OPTIMAL DESIGN OF AXIAL HYDRAULIC TURBINE USING GLOBAL OPTIMIZATION TECHNIQUES BASED ON METAMODEL CONSTRUCTION AND COMPUTATIONAL FLUID DYNAMICS (CFD).

Edna R. Da Silva, ednaunifei@yahoo.com.br
Nelson Manzanares Filho, nelson@unifei.edu.br
Ramiro G. Ramirez Camacho, rgramirez65@hotmail.com
IEM-UNIFEI, Federal University of Itajubá, Itajubá, Brazil

Abstract. *This work presents a methodology for performance optimization of axial hydraulic turbine runners. The blade shape has great influence on the hydraulic performance of the turbine, as well as the stagger angles, blade spacing, hub-to-tip ratio and other geometric parameters. The objective is the maximization of the hydraulic efficiency starting from a preliminary design. A global optimization algorithm known as Controlled Random Search Algorithm (CRSA) is applied with guidance of a strategy for accelerating the optimization process when costly functions are involved (as is the case). This strategy is called CORS (Constrained Optimization using Response Surfaces) and is based on iterative constructions of metamodels to be used in place of the actual model for the CRSA intervention. Relatively few design points are exactly evaluated and recorded with their function values in a database for further metamodel construction. Radial Basis Functions like multiquadrics are used for this construction (CORS-RBF) and heuristic criteria are adopted to select candidate points for exact evaluation. In order to balance global and local searches, the candidate points are constrained to rest at safe distances from the database points in a cyclical manner: large distances help to improve the metamodel construction in regions not yet explored while small distances promote local searches around promising points according to the metamodel optimization. The methodology implementation consists in automatic building of parameterized blade geometries and meshes via "script files" with editing commands written in Tcl/Tk language, which will be interpreted by the commercial software ICEM-CFD, in batch mode. For the numerical calculation of the flow in the turbine runner domain, the software FLUENT is used with fluid properties, turbulence model and boundary conditions set through "journal files". Besides the presentation of an improved runner design methodology, this work aims to evaluate the computational gain allowed by the CORS-RBF strategy on a high performance computational environment. Three-dimensional meshes with the number of nodes counted in millions are commonly employed in such environments, and the time spent for a unique evaluation may vary from few minutes to several hours or days. Since dozens or hundreds of evaluations are normal in optimization tasks, an acceleration strategy like CORS-RBF is certainly welcome. An example presented in this paper shows that acceleration rates from 3 to 5 times are attainable.*

Keywords*: Metamodels; Hydraulic Turbine; Optimization; CFD; CRSA; CORS-FBR*

1. INTRODUCTION

CFD techniques have been developed over the past decades as a powerful analysis tool for quantification of flow fields in complex geometries, especially those found in turbomachinery design. Such techniques have been used by the aeronautical industry since the 60ths, begging with the classical panel method with boundary layer interaction to account viscous effects. Nowadays, the use of computational fluid dynamics (CFD) for solving the full Navier-Stokes equations has become a common issue in several industrial design activities. Nevertheless, such computations may represent a bottleneck when a great number of concurrent geometrical and flow parameters must be analyzed during the searching of good solutions for satisfying certain design objectives. Normally, such task is better accomplished by means of a suitable optimization algorithm (OA). But taking into account real life constraints, such as the available computational environment and budget, the number of comparative evaluations required by an OA may become prohibitive in a specific design situation.

To overcome this drawback, several strategies have been conceived for accelerating the optimization task such as: (i) use of multiprocessing; (ii) use of better optimization algorithms; (iii) use of metamodels (surrogate models) for reducing the number of calls to the true solver model. From a strict engineering point of view, the 3rd strategy seems to be more inexpensive and universal, since it does not rely on costly hardware improvements neither on technical advances in optimization algorithms.

Metamodels (surrogate or response surface models) have been extensively used for representing expensive black box functions. A recent published monograph on the subject is already available (Forrester et al., 2008) focusing practical aspects. Praveen and Duvigneau (2007) classified surrogate models into four main categories: (i) data-fitting models, where an approximation of the expensive function is constructed using an available data bank; (ii) variable convergence models, where the expensive function depends of the numerical solution of a partial differential equation with a relaxed stopping criterion; (iii) variable resolution models, where a hierarchy of grids is used and the surrogate model is just the costly evaluation tool but run on a coarse grid; (iv) variable fidelity models, where an hierarchy of physical models is used. The first category is focused in this paper.

Often the constructed metamodel itself is of prime importance for the user when it is meant for utilization without further callings to the costly model. Sometimes, however, surrogate models are employed just for accelerating global optimization algorithms which otherwise would require a prohibitive number of costly function evaluations. In this context, provisory metamodels are iteratively constructed from points already evaluated by the costly model (Jones, 2001a). Here it is not so important to construct a representative metamodel over the whole search region but just in the vicinity of promising points. But the conscientious specification of newly points for exact evaluation is very important because they also must provide a selective metamodel improvement during the iterative process.

Regis and Shoemaker (2005) developed a Constrained Optimization using Response Surfaces (CORS) method. Cyclic search patterns for metamodel constrained optimizations are iteratively used in order to specify newly evaluated points. These patterns induce distance constraints between the points already evaluated and the metamodel optima for balancing local and global searches and also for improving the metamodel construction. The method was shown to converge to the global minimizer of any continuous function on a compact set regardless of the response surface model and the basic optimization algorithm that are used. Regis and Shoemaker (2005) used radial basis functions (CORS-RBF) for the metamodel construction and the DIRECT method of Jones (2001b) for global optimization.

This paper presents an implementation of the CORS-RBF method using inverse multiquadrics and multiquadrics for metamodel construction and the Controlled Random Search Algorithm (CRSA). CRSA is a stochastic, population-set based algorithm, capable of performing global optimization tasks efficiently. CRSA was first proposed by Price (1977) and later improved by Ali et al. (1997, 2004). Further improvements were introduced by Manzanares-Filho et al. (2005).

The paper is outlined as follows. Section 2 describes briefly the basics of the CORS-RBF strategy. Some implementation issues are discussed in Section 3. Section 4 discusses optimization tool. Section 5 presents and discusses the results obtained with the application of the proposed CORS-RBF implementation to some Dixon-Szegö test functions and results for a cascade design. Test results for axial hydraulic turbine design are presented and discussed in Section 6. Concluding remarks are considered in Section 7.

2. GLOBAL OPTIMIZATION USING METAMODELS

The CORS^{PP} strategy is meant for finding a global minimum of a box-constrained continuous function f: $D \rightarrow \Re$, where $D = \{\boldsymbol{x} \in \Re^d$: $x_j^L \le x_j \le x_j^U$, $j = 1, ..., d\}$; x_j^L and x_j^U are lower and upper bounds for the d coordinates of $\boldsymbol{x}$ respectively. A point $\boldsymbol{x}^*$ is said to be a global minimum of f if $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in D$. The focus is on problems where f is a black box function that is expensive to evaluate. Thus it is important to find a point $\boldsymbol{x}' \in D$ such that $f(\boldsymbol{x}')$ is close to $f(\boldsymbol{x}^*)$ using only a relatively small number of function evaluations.

The strategy is iterative. In each iteration, a response surface metamodel is constructed and one point is selected for costly function evaluation. This is the point that minimizes the current response surface model subject to the box constraints and to some constraints on the distance from previously evaluated points. The selection of points for costly function evaluation has two goals: (a) finding new points with lower objective function values than those already evaluated, and (b) improving the response surface model by sampling regions of D not well explored yet. Hence, the next point for costly function evaluation is the one that minimizes the current response surface model subject to constraints on how close the next point evaluated can be to previously evaluated points. There is a natural limit on how far a point can be from a previously evaluated point. If $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are the previously evaluated points, then this limit is given by

$$\Delta = \max_{\boldsymbol{x}' \in D} \min_{1 \le j \le n} \left\| \boldsymbol{x}' - \boldsymbol{x}_j \right\| \tag{1}$$

A point satisfying (1) is called a *maxmin point*. A scheme for calculating Δ is assumed. The next evaluation point is required to be at a distance no less than $\beta\Delta$ from all previously evaluated points, where $0 \le \beta \le 1$. The CORS general algorithm can be found in reference (Regis and Shoemaker, 2005; Silva and Manzanares-Filho, 2010).

A search pattern $\langle \beta_1, \beta_2, \dots, \beta_N \rangle$ is chosen with $1 \ge \beta_1 \ge \beta_2 \ge \dots \ge \beta_N$ and applied in cycles of N iterations such that $\beta_i = \beta_{i+N}$. Regis and Schoemaker (2005) showed that for CORS strategy converges to a global minimum in D it is sufficient that the constraints be satisfied and the search pattern have at least one nonzero entry. The metamodel construction method and the optimization tools employed are not fundamental for convergence. However, all these implementation aspects may be important for the convergence fastness of CORS-RBF strategy.

3. METAMODEL CONSTRUCTION

The metamodel construction is made with inverse multiquadric radial basis functions for Dixon-Szegö test functions (Dixon and Szegö, 1978) and multiquadric for blade cascade and axial hydraulic turbine design example. Assume that we have n distinct points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \Re^d$ where the function values $f(\boldsymbol{x}_i)$ are known. The function interpolant is of the form

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \, \phi \|\boldsymbol{x} - \boldsymbol{x}_i\|, \qquad \boldsymbol{x} \in \Re^d \tag{2}$$

where $\| \ \|$ stands for the Euclidean norm in $\Re^d$ and λ_i are real coefficients to be determined. The multiquadric radial basis function is given by

$$\phi(r) = \sqrt{r^2 + c^2} \tag{3}$$

and the inverse multiquadric radial basis function is given by

$$\phi(r) = \frac{1}{\sqrt{r^2 + c^2}} \tag{4}$$

In (3) and (4), c is called *shape parameter* which has a strong influence on the interpolation. Increasing the c value makes the interpolation more predictive at expenses of ill-conditioning. Use of optimized values for c may improve the interpolation and thus the CORS-RBF performance as compared with radial functions without a shape parameter (thin plate splines, for example).

The interpolant in (2) is applied to the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and an algebraic linear system results for the λ_i coefficients:

$$\sum_{i=1}^{n} \lambda_i \, \phi \|\boldsymbol{x}_j - \boldsymbol{x}_i\| = f(\boldsymbol{x}_j) \tag{5}$$

The coefficient matrix in (5) is symmetric and positive definite for the inverse multiquadric function given in (5). Thus the matrix is invertible and the system (5) is solvable. For the multiquadric function in (3) this solvability is not always assured, but cases of singularities are quite rare (Hon and Shaback, 2001). On the other hand multiquadrics usually performs better than inverse multiquadrics for the same value of c.

4. OPTIMIZATION TOOLS

For the solution of the optimization problem of CORS-RBF strategy we propose here the use of a Controlled Random Search Algorithm (CRSA). CRSA is an iterative, stochastic, population-set based algorithm, which promotes the substitution of the worst point of the population by a better one at each iteration (Price, 1977). The CRSA version used in this paper adopts improvements introduced by Ali et al. (1997, 2004) and Manzanares-Filho et al. (2005). The extrema of coordinate-wise parabolic interpolations are used for selecting candidate points for entering the population. A penalty function scheme is used for treating the constraints.

Like in other stochastic algorithms, the convergence of CRSA to a global optimum is not assured in general. Thus it is convenient to apply CRSA several times in order to increase the chance of obtaining a solution closer to a global optimum. When this is done, the better found solution is adopted as the candidate point.

5. COMPUTATIONAL TESTS

5.1 Dixon_Szegö Functions Example

The proposed CORS-RBF implementation was tested on some Dixon-Szegö test functions (Dixon and Szegö, 1978). Table 1 shows the characteristics of these functions. Their actual functional expressions can be found in that reference.

Table 1. The Dixon-Szegö test functions.

Test function	Dimension d*	Domain	Nº of local minima	Nº of global minima	Global min value
Branin	2	[-5,10] x [0,15]	3	3	0.398
Goldstein-Price	2	$[-2,2]^2$	4	1	3
Hartman3	3	$[0,1]^3$	4	1	-3.86
Shekel5	4	$[0,10]^4$	5	1	-10.1532
Shekel7	4	$[0,10]^4$	7	1	-10.4029
Shekel10	4	$[0,10]^4$	10	1	-10.5364
Hartman6	6	$[0,1]^6$	4	1	-3.32

* number of full factorial points for CORS is equal to 2^d

Due to stochastic nature of CRSA, each case was tested by using 20 independent runs. The initial CORS data bank was set by a two-level full factorial sampling (2^d points placed at the domain corners). For larger dimensions than those of the tests, other initial samplings may be necessary. The search pattern $\langle 0.95, 0.50, 0.25, 0.005, 0.0005, 0.00\rangle$, ($N = 6$), was chosen for all tests. The CRSA was applied 10 times for solving the auxiliary problem in Step 3.2. A population equal to 10(d+1) is adopted; following a suggestion of Ali et al. (1997) (d is the problem dimension). Within the CORS-RBF framework, the CRSA stopping criterion was a population contraction tolerance below 10^{-4} (absolute difference between the function values of the worst and best points in the population) or the maximum number of functions evaluations equal to 10000. The criterion of convergence of CORS-RBF was to get a relative error between calculated and known minimum values less than 1%. Besides the CORS-RBF strategy evaluation, tests were also made using the CRSA directly for comparison. In this case, the same criterion of convergence of CORS-RBF was used and 20 independent runs were also performed.

A run is considered unsuccessful when the 1% criterion is not satisfied after 10000 function evaluations. The ratio between the successful runs and the total runs is called the ratio of success, RS.

During the tests it was observed that the shape parameter c in the inverse multiquadrics exerted a strong effect on the convergence fastness. The Shekel functions require values of c substantially smaller than the other functions. It is certainly advisable to implement a consistent scheme for optimizing the shape parameter in order to decrease this effect (Praveen and Duvigneau, 2007). Preliminary studies for implementing such a scheme in the CORS-RBF context were already made but they are not sufficiently conclusive yet (Silva et al., 2010). For some functions (Goldstein-Price, for example), this preliminary scheme does not work at all. It is expected that the situation will change in the near future. For this reason, results for just two values of c chosen by trial-and-error for each function will be presented in this work.

Table 2. Comparison of CORS implementations and direct CRSA on the Dixon-Szegö test functions

Test function	c	CORS-RBF (average)	CORS-RBF (min)	CORS-RBF (max)	ORIGINAL CORS [*]	CRSA (average)	CRSA RS
Branin	0.05 / 0.5	21.7 / 19.8	20 / 16	32 / 20	34 / 40	138	100 %
Goldstein-Price	0.05 / 0.5	88.9 / 46.8	51 / 35	142 / 65	49 / 64	257	100 %
Hartman3	0.05 / 0.5	71.9 / 36.8	17 / 22	131 / 48	25 / 61	113	100 %
Shekel5	0.05 / 0.1	57.0 / 66.4	44 / 22	74 / 147	41 / 52	621	55 %
Shekel7	0.05 / 0.1	55.0 / 61.4	39 / 39	104 / 80	46 / 64	588	55 %
Shekel10	0.05 / 0.1	46.3 / 62.1	33 / 33	76 / 112	51 / 64	553	80%
Hartman6	0.05 / 0.5	215.8 / 105.0	153 / 93	249 / 117	104 / 108	427	80 %

The values in the table indicate the number of function evaluations to get a relative error less than 1%
*Results obtained by Regis and Shoemaker (2005) with two different search patterns

Table 2 shows the obtained results including in the 2nd column the values of c used for each function. CORS-RBF denotes the implementation of the present paper. The average, minimum and maximum number of functions evaluations along the 20 independent runs are listed in the 3rd, 4th and 5th columns, respectively (according to the corresponding value of c). The best and worst results obtained by Regis and Shoemaker, (2005) using two different search patterns are also shown in the 6th column. The last two columns list the average values and the rate of success RS obtained with the direct application of CRSA.

Two aspects are salient from the results presented in Table 2: (*i*) the strong influence of the c value on the convergence speed of CORS-RBF; (*ii*) the relatively high difference between the maximum and minimum numbers of function evaluations for convergence. This second aspect is associated to CRSA randomness. Both aspects are interrelated: except for the Shekel7 function, the better average values are accompanied by smaller relative differences ((max – min)/average). A plausible explanation for this trend is that optimum values of c might exist for improving the metamodel construction; moreover the use of such values may not only accelerate the CORS-RBF convergence but also decrease the randomness effects of CRSA. An evident conclusion is that a consistent scheme for optimizing the *c* values for CORS-RBF metamodel construction will not only be welcome but rather necessary when using CRSA.

Using the best average values of CORS-RBF and CRSA, we see that the metamodeling strategy is able to reduce the number of function evaluations substantially. The minimum acceleration occurred for the Hartman3 function (~3 times) while the maximum acceleration occurred for the Shekel10 function (~12 times). Further, it is important to report that a rate of success of the CORS-RBF was 100% for all the tests what did not always occur when CRSA was applied directly. Thus, the CORS-RBF was able not only to accelerate the optimization of costly functions but also to improve the reliability and robustness of the search process proper of CRSA.

From the results of Table 2, it is also possible to conclude that the implementation of CORS-RBF presented in this paper may be in general competitive with the original implementation of Regis and Shoemaker (2005).

5.2 Blade Cascade Optimization Example

Blade cascade analysis still represents a fundamental tool in turbomachinery design context. Relying on 2-D flow models, cascade flow computations are much faster than 3-D models of similar physical complexity. For testing purposes, the CORS-RBF/CRSA methodology is applied now to a simple case of blade cascade design.

A periodical series of identical profile shapes (airfoils) deflects an upstream uniform flow (velocity W_1, angle β_1) to a downstream flow (velocity W_2, angle β_2). The geometrical cascade parameters are the airfoil shape, the pitch-to-chord ratio, $\lambda = t/l$, and the stagger angleβ. The Reynolds (Re) and Mach number are flow parameters .Here only the Reynolds number is relevant since the flow is considered as incompressible (Mach = 0).

Relevant resulting quantities include the pressure distribution on the blade surface, the flow deflection angle $\theta = \beta_1 - \beta_2$, energy losses, lift and drag blade forces. The flow computations were made using the CFD software FLUENT. The required meshes were generated by the software ICEM-CFD. Some comparisons between results obtained by FLUENT, a viscous-inviscid flow injection (VIFI) panel method (Ramirez and Manzanares-Filho, 2005), potential flow, and experimental results (Emery et al., 1958) are shown in Figure 2. One can see that both FLUENT and VIFI produces acceptable results, VIFI being significantly faster. In the future, both may be used together within a multi-fidelity computational scheme, but in this work only FLUENT was adopted for CORS-RBF/CRSA evaluation.

The 2-D meshes are generated by a script written in Tcl/Tk language that can be modified by the optimizer and interpreted by software ICEM-CFD. Care was taken in the refinement of the mesh near the wall in order to properly quantify the friction stresses. All meshes were constructed of 122,688 hexahedrical elements, irrespectively of the cascade parameters. Figure 1 shows a mesh sample around a blade airfoil. The mesh elements grow geometrically from the walls to outside with a ratio of 1.1. The non-dimensional wall distance Y^+ of the mesh points nearest to the walls is set equal to one.

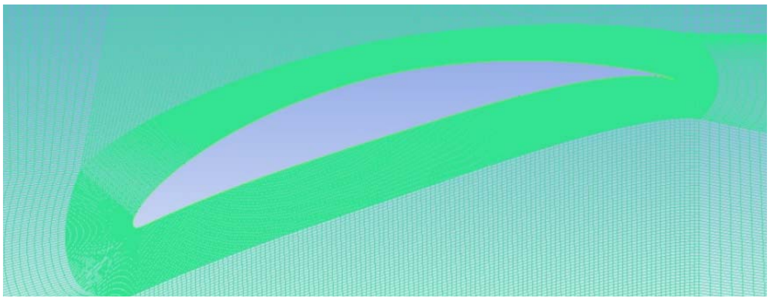


Figure 1. Mesh generated for the initial Profile NACA65(15)-10

The flow velocity vector is set at the cascade inlet and the pressure at outlet. Periodic boundary conditions were considered for reducing the computational domain to a unique periodic region around an airfoil. The turbulence model Spalart-Allmaras (SA) with wall functions was chosen since this enables realistic responses for aerodynamic problems with relatively low Reynolds numbers and boundary layers subjected to adverse pressure gradients. A flow computation performed by FLUENT was considered converged when the normalized residuals of the conservation and turbulence model equations became below 10^{-7}.

The drag and lift coefficients were calculated with basis on the magnitude of the mean velocity vector, W_∞. Lift coefficient was computed by direct integration of pressure and friction stress distributions on the airfoil surface. Drag coefficient was computed differently: first, the difference of the mass averaged total pressure between cascade inlet and outlet is evaluated and the following loss coefficient ζ is computed:

$$\zeta = \frac{P_1 - P_2}{(\rho/2)W_2^2} \tag{6}$$

The outlet mass averaged quantities were evaluated by control line (line/rake) located at a distance of a chord length from the trailing edge. Hence, the drag coefficient is computed by the following relation (Vavra, 1974):

$$C_d = \frac{\zeta \cos^3 \beta_\infty}{(l/t)\cos^2 \beta_2} \tag{7}$$

This methodology for calculating the drag coefficient avoids numerical errors associated with the integration of the blade surface forces. These errors may be intensified by the mesh variations required by the optimization process (even small) by producing numerical noise.

The NACA65(XX)-10 family was selected for the test. Only two design variables were considered: the camber parameter ARC = XX/10 and the pitch-to-chord ratio $\lambda = t/l$. The cascade baseline had NACA65(15)-10 airfoils (ARC = 1.5) and $\lambda = 1$. Table 3 shows the parameters and design ranges used in the test. A Reynolds number equal to 2.45 x10^5 was considered along the entire test.

Table 3. Geometry Data, Boundary Condition and Fluid Properties.

Cascade Parameters	
Pitch-to-chord ratio (λ)	0.85 – 1.15
Camber (ARC)	14% - 16%
Stagger angle (β)	15°
Chord (l)	1 m
Inlet angle flow (β_1)	30°
Incidence angle (α_1)	$\alpha_1 = \beta_1 - \beta$
Flow properties (air)	
Mass density (ρ)	1.225 kg/m^3
Viscosity (μ)	1.7894x10^{-5} kg/(m.s)
Inlet Boundary Condition	
Component velocity in x	3.1783 m/s
Component velocity in y	1.8350 m/s

The maximization of the lift-to-drag ratio (C_l/C_d) was chosen as objective without constraints on the outlet flow angle. In principle, this is very close to the minimization of the loss coefficient ζ with a constrained outlet angle equal to that of the baseline.

First the CRSA was tested without CORS-RBF. As in the computational tests of Section 5.1, an initial a population of 15 individuals was randomly set and evaluated. The stopping criterion was the population contraction tolerance below 10^{-4}. Only one run was performed and converged results were obtained after 47 exact function calls.

Two-level full factorial sampling (4 points at the corners of the design domain, like in Section 4, d = 2). A global optimum is not known in advance in this case, and there are no obvious stopping criteria. Thus, the optimization process was stopped after exactly 47 exact function calls, as was the case when CRSA was tested without CORS-RBF. The intent now is to compare the strategies along the entire optimization process, not only final results.

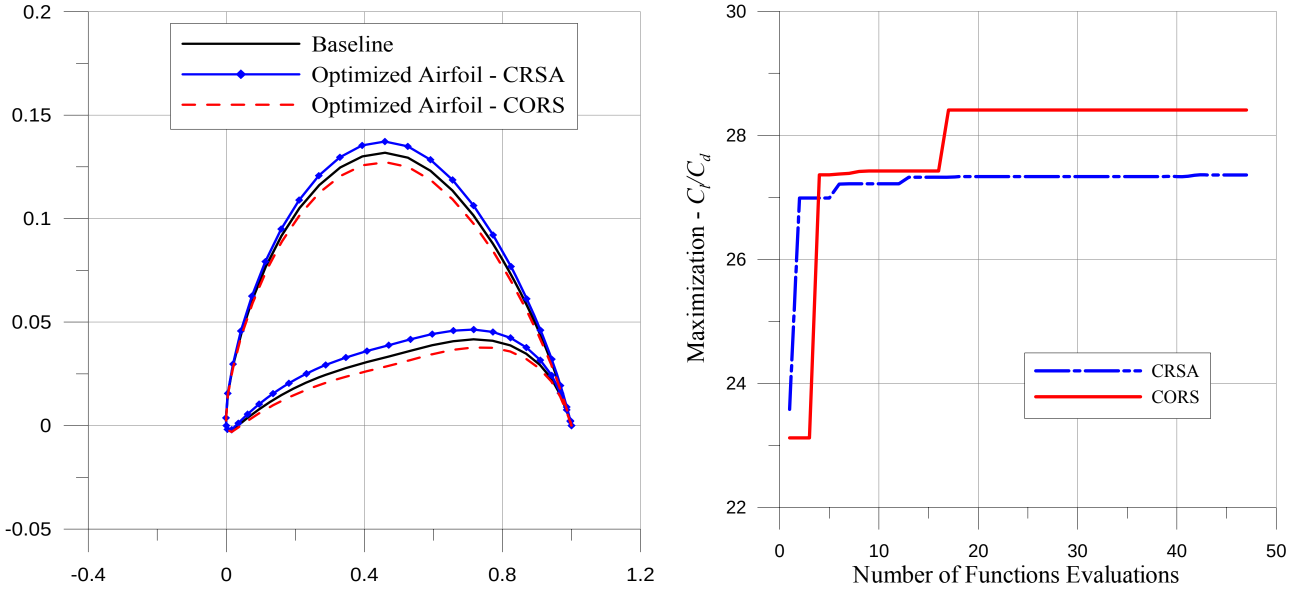
Table 4 shows some results obtained after the 47 exact function calls. One can see that the CORS-RBF strategy was able to obtain a higher lift-to-drag ratio and a lower loss coefficient than CRSA alone. Both strategies produces slightly lower flow deflections as compared with the baseline, but this effect is less pronounced with CORS-RBF. These results are indicative of the global search effectiveness of CORS-RBF — although it might be impossible to evaluate the closeness of the final CORS-RBF results to a global optimum.

Table 4. Optimization results

	Camber	Pitch to chord	C_l	C_d	C_l/C_d	ζ	θ (deflection angle)
Baseline	1.500	1.0000	0.7186	0.0279	25.76	0.0320	24.32°
CRSA	1.540	1.1500	0.8022	0.0293	27.36	0.0294	23.55°

CORS-RBF	1.400	0.9675	0.6846	0.0241	28.41	0.0286	23.98°

Figure 2a shows the final airfoil shapes in comparison with the baseline shape. Only the camber was varied and better parameterizations are obviously conceivable. The airfoil camber obtained with CORS-RBF was lower than the baseline but the pitch-to-chord ratio was lower (see Table 4). With the CRSA alone, the trends were inverted. Observe that a lower pitch-to-chord ratio implies the use of a greater number of blades or/and bigger blades. In either case, more material would be necessary for manufacturing the blades — an issue that would deserve a multiobjective treatment.



(a)

(b)

Figure 2. a) Comparison of baseline and optimized airfoils, b) History of optimization process

Figure 2b shows the history for the convergence process. Incidentally, one of the four points of the initial CORS-RBF data bank was already better than the best point found by CRSA alone. Since the 15 members of the initial CRSA population were always randomly set this event cannot be seen as abnormal. Note that with 6 function evaluations, CRSA was able to find a point which is only 0.6% worse than the corresponding point obtained by CORS-RBF. However, thereafter until the end of the optimization process the improvement obtained by CRSA was around 0.53% while CORS-RBF produced an improvement of 3.8% - 7 more times. Thus, due to the specific nature of this example, the acceleration properties of CORS-RBF is better evaluated viewing the entire evolution instead of the number of exact function evaluations necessary to attain a certain objective.

6. OPTIMAL DESIGN OF AXIAL HYDRAULIC TURBINE

The methodology tested in Section 5 is now extended for the case of the runner of an axial hydraulic turbine with a full 3D model. Only the runner is considered because the consideration of other turbine components, like guide vanes and draft tube, would be beyond the computational resources accessible by the authors for the moment. Further only the blade stagger angles at tip and mid sections are treated as design parameters to be optimized (Fig. 3). Although limited, this parameterization is able to capture representative variations in the runner hydraulic efficiency.

6.1. Initial Design

Based on the blade element theory (lift wing approach) and the hypothesis of free vortex radial equilibrium, a preliminary runner design methodology was developed (Bran & Souza, 1979; Souza, 1991). The main geometric dimensions, such as the hub-to-tip ratio, blade section profiles, stagger angles and radial stacking of the blade cascades from the hub to tip, are thus obtained. The definition of a more general blade would also require other geometric parameters, such as the axial and circumferential stacking effects (sweep and dihedral displacements), among others. In the search for better efficiency, the independent variation of these parameters would result in excessive computational effort and a less feasible design by facing the computing resources available at this time.

The following design parameters were considered: Q = 8 m^3/s (flow rate), H = 6 m (net energy head), Z_{nj} = 600 m (local altitude), T = 25 ºC (local temperature), P_{atm}= 101325 Pa (local pressure), f = 60 Hz (frequency of the electrical system).

Figure 3 shows a scheme of the runner channel, including the stacking position of the profiles and the variation of the profile shapes, chord lengths and stagger angles (β) along the blade radius.

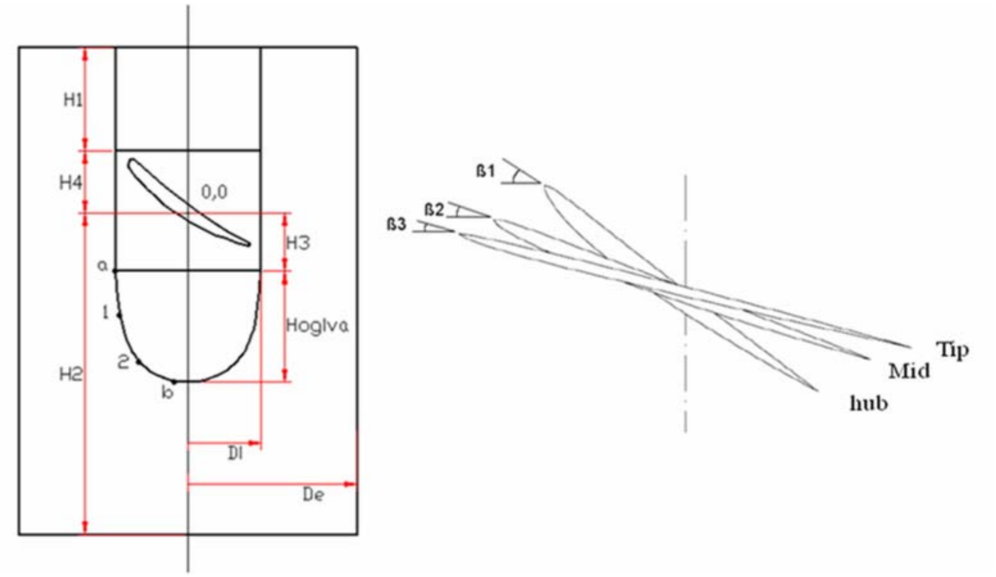


Figure 3. Geometry e periodic channel of the axial turbine runner

6.2. Flow Calculation and Computational Mesh

For the calculation of the flow through the channel of the axial turbine runner one has used the commercial CFD software FLUENT®. For the process of integration with optimization algorithms (CRSA and CORS-RBF), it is extremely important that the parametric mesh construction fits smoothly to the blade shape variations and also presents refinements in regions of higher pressure or velocity gradients, especially those near the walls, where the mesh growth is controlled by the *law of the wall* defined in turbulence models (criterion $Y^+ \approx 1.0$).

A hybrid mesh was chosen, by allowing prismatic elements to be distributed on the wall and a transition to a core of hexahedral elements by means of intermediate tetrahedral elements (Figure 4).

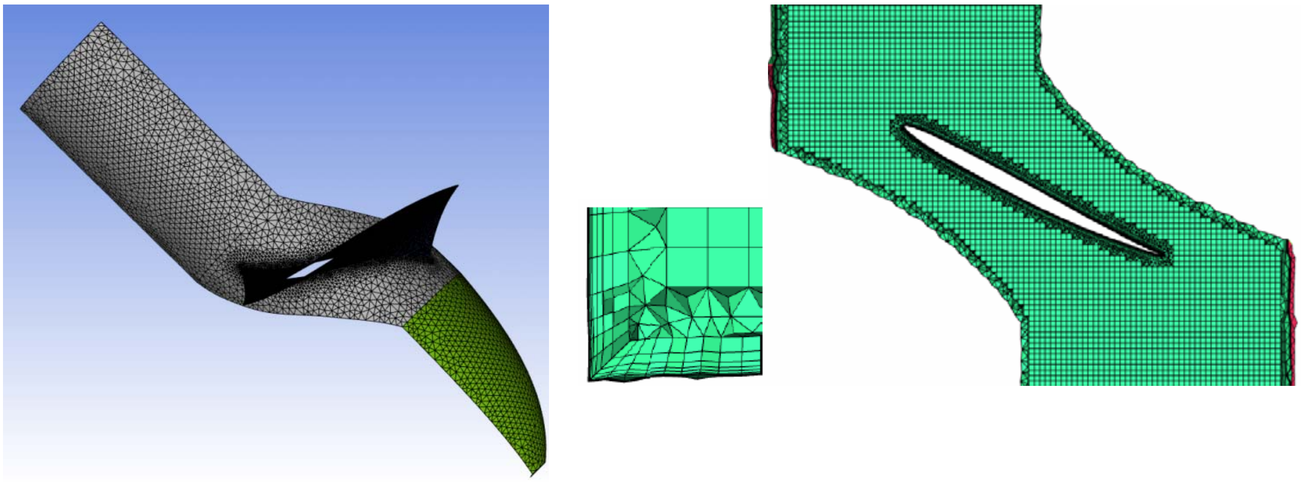
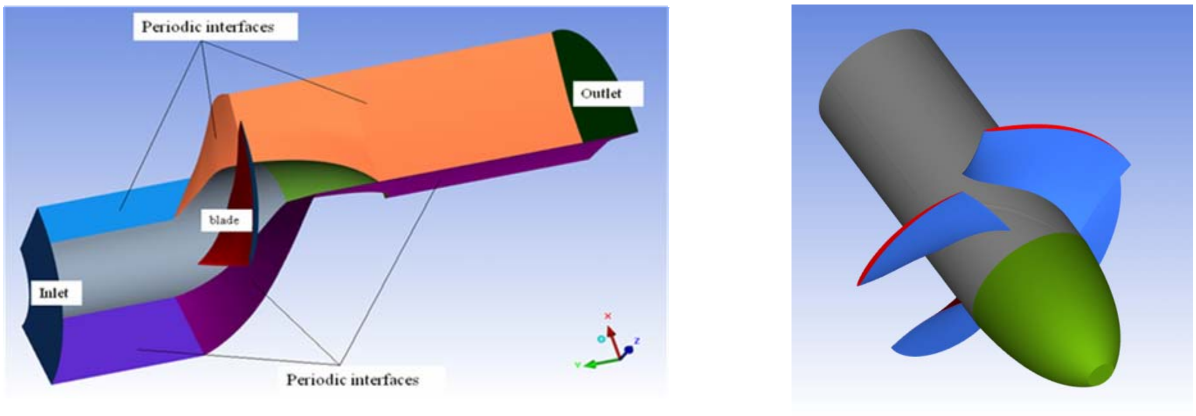


Figure 4. Hybrid mesh, prism layers, tetra and hexa-core, 2699545 cells



(a) (b)

Figure 5. a) Geometry of computational domain, generated automatically from the *script file*,
b) Geometry of runner (replicated)

Based on the principal dimensions of the turbine runner, a *script file* was constructed with editing commands written in Tcl/Tk language, interpreted by the commercial software ICEM-CFD®. This methodology allows the automatic construction of the geometric and parameterized mesh. It is similar to that adopted for the cascade test case in Section 5.2.

Since this initial project does not include the inlet guide vanes, a periodic channel has been generated containing a single blade, since the geometry is perfectly regular and all surfaces are of revolution. For this simplification is appropriate to use the *Single Rotating Frame* SRF model, helping to reduce the computational cost (FLUENT®, 2006).

Figure 5a shows the channel of the runner with the boundary conditions such as: inlet, outlet, blade-wall and periodic interfaces. Figure 5b shows the geometry of the channel periodically replicated, resulting in the runner axial turbine.

Due to its robustness, economy and reasonable accuracy for a wide range of turbulent flows, the k-ε model was chosen for all simulations of this work (Versteeg & Malalasekera, 1995; Wilcox, 1993).

At channel entrance one defines the mass flow rate and the direction of the relative velocity vector, with reference to the blade mid section in cylindrical coordinates. At channel outlet one has imposed the *outflow* condition that guarantees the continuity condition of mass between inlet and outlet surfaces (FLUENT®, 2006)

The FLUENT *journal file* allows a list of commands in sequence such as reading the mesh, check for errors in the mesh, scaling, definition of work units, definition of the fluid material; configuration mathematical model, adjusting the parameters of the sub-relaxation model, adjustment of parameters of convergence; adjustments of operating conditions, definition of boundary conditions, defining interfaces, topology configuration of turbomachinery (*turbo topology*).

FLUENT® also has a tool for analysis of turbomachinery flow field in the "*turbo topology*" file, which allows for the integration of wall pressure and shear torques (due to hub and blade) with respect to a reference system on the runner shaft and so the computation of the power shaft P_s. The hydraulic power P_h is obtained with basis on the computed net head *H,* by calculating the total pressure difference between the runner entrance and outlet. The overall hydraulic efficiency of the runner is defined as: $\eta_T = P_s/P_{h.}$ The value of hydraulic efficiency was achieved within the range of the expected efficiency (Souza, 1979, 1991).

6.3. Optimization Results

The maximization of the overall hydraulic efficiency of the runner has been chosen as objective function. The design variables are the stagger angles at the mid and tip blade sections (β_2 and β_3 in Figure 3). The search intervals were set as [17.75° – 19.75°] for β_2 and [12.45° – 14.45°] for β_3.

First the CRSA was tested without CORS. An initial a population of 9 individuals was randomly set and evaluated. The stopping criterion was the population contraction tolerance below 10^{-4}. Only one run was performed and converged results were obtained after 25 exact function calls.

For the CORS-RBF strategy, the multiquadric function with c = 1 was employed. The initial data bank was set with a two-level full factorial sampling. Like in Section 5.2, a global optimum is not known in advance in this case, and there are no obvious stopping criteria.

Figure 6 shown the results obtained for strategy CORS-RBF with CRSA represented by line red with symbols triangular, and CRSA directly represented by line blue with symbols lozenge, respectively. It may be noted that the strategy was able to obtain a higher efficient hydraulic than CRSA alone.

The optimization algorithm was coded in FORTRAN language. It integrates the ICEM-CFD® *script file* and FLUENT® *journal file* in order to automatically generate the parametrical meshes, read the boundary conditions and model parameters, and perform the flow computations.

Table 5. History of optimization process

CRSA		CORS	
NFE*	η	NFE	η
1	81.9352	1	81.7153
2	84.8153	6	84.5503
12	84.8736	8	85.3719
13	84.9355	25	85.3719
14	84.9427		
25	84.9427		

*Number of Function Evaluations

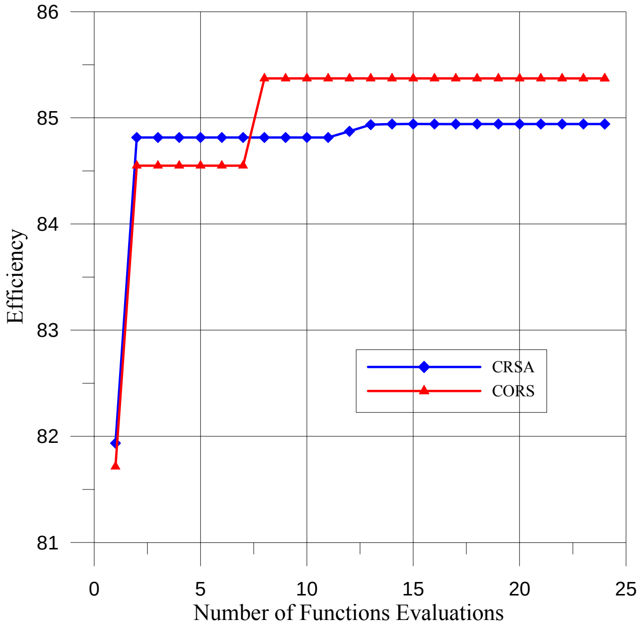
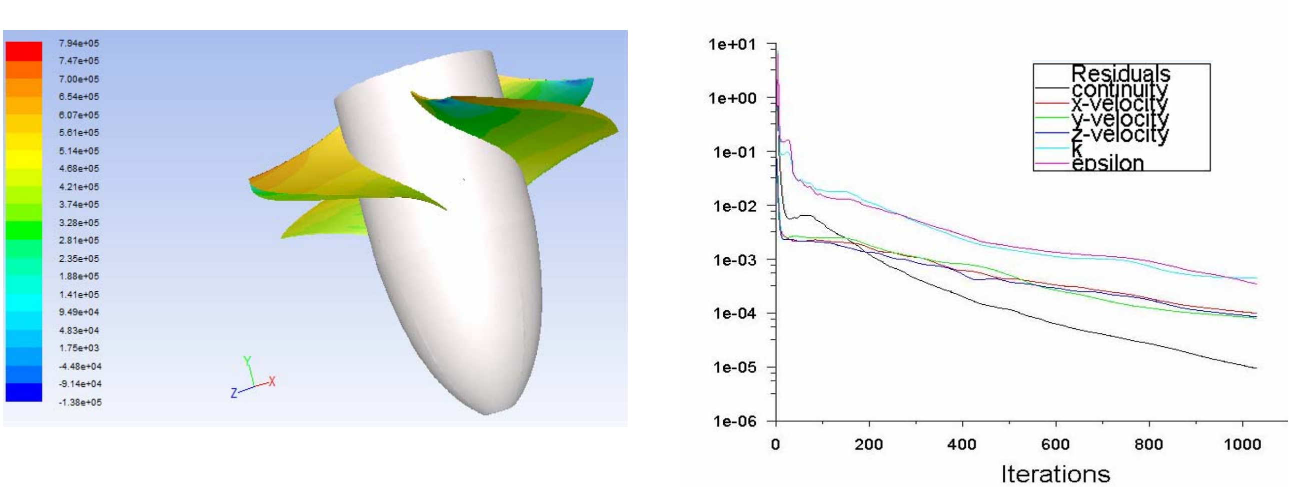


Figure 6. History of optimization process

In Table 5, the occurrences of the efficiency improvement along the optimization are shown. Figure 6 shows this optimization process graphically. Note that the CORS-RBF remained below the CRSA between the 2nd and 8th function evaluation but then it has obtained its best point and remained thereafter until the end of the process, with 0.65% of improvement. On the other hand, CRSA was able to find its best point at the 12th function evaluation, which was 0.57% worse than CORS-RBF. From the best point of the initial population until the end of the optimization process, CRSA obtained an improvement of 0.15%, while CORS-RBF produced an improvement of 0.97% — 6 more times. Like in the example of the blade cascade, the properties of an acceleration of CORS–RBF is best assessed by viewing the whole evolution, rather than the number of function evaluations needed to accurately reach a certain goal.

Some illustrative field results for the best axial turbine found by CORS-RBF are shown in Figures 7. Figure 7a shows the contours of static pressure. Figure 7b shows the iterative convergence of FLUENT®. In this case, each convergence completion had a computational cost equivalent to 3 hours of processing. The objective function was evaluated 25 times, thus resulting in 75 hours of simulation for the optimal design. The use of CORS-RBF is safer than the use of CRSA alone, since better results can be obtained with the same computational effort.



(a) (b)

Figure 7. a) Contours of Static Pressure (Pascal), b) Convergence

7. CONCLUSION

An alternative implementation of the Constrained Optimization using Response Surfaces (CORS) algorithm originally proposed by Regis and Shoemaker (2005) was presented in this work. The response surfaces (metamodels) were constructed using direct and inverse multiquadrics and multiquadrics radial basis functions. The metamodel optimization process is carried out by means of a Controlled Random Search Algorithm (CRSA).

The CORS strategy presented in this paper (CORS-RBF) was applied on the Dixon-Szegö test functions for two values of the shape parameter *c* chosen for each function and 20 independent runs for each test. The best average results obtained compares favorably with the results obtained by Regis and Shoemaker (2005) that employed thin plate splines for metamodel construction and a deterministic algorithm for optimization. Substantial reductions in the number of function evaluations were observed in comparison with results obtained when the CRSA algorithm was applied directly. Further, CORS-RBF has attained a rate of success of 100% in all the tests what did not always occur with CRSA alone. On the other hand one has observed that the shape parameter variation affects somewhat the speed of CORS-RBF. Due to CRSA randomness, one has also observed a relatively large difference between the maximum and minimum number of function evaluations among independent runs. This difference tends to decrease when better values of the shape parameter are employed, i.e., values that increase the speed of CORS-RBF.

Two design examples were also presented: a blade cascade (2D flow) and an axial hydraulic turbine runner (3D flow). The objective function for the first example was the maximization the lift-to-drag ratio with airfoil camber and pitch-to-chord ratio as design variables. For the second example, the objective was the maximization the hydraulic efficiency; the stagger angles at the mid and tip blade sections were chosen as design variables. The CFD code FLUENT® was used as solver. Like it was observed with the Dixon-Szegö test functions, the optimization using CORS-RBF strategy increase the improvement rates of the objective function as compared with CRSA alone.

8. ACKNOWLEDGMENTS

This work was supported by the Brazilian agencies CAPES, CNPq and FAPEMIG.

9. REFERENCES

Ali, M. M., Törn, A. and Viitanen, S., 1997 ‘A Numerical Comparison of Some Modified Controlled Random Search Algorithms’, Journal of Global Optimization, Vol. 11, 377-385.

Ali, M. M. and Törn, A., 2004 ‘Population set-based global optimization algorithms: some modifications and numerical studies’, Computers & Operations Research, Vol. 31, 1703-1725.

Bran, R. & Souza, Z., 1979 ‘Máquinas de Fluxo’, 2ª Edição, Ao Livro Técnico S.A.

Dixon, L. C. W. and Szegö, G., 1978 ‘The global optimization problem: an introduction’, Towards Global Optimization, L.C.W. Dixon, G. Szegö (Eds.), Vol. 2, 1–15. North-Holland, Amsterdam.

Emery, J.C. Herrig, L.J., Erwin, J. R., Felix, R., 1958 ‘Systematic Two- Dimensional Cascade Tests of NACA 65-Series Compressor Blades at Low Speeds’, NACA TR-1368.

Fluent, 2009 "User Guide e Tutorial Guide". *Ansys Inc*.

Forrester, A. I. J., Sóbester, A. and Keane, A. J., 2008 'Engineering Design via Surrogate Modelling - A Practical Guide', John Wiley & Sons Ltd.

Hon, Y.C., Schaback, R., 2001 'On unsymmetric collocation by radial basis functions', Applied Mathematics and Computation, 119, pp. 177–186.

Jones, D. R., 2001a 'A taxonomy of global optimization methods based on response surfaces', Journal of Global Optimization, 21 (4), pp. 345–383.

Jones, D. R., 2001b 'The DIRECT global optimization algorithm', Encyclopedia of Optimization, Floudas, C.A. and Pardalos, P.M. (Eds.), Vol. 1, pp. 431–440. Kluwer Academic Publishers.

Manzanares-Filho, N., Moino, C. A. e Jorge, A. B., 2005 'An Improved Controlled Random Search Algorithm for Inverse Airfoil Cascade Design', 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO-6), paper n. 4451, J. Herskovits, S. Mazorche, A. Canelas (Eds.), ISSMO, Rio de Janeiro, Brazil, 2005 (ISBN: 85–285–0070–5).

Praveen, C. and Duvigneau, R., 2007 'Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization', RR-6151, INRIA, France, 40 p.

Price, W. L., 1977 'A controlled random search procedure for global optimisation', The Computer Journal, Vol. 20, pp. 367-370.

Ramirez, R.G. and Manzanares-Filho N., 2005 'A Source Wake Model for Cascades of Axial Flow Turbomachines', Journal of the Brazilian Society of Mechanical Sciences and Engineering*,* Vol. XXVII-No 3, pp. 288-299.

Regis, R. G. and Shoemaker, C. A., 2005 'Constrained global optimization using radial basis functions'*, Journal of Global Optimization*, Vol. 31, pp. 153–171.

Silva, E. R., Manzanares-Filho, N., 2010 'Metamodeling for Global Optimization using Radial Basis Functions with Cross-Validation Adjustment of the Shape Parameter', 2nd International Conference on Engineering Optimizatio*n*, EngOpt 2010, Lisbon, Portugal.

Souza, Zulcy, 1991, 'Dimensionamento de Maquinas de Fluxo'. Editora Edgard Blücher Ltda. São Paulo.

Vavra, M. H., 1974 'Aero-Thermodynamics and Flow in Turbomachines', Reprint of the edition Published by Wiley, New York, ISBN 0-88275-189-1.

Versteeg, H.K., Malalasekera, 1995 'An Introduction to Computational Fluid Dynamics', Longman, New York.

Wilcox, D.C., 1993 'Turbulence Modeling for CFD', 1. ed. California: DCW.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.