Evolutionary algorithm based optimization of hydraulic machines utilizing a state-of-the-art block coupled CFD solver and parametric geometry and mesh generation tools

Kyriacou, S.1; Kontoleontos, E.2; Weissenberger, S.2; Mangani, L.2; Casartelli, E.3; Skouteropoulou, I.4; Gattringer, M.4; Gehrer, A.4; Buchmayr, M.4
1 Andritz Hydro Zurich, Zurich, Switzerland; 2 Andritz Hydro Linz, Linz, Austria; 3 University of Applied Sciences Luzern, Luzern, Switzerland; 4 Andritz AG Graz, Graz, Austria

Abstract. An efficient hydraulic optimization procedure, suitable for industrial use, requires an advanced optimization tool (EASY software), a fast solver (block coupled CFD) and a flexible geometry generation tool. EASY optimization software is a PCA-driven metamodel-assisted Evolutionary Algorithm (MAEA (PCA)) that can be used in both single- (SOO) and multi-objective optimization (MOO) problems. In MAEAs, low cost surrogate evaluation models are used to screen out non-promising individuals during the evolution and exclude them from the expensive, problem specific evaluation, here the solution of Navier-Stokes equations. For additional reduction of the optimization CPU cost, the PCA technique is used to identify dependences among the design variables and to exploit them in order to efficiently drive the application of the evolution operators. To further enhance the hydraulic optimization procedure, a very robust and fast Navier-Stokes solver has been developed. This incompressible CFD solver employs a pressure-based block-coupled approach, solving the governing equations simultaneously. This method, apart from being robust and fast, also provides a big gain in terms of computational cost. In order to optimize the geometry of hydraulic machines, an automatic geometry and mesh generation tool is necessary. The geometry generation tool used in this work is entirely based on b-spline curves and surfaces. In what follows, the components of the tool chain are outlined in some detail and the optimization results of hydraulic machine components are shown in order to demonstrate the performance of the presented optimization procedure.

1. Introduction

The optimization of the hydraulic performance of a hydraulic machine by modifying the geometric shape of its components can be a challenging task; not only because an advanced optimization software is required, but also because the necessary components of the toolchain employed within the optimization procedure should enable the optimization loop to run robustly, accurately and in a time efficient way.

In this paper the main components of the toolchain used for the hydraulic optimization of a pump impeller and turbine runner are presented. As shown in Figure 1, this toolchain consists essentially of a parametric geometry generation tool, a parametric mesh generation tool, an incompressible Computational Fluid Dynamics (CFD) solver and an efficient optimization platform (herein the Evolutionary Algorithm System - EASY). For the initiation of the optimization loop, a set of user-defined geometric parameters (design variables) is generated by EASY in order to obtain a set of new geometries (to be referred to as candidate solutions). For each one of these geometries, a numerical grid is subsequently created and a CFD simulation is then carried out. The post-processing of the CFD results returns the values of the user-defined objective function(s), which for the sake of this example are the hydraulic efficiency (eta, F1), the cavitation (sigma, F2) and the hydraulic head (F3). The values
of the objective function(s) indicate the quality of each candidate solution and are subsequently used by EASY to generate the next generation of candidate solutions.

![Toolchain for hydraulic optimization](image)

**Figure 1:** Toolchain for the hydraulic optimization of a pump impeller or a turbine runner

In what follows, the Optimization platform, the geometry and mesh generation tools, and the incompressible CFD solver employed in this work are presented and demonstrated in two typical hydraulic design cases.

## 2. Optimization Platform

Evolutionary Algorithms (EAs) have found widespread use as a means to solve hydraulic design optimization problems [1], [2], [3], [4]. The main advantages of EAs are their ability to reach the global optimum without being trapped into local optima, to compute Pareto fronts of optimal solutions in Multi-Objective Optimization (MOO) problems and to accommodate any ready-to-use analysis/evaluation software as a black-box tool, such as the CFD codes used in the applications of this paper, without requiring access to its source code. In fact, the only prerequisite for carrying out an EA-based optimization is the availability of an appropriate evaluation software (considered as a black-box tool by the EA) and well-defined objective functions and design variables.

However, solving optimization problems associated with computationally demanding evaluation software can become expensive. To reach the optimal solution(s), EAs may require a great number of objective function evaluations, thus increasing the CPU cost. The optimization of hydraulic machinery, which relies on expensive CFD software, is a typical example of this. In order to reduce the wall-clock time of EA-based optimizations, various methods have been developed. The most known among them rely on the use of surrogate evaluation models (also known as ‘metamodels’), giving rise to the so-called Metamodel-Assisted EAs (MAEAs) [5], [6]. In MAEAs, metamodels are used to approximately evaluate the outcome of the problem-specific evaluation model (i.e. the CFD code), after being trained on a number of already evaluated individuals. In this paper, artificial neural networks are used as metamodels.

Metamodels can be coupled with EAs in an either on-line or off-line way, depending on whether their training takes place during (on-line) [7], [8], [9], [10] or separately from the evolution (off-line) [11], [12]. EAs supported by an off-line trained metamodel proceed by (a) statistically sampling the search space, (b) evaluating the samples, (c) training a single metamodel for the entire search space on the evaluated samples, and (d) performing the EA-based optimization based exclusively on the ‘inexpensive’ metamodel. A limited number of optimization loops, during which new individuals are evaluated and the metamodel is upgraded if its predictions for the current optimal solutions deviate
from the outcome of the problem-specific model, follow. EAs supported by off-line trained metamodels are beyond the scope of this paper.

This paper is concerned with EAs assisted by on-line trained metamodels which are alternated with the problem-specific evaluation model during the evolution, either periodically or by switching from metamodels to the exact model and vice-versa according to several criteria. The on-line trained metamodels used in this paper are in conformity with the method presented in [7], [10] and [13]. With the exception of few starting generations, all population members are approximately evaluated using local metamodels trained on the fly and only a limited number of the most promising among them are then re-evaluated on the problem-specific model. This is referred to as an EA assisted by the Inexact Pre-Evaluation (IPE) of the population members within each generation. The term ‘inexact’, rather than ‘approximate’, was used on purpose to emphasize that MAEAs outperform EAs by even using ‘non-exact’ values of the objective function(s).

As far as modern industrial optimization problems, such as hydraulic design problems, are concerned, several objectives and/or constraints and a great number of design variables (N) may be involved. In conventional EAs, the high number of design variables causes efficiency deterioration, since they require more evaluations and, thus, increase the computational cost for reaching the optimal solution, even in well-posed problems. MAEAs are also badly affected by N since both the metamodels’ training time and the prediction error increase noticeably. The loss in efficiency due to the ‘less-accurate’ metamodel predictions is in practice superimposed on the performance degradation due to the slow evolution caused for the same reason. A remedy is to decrease the problem dimension via dimension reduction techniques. To this end, Principal Component Analysis (PCA) [14] is one of the most widely used techniques. For milder N values (some tens of design variables), one may alternatively use factor analysis [15], discriminant analysis [16], multidimensional scaling [17], projection pursuit [18], etc. To tackle optimization problems with an excessive number of design variables and/or objective functions with complex landscapes, new evolution operators, the so-called PCA-driven ones, and PCA driven metamodels have been devised and presented in [19]. The PCA-driven evolution operators, with the new design space coordinates resulting from the PCA of the current approximation of the elite set, has been proved [1] to improve the performance of the underlying EA. The PCA-driven metamodels reduce the dimension of the problem during their training i.e. by the use of a much smaller number of sensory units. The latter, along with the fact that the use of metamodels starts much earlier, leads to much better performing MAEAs [20].

3. Parametric Geometry and Mesh Generation Tools

In order to set up an optimization loop for hydraulic shapes by means of CFD; a fast, robust and reliable parametric geometry generation tool is necessary. A crucial prerequisite for such a tool is that, within the user-defined parametric limits, it should always manage to create a blade shape of good geometric quality. In the case of turbine runners or pump impellers this means that the resulting blades should consist of sufficiently smooth surfaces and have a suitable thickness distribution (be manufacturable).

In this work a geometry generation tool that is almost entirely based on B-spline curves and surfaces is employed. Without loss in generality, the blade shape can be modelled by superimposing a parameterized thickness distribution onto an also parameterized mean camber surface. The parametric setup employs control points for the blade profiles and blade curvature in the spanwise and streamwise direction. The design variables are the control points’ coordinates of the Bezier curves representing the spanwise distributions of (a) the mean camber surface angles at the leading (LE) and trailing (TE) edges, (b) the circumferential position of the blade LE and TE, (c) the mean camber surface curvature and (d) the blade thickness. In Figure 2 an example of the obtained blade surface for the pump impeller case is illustrated.
The parametric setup of the employed grid generation tool matches the parametric setup of the geometry generation tool. Therefore, the grid can be easily aligned with the geometry. The parametric grid generator enables the use of the same mesh topology and refinement for all simulated cases. This abides by the requirement of always using similar meshes in order to guarantee comparability of quantitative results.

4. Block Coupled Incompressible CFD Solver

Speed and robustness are among the most important requirements for any software that is plugged into an optimization loop. When programming a CFD code the best way of combining both requirements is to couple the governing equations implicitly, since resolving the pressure-velocity coupling is essential for the performance of any CFD code. However until today the SIMPLE family of algorithms [19], which couples the governing equations only by means of sub-looping, solving sequentially each governing equation, still remains the predominant methodology used in the CFD community. Therein a segregated approach in resolving the pressure velocity coupling is followed. Compared to block coupled implicit algorithms, segregated algorithms lack scalability with mesh size robustness and calculation speed, which is inherent partly due to the under-relaxation needed to stabilize the algorithm.

In order to overcome these shortcomings Mangani et al. [22] [23] developed a block coupled incompressible solver using the leading open-source CFD library OpenFOAM as programming platform [24]. This pressure-based solution procedure has originally been published by Darwish et. al. [25]. Therein the algebraic equations resulting from the Navier-Stokes equations are solved simultaneously. To enhance computational performance an algebraic multi-grid solver following Keller [26] has been implemented and used for the solution of the block-coupled system of equations. Simulations of turbo-machinery applications in which components of the calculation domain rotate are often treated in rotational reference frame. Using a special transport theorem the governing equations can be rewritten for this rotational reference frame, which makes it possible to employ steady-state calculations.

The resulting set of Navier-Stokes equations reads [23],

\[ \nabla \cdot \mathbf{u} = 0 \]
\[ \nabla \cdot (\mathbf{u} \cdot \mathbf{u}) + \mathbf{\Omega} \times \mathbf{u} = -\frac{1}{\rho} \nabla p + \nabla \cdot \left( \nu_{\text{eff}} (\nabla \mathbf{u}) \right) \]

Equation 1: Navier-Stokes equations in rotational reference frame formulation
Where $\Omega$ is the domain’s rotational velocity, and $\mathbf{u}_r$ is defined as,

$$\mathbf{u}_r = \mathbf{u} - \Omega \times \mathbf{r}$$

**Equation 2: Relation between velocities in stationary and relative rotational frame of reference**

The discretization is then cast into a block coupled linear system of equations, which is solved simultaneously. The conceptual difference compared to segregated, SIMPLE-like algorithms is shown in Equation 3 and Equation 4.

$$
\begin{bmatrix}
    a_{uu}^C & a_{uv}^C & a_{uw}^C & a_{up}^C \\
    a_{vu}^C & a_{vv}^C & a_{vw}^C & a_{vp}^C \\
    a_{wu}^C & a_{wv}^C & a_{ww}^C & a_{wp}^C \\
    a_{pu}^C & a_{pv}^C & a_{pw}^C & a_{pp}^C
\end{bmatrix}
\begin{bmatrix}
    u_C \\
    v_C \\
    w_C \\
    p_C
\end{bmatrix}
+
\sum_{NB}
\begin{bmatrix}
    a_{uNb}^C & a_{vNb}^C & a_{wNb}^C & a_{pNb}^C \\
    a_{vNb}^C & a_{wNb}^C & a_{wNb}^C & a_{pNb}^C \\
    a_{wNb}^C & a_{wNb}^C & a_{wNb}^C & a_{pNb}^C \\
    a_{pNb}^C & a_{pNb}^C & a_{pNb}^C & a_{pNb}^C
\end{bmatrix}
\begin{bmatrix}
    u_{NB} \\
    v_{NB} \\
    w_{NB} \\
    p_{NB}
\end{bmatrix}
=
\begin{bmatrix}
    b_u^C(u,v,w,p) \\
    b_v^C(u,v,w,p) \\
    b_w^C(u,v,w,p) \\
    b_p^C(u,v,w,p)
\end{bmatrix}
$$

**Equation 3: Discretized set of equations for block-coupled pressure-velocity coupling**

$$
\begin{align*}
    a_{u}^C \cdot u_C + \sum_{NB} a_{uNb}^C \cdot u_{NB} &= b_u^C(u,v,w,p) \\
    a_{v}^C \cdot v_C + \sum_{NB} a_{vNb}^C \cdot v_{NB} &= b_v^C(u,v,w,p) \\
    a_{w}^C \cdot w_C + \sum_{NB} a_{wNb}^C \cdot w_{NB} &= b_w^C(u,v,w,p) \\
    a_{p}^C \cdot p_C + \sum_{NB} a_{pNb}^C \cdot p_{NB} &= b_p^C(u,v,w,p)
\end{align*}
$$

**Equation 4: Discretized set of equations for pressure-velocity coupling in segregated algorithms**

Segregated algorithms operate using many sub-loops to account for inter-equation coupling, continuously updating the RHS of Equation 4, which contains field values of previous iteration steps. Additionally, under-relaxation of the governing equations is needed to assure that the solution process remains numerically stable. In block-coupled algorithms sub-looping is also applied in order to update second order derivative terms and non-linear terms. However, in contrast to segregated algorithms, the inter-variable coupling is still much stronger and fewer sub-loops are needed. Moreover, under-relaxation can be completely avoided using the so-called false transient time stepping. The solution of its discretized system of equations therefore, results to be numerically much more stable than that of segregated algorithms and also turns out to be significantly faster in terms of calculation time, which has been demonstrated by Mangani et al. [22] [23]. A more detailed comparison between segregated and block-coupled solvers can be found in [27].

The solver described in [23] has been used for the optimization runs carried out in this paper. Therein a k-Omega turbulence model has been solved in addition to momentum and continuity equations.
5. Hydraulic Design Optimization Cases

5.1. Francis Case

The first application presented in this work is concerned with the 3-objective redesign (rehabilitation) of a Francis runner with relatively high specific speed. In this problem the first objective ($F_1$) is related to the ‘quality’ of the outlet velocity profile. Practically this objective quantifies the draft-tube coupling quality of the designed runner. The second objective ($F_2$) is related to the cavitation behavior of the runner. It requires the maximization of the minimum pressure observed on the runner surface. The third objective ($F_3$) is concerned with the efficiency of the runner.

The runner geometry is parameterized by superimposed predefined thickness distribution onto a mean-camber surface. Regarding the latter, the spanwise metal angle (beta) distributions at leading (LE) and trailing (TE) edge are described using B-spline curves with 7 and 8 control points, respectively. The spanwise coordinates of all control points are fixed, which gives rise to 15 design variables. For the circumferential positions of the LE and TE (theta), two B-spline curves with 4 and 8 control points respectively, with fixed spanwise coordinates are used, yielding another 12 design variables. The curvature of the mean camber surface is controlled via 12 design variables and the superimposed thickness with an additional 18. The hub generatrice is parameterized with 10 design variables and the shroud one is fixed. Finally, the LE and TE meridional positions are parameterized with 6 and 9 design variables respectively, thus reaching a total of 82 design variables.

The evaluation software used herein is the coupled OpenFoam-based solver presented in section 4 of this paper, enhanced with the so-called auto-alfa functionality. Auto-alfa functionality allows for the intra-solver, inlet swirl level (guide vane angle) adaptation so as to achieve a predefined head difference level hence ensuring computation at the desired operating point.

This case was studied using a MAEA (PCA) with $\mu=10$ parents and $\lambda = 30$ offspring. The IPE phase started after 200 exactly evaluated individuals were stored in the DB. The $\lambda_e = 6$ most promising individuals according to the metamodel prediction, among the 30 offspring, were then evaluated using the CFD solver. During the initial (non IPE) generations, 30 concurrent evaluations were sent to 30 different CPUs. As soon as the IPE phase started, the computational burden was reduced to 6 concurrent evaluations, thus freeing valuable resources. At the cost of 600 CFD simulations, the computed Pareto front of non-dominated solutions can be seen in Figure 3.
Figure 3: Pareto Front of non-dominated solutions at the cost of 600 CFD simulations. $F_1$ shows the draft-tube coupling quality. $F_2$ the cavitation safety index, smaller $F_2$ translates into larger minimum-pressure observed on the runner and designs with $F_2 < 0$ are considered to be cavitation free. $F_3$ shows the normalized runner efficiency. The size of the points is proportional to $F_3$. Initial design is clearly worst in all three objectives and suffering from cavitation.

From this Pareto front, design A was selected by an experienced designer to undergo further analysis. The comparison between the optimized design A and the initial design, as far as the three objectives are concerned, can be seen in Figure 5 and Figure 6. Optimal design A outperforms the initial design in all three objectives. It facilitates higher runner efficiency by 0.8%. It has an almost perfect agreement with the outlet velocity target distributions which ensures good draft-tube coupling and is free of cavitation with a significant safety margin. The geometric differences between the optimal design A and the initial design are significant and can be observed in Figure 4.

Figure 4: Geometric differences between Optimal design A (red) and initial design (white)
Figure 5: Analysis of objectives for the initial design. High disagreement between target and actual outlet velocity profiles both at the circumferential and meridional plane (bottom left). The runner efficiency is high. Cavitation is observed at the LE near shroud location (top and bottom right).

Figure 6: Analysis of objectives for Design A. Very good agreement between target and actual outlet velocity profiles, both at the circumferential and meridional plane (bottom left). A 0.8% increase at runner efficiency, with respect to the initial design, is achieved. The runner is free of cavitation, with a significant safety margin (top and bottom right).
5.2. **Radial Pump Impeller Case**

The second application represents the design-optimization of a pump impeller with a specific speed of \( n_q = n^*Q^{0.5}/H^{0.75} = 65 \) and is handled as a 2-objective optimization problem. The first objective \( (F_{1t}) \) requires a maximum weighted efficiency for three different operating points. The second objective \( (F_{2t}) \) is related to the behavior of the impeller regarding cavitation. This second objective requires the maximization of the minimum value of a pressure histogram value, which corresponds to a predefined integrated surface on the impeller. The pressure histogram value is also cumulated over three different operating points.

A total of 26 design variables for the geometry generation are altered by EASY comprising B-spline parameters at leading (LE) and trailing (TE) edge, inlet and outlet blade angles, blade enrollement angles and blade curvature parameters. The meridional contour of the impeller was fixed.

As in the previous case, the evaluation software used is the coupled OpenFoam-based solver. MAEA (PCA) was used in this case as well, with \( \mu = 10 \) parents and \( \lambda = 30 \) offspring. Metamodels (IPE) were applied after the first 60 entries have been stored in the DB. The optimization was configured to terminate after completing 20 generations.

The Pareto front in Figure 7 shows that the optimized candidate geometries in generation 20 (Gen. 20) have significant increase in hydraulic efficiency, as well as considerable gain in terms of cavitation, in respect to the initial generation (Gen. 0) which contains the initial design.

![Figure 7](image-url)

**Figure 7:** Pareto front showing non dominated solutions in generation 20 (Gen. 20) of the optimizer in comparison with the four initial solutions (Gen.0). An increase in both normalized efficiency and cavitation behaviour is observed (smaller \( F_{2t} \) translates into larger pressure histogram values).

In Figure 8, an indicative initial blade shape (solid surfaces) is compared to an indicative individual of generation 20 (dotted). The geometrical differences between the blades are apparent. For the final blade shape, the geometric features which have been gained by means of optimization have been adopted, leading to an optimal pump impeller that numerically meets all requirements set for the given project.
Figure 8: A selected original geometry (with solid surfaces) vs. a selected geometry of Gen. 20 (with dots).
References

[1] S.A. Kyriacou, S. Weissenberger and K. Giannakoglou, "Design of a matrix hydraulic turbine using a metamodel-assisted evolutionary algorithm with PCA-driven evolution operators," International Journal of Mathematical Modelling and Numerical Optimization (SIM: Simulation-Based Optimization Techniques for Computationally Expensive Engineering Design Problems), vol. 3, no. 2, pp. 45-63, 2012.

[2] I.A. Skouteropoulou, S.A. Kyriacou, V.G Asouti, K.C. Giannakoglou, S. Weissenberger and P. Grafenberger, "Design of a Hydromatrix turbine runner using an Asynchronous Algorithm on a Multi-Processor Platform," in 7th GRACM International Congress on Computational Mechanics, Athens, 2011.

[3] R.G. Raimunda da Silva, E. Ramirez Camacho and N. M. Filho, "Global optimization based on metamodel construction applied to design axial turbomachinery cascades using cfd," in 25th IAHR, Symposium on Hydraulic Machinery and Systems, Timisoara, 2010.

[4] P. Grafenberger, E. Parkinson, C. Georgopoulou, S. Kyriacou and K. Giannakoglou, "Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel-assisted evolutionary algorithm: Part 2: Applications," in 24th IAHR, Symposium on Hydraulic Machinery and Systems, Foz do Iguaçu, 2008.

[5] K. C. Giannakoglou, "Design of Optimal Aerodynamic Shapes Using Stochastic Optimization Methods and Computational Intelligence," Progress in Aerospace Sciences, vol. 38, no. 1, pp. 43-76, 2002.

[6] Y. Jin, M. Olhofer and B. Sendhoff, "A Framework for Evolutionary Optimization with Approximate Fitness Functions," IEEE Transactions on Evolutionary Computation, vol. 6, no. 5, pp. 481-494, 2002.

[7] M. K. Karakasis, A. P. Giotis and K. C. Giannakoglou, "Inexact Information Aided, Low-Cost, Distributed Genetic Algorithms for Aerodynamic Shape Optimization," International Journal for Numerical Methods in Fluids, vol. 43, no. 10-11, p. 1149–1166, 2003.

[8] H. Ulmer, F. Streichert and A. Zell, "Evolution Strategies Assisted by Gaussian Processes with Improved Pre-Selection Criterion," in Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC '03), vol. 1, Piscataway, NJ, IEEE Press, 2003, p. 692–699.

[9] Y. S. Ong, K. Lum, P. B. Nair, D. M. Shi and Z. K. Zhang, "Global Convergence of Unconstrained and Bound Constrained Surrogate-Assisted Evolutionary Search in Aerodynamic Shape Design," in Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC '03), Piscataway, NJ, IEEE Press, 2003, p. 1856–1863.

[10] M. K. Karakasis and K. C. Giannakoglou, "On the Use of Metamodel-Assisted, Multi-Objective Evolutionary Algorithms," Engineering Optimization, vol. 38, no. 8, p. 941–957, 2006.

[11] D. Büche, N. Schraudolph and P. Koumoutsakos, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models," IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, vol. 35, no. 2, p. 183–194, 2005.

[12] W. Shyy, N. Papila, R. Vaidyanathan and K. Tucker, "Global Design Optimization for Aerodynamics and Rocket Propulsion Components," Progress in Aerospace Sciences, vol. 37, no. 1, p. 59–118, 2001.

[13] K. C. Giannakoglou, A. P. Giotis and M. K. Karakasis, "Low-Cost Genetic Optimization Based on Inexact Pre-Evaluations and the Sensitivity Analysis of Design Parameters," Inverse Problems in Engineering, vol. 9, no. 4, p. 389–412, 2001.

[14] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Upper Saddle River, NJ: Prentice-Hall, 1998.

[15] D. Bartholomew, M. Knott and I. Moustaki, Latent Variable Models and Factor Analysis: A Unified Approach, 3rd Edition., Chichester, UK: Wiley, 2011.

[16] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, 2nd Edition., New York: Wiley, 2004.

[17] I. Borg und P. Groenen, Modern Multidimensional Scaling: Theory and Applications, 2nd
[18] P. J. Huber, "Projection Pursuit," *The Annals of Statistics*, vol. 13, no. 5, p. 435–475, 1985.

[19] S. Kyriacou, *Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery*, PhD thesis, National Technical University of Athens, 2013.

[20] S. Kyriacou, V. Asouti and K. Giannakoglou, "Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery," *Engineering Optimization*, vol. 46, no. 7, pp. 895-911, 2013.

[21] S. Patankar and D. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, vol. 15, 1972.

[22] L. Mangani, M. Buchmayr and M. Darwish, "Development of a Novel Fully Coupled Solver in OpenFOAM: Steady State Incompressible Turbulent Flows," *accepted by Numerical Heat Transfer Part B: Fundamentals*, 2013.

[23] L. Mangani, M. Buchmayr and M. Darwish, "Development of a Novel Fully Coupled Solver in OpenFOAM: Steady State Incompressible Turbulent Flows in Rotational Reference Frames," *accepted by Numerical Heat Transfer Part B: Fundamentals*, 2013.

[24] H. Weller, G. Tabora, H. Jasak und C. Fureby, „A tensorial approach to computational continuum mechanics using object-oriented techniques,“ *Computers in Physics*, 1998.

[25] M. Darwish, I. Sraj und F. Moukalled, „A coupled finite volume solver for the solution of incompressible flows on unstructured grids,” *Journal of Computational Physics*, 2008.

[26] S. Keller, „Additive correction multigrid method applied to diffusion problems with unstructured grids,“ in *ENCIT 2004, Proceedings of the 10o Brazilian Congress of Thermal Sciences and Engineering*, 2004.

[27] M. Buchmayr, submitted PhD thesis: *Development of Fully Implicit Block Coupled Solvers For Incompressible Turbulent Flows*, Graz, 2014.