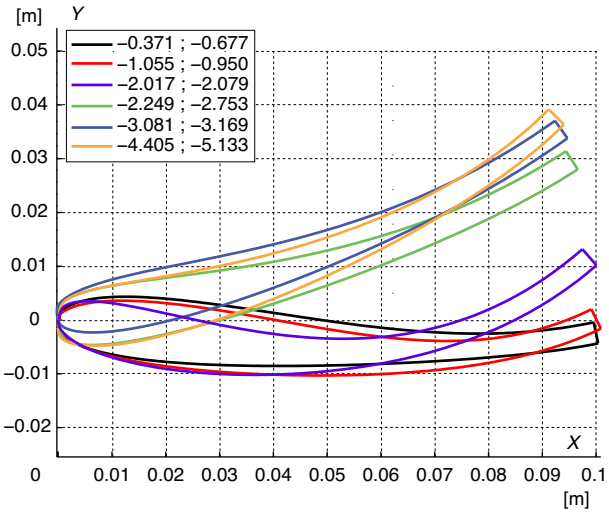

Figure 13: Geometries on Pareto front ; tool path G^1 continuity

5. Conclusions

In this paper, a different approach to design hydrodynamic profiles has been developed. This approach is based on the definition of a tool path such that the envelope of the movement of the tool optimizes one or more hydrodynamic criteria. A unified parametric model to design the blades for both approaches (design and tool path) was proposed as well as a robust automatic meshing strategy to mesh the fluid domain based on different blade geometries.

Overall, the results show that the proposed approach can generate geometries whose performances are comparable to the ones obtained by the classical approach. The advantage is that these performances are not degraded because the same polynomial curves would be used from the design stage to the machining stage without modifications in the post-processor or in the NC unit. Polynomial format ensures the smoothness of the trajectory, which is one of the parameters required to provide a good surface finish. To enhance the optimization, machining phenomena such as dynamical behavior of the machine tool could be introduced in the machining simulation.

The method was validated on a 2D example. The next step is the design and manufacture of impellers, usually designed with non-developable ruled surfaces, which are impossible to machine in 5-axis flank milling without geometrical deviations. However, the large number of parameters to control the geometry of such complex blades suggests to use faster optimization technics than genetic algorithms.

Nomenclature

Geometric parameters

m	Degree of the curve [−]
u	Curvilinear abscissa [−]
$B_{im}(u)$	Bernstein polynomial i [−]
$Q(u)$	Bezier curve [−]
Q_i	Control point i [−]
β_1, $\beta_{\overline{1}}$	Inlet, outlet angle [°]
L	Chord length [m]
α	Chord angle [°]
$C_L(u)$	Camber line function [m]
CL_i	Camber line control point i [m]
$\boldsymbol{n}$	Vector normal to camber line [m]
$\boldsymbol{t}$	Vector tangent to camber line [m]
$S(t)$, $P(t)$	Suction, pressure side function [m]
S_i, P_i	Control point i [m]
u_{si}, u_{pi}	Control points abscissa i [−]
d_{si}, d_{pi}	Control points distance i [m]
G^n, C^n	Continuity degree n [−]
k_n	G^n/C^n continuity parameter [−]
R	Tool radius [m]
R_κ	Curvature radius [m]
X_d^*, X_t^*	Optimal design, optimal tool path [−]

CFD parameters

X, Y, Z	Cartesian component [m]
X_i	Cartesian component i [m]
$\boldsymbol{C}$	Absolut velocity [$\mathrm{m\,s^{-1}}$]
C_i	Absolut velocity component i [$\mathrm{m\,s^{-1}}$]
C_0	Reference velocity at inlet [$\mathrm{m\,s^{-1}}$]
p	Static pressure [Pa]
ρ	Density [$\mathrm{kg\,m^{-3}}$]
ν	Kinematic viscosity [$\mathrm{m^2\,s^{-1}}$]
F_X, F_Y	drag, lift force [N]
y^+	Wall distance [$-$]
τ_p	Wall shear stress [Pa]
C_L	Lift coefficient [$-$]
C_p	Pressure coefficient [$-$]

Abbreviations

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CFD	Computer Fluids Dynamics
FEA	Finite Element Analysis
GSO	Global Shape Optimization
LE	Leading Edge
NS	Navier Stokes
NC	Numerical Controllers
RANS	Reynolds Averaged Navier Stokes
SID	Shape Inverse Design
SST	Shear Stress Transport
TE	Trailing Edge

References

[1] B.K. Choi , R.B. Jerard. Sculptured surface machining, theory and applications. Dordrecht: Kluwer Academic Publishers, 1998.

[2] C. Lartigue, F. Thiebaut, T. Maekawa, CNC tool path in terms of B-spline curves, Computer-Aided Design 2001;33:307-319.

[3] Cheng, M.-Y., Tsai, M.-C., Kuo, J.-C., Real-time Nurbs command generators for CNC servo-controllers, International Journal of Machine Tools and Manufacture 2002;42(7):801-813.

[4] E. Duc, C. Lartigue, C. Tournier and P. Bourdet, A new concept for the design and the manufacturing of free-form surfaces: the machining surface. CIRP Annals - Manufacturing Technology 1999;48(1):103-106.

[5] H. Tonshoff, N. Rackow, Optimal tool positioning for five-axis flank milling of arbitrary shaped surfaces, Annals of the German Academic Society for Production Engineering (WGP) 2000;7(1): 57-60.

[6] C. Menzel, S. Bedi, S. Mann, Triple tangent flank milling of ruled surfaces, Computer-Aided Design 2004;36(3):389-296.

[7] J. Senatore, F. Monies, J.-M. Redonnet, W. Rubio, Analysis of improved positioning in five-axis ruled surface milling using envelope surface, Computer-Aided Design 2005;37(10):989-998.

[8] P.-Y. Pechard, C. Tournier, C. Lartigue, and J.-P. Lugarini. Geometrical deviations versus smoothness in 5-axis flank milling. International Journal of Machine Tools and Manufacture 2009;49(6):454-461.

[9] C. Lartigue, E. Duc, A. Affouard, Tool path deformation in 5-axis milling using envelope surface, Computer-Aided Design 2003;35(4):375-382.

[10] L. Zhu, G. Zheng, H. Ding, and Y. Xiong. Global optimization of tool path for five-axis flank milling with a conical cutter. Computer-Aided Design 2010;42(10):903-910.

[11] J. Chaves-Jacob, G. Poulachon, and E. Duc. New approach to 5-axis flank milling of free-form surfaces: Computation of adapted tool shape. Computer-Aided Design 2009;41(12):918-929.

[12] H. Gong, F. Z. Fang, X. T. Hu, L.-X. Cao, and J. Liu. Optimization of tool positions locally based on the bceltp for 5-axis machining of free-form surfaces. Computer-Aided Design 2010;42(6):558-570.

[13] T. Korakianitis and G. Pantazopoulos. Improved turbine-blade design techniques using 4th-order parametric-spline segments. Computer-Aided Design 1993;25(5):289-299.

[14] B. Dennis, G. Dulikravich, and Z.-X. Han. Constrained shape optimization of airfoil cascades using a navier-stokes solver and a genetic/sqp algorithm. In ASME, International Gas Turbine and Aeroengine. Congress and Exhibition, Indianapolis, USA, 1999.

[15] W. S. Ghaly and T. T. Mengistu. Optimal geometric representation of turbomachinery cascades using nurbs. Inverse Problems in Science and Engineering 2003;11(5):359-373.

[16] L. Ferrando, J.-L. Kueny, F. Avellan, C.Pedretti, and L. Thomas. Surface parameterization of a francis runner turbine for optimum design. In 22nd IAHR Symposium on Hydraulic Machinery and Systems, 2004.

[17] L. Ferro, L. Gato, and A. Falcao. Design and experimental validation of the inlet guide vane system of a mini hydraulic bulb-turbine. Renewable Energy 2010;35(9):1920-1928.

[18] I. Abbott and A. Doenhoff. Theory of wing sections: including a summary of airfoil data. Dover books on physics and chemistry. Dover Publications, 1959.

[19] D. Bonaiuti, A. Arnone, M. Ermini, and L. Baldassarre. Analysis and optimization of transonic centrifugal compressor impellers using the design of experiments technique. Journal of Turbomachinery 2006;128(4):786-797.

[20] G. N. Koini, S. S. Sarakinos, and I. K. Nikolos. A software tool for parametric design of turbomachinery blades. Advances in Engineering Software 2009;40(1):41–51, 1.

[21] L. Pritchard. An eleven parameter axial turbine airfoil geometry model. In ASME 1985;85-GT-219.

[22] S. Pierret and R. A. Van den Braembussche. Turbomachinery blade design using a navier-stokes solver and artificial neural network. Journal of Turbomachinery 1999;121(2):326–332.

[23] J. M. Anders, J. Haarmeyer, and H. Heukenkamp. A parametric blade design system (part i + ii) Von Karman Institut lecture serie1, 2008.

[24] D. Buche, G. Guidati, and P. Stoll. Automated design optimization of compressor blades for stationary, large-scale turbomachinery. ASME Conference Proceedings 2003;(36894):1249-1257.

[25] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Progress in Aerospace Sciences 2002;38(1):43-76.

[26] S. Goel. Turbine airfoil optimization using quasi-3d analysis codes. International Journal of Aerospace Engineering, 2009.

[27] Siemens. www.automation.siemens.com/doconweb, Sinumerik 840D/840Di/810D user, Programming Manual Job planning, 2010.

[28] R. Jerard, R. Drysdale, K. Hauck, B. Schaudt, and J. Magewick. Methods for detecting errors in numerically controlled machining of sculptured surfaces. IEEE Computer Graphics and Applications 1989;9(1):26–39.

[29] ANSYS ICEM CFD 12.1. Help Manual, 2009.

[30] F. R. Menter, M. Kuntz, R. Langtry. Ten years of industrial experience with the SST turbulence model. Begell House Inc, 2003.

[31] ANSYS CFX-Solver 12.1. Theory Guide, 2009.

[32] I. L. Ryhming. Dynamique des fluides. 2nd Edition, Presses Polytechniques et Universitaires Romandes, 2004

[33] U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay. Multiobjective genetic algorithms for clustering. Springer, 2011