Palacký University Olomouc
Faculty of Science
Department of Experimental Physics

Dissertation

Shape optimization of hydraulic surfaces of the impeller and stator parts of hydrodynamic pumps

Author:	Mgr. Tomáš Krátký
Studijní program:	Fyzika
Studijní obor:	Aplikovaná fyzika
Forma studia:	Kombinovaná
Vedoucí práce:	doc. Ing. Luděk Bartoněk, Ph.D.
Termín odevzdání práce:	10. prosince 2020

Prohlašuji, že jsem předloženou diplomovou práci vypracoval(a) samostatně pod vedením doc. Ing. Luďka Bartoňka, Ph.D. a že jsem použil(a) zdrojů, které cituji a uvádím v seznamu použitých pramenů.

V Olomouci ……………….   ...………………………......

Bibliografická identifikace:

Jméno a příjmení autora	Tomáš Krátký
Název práce	Tvarová optimalizace hydraulických povrchů oběžného kola a statorových částí hydrodynamického čerpadla
Typ práce	Doktorská (Dizertační)
Pracoviště	Katedra experimentální fyziky
Vedoucí práce	doc. Ing. Luděk Bartoněk, Ph.D.
Rok obhajoby práce	2021
Abstrakt	Práce se zabývá tvarovou optimalizací hydraulických částí čerpadel s využitím numerického modelování proudění.

Klíčová slova	Pumps, Shape optimization, CFD
Počet stran	81
Počet příloh	0
Jazyk	Anglický

Bibliographical identification:

Autor’s first name and surname	Tomáš Krátký
Title	Shape optimization of hydraulic surfaces of the impeller and stator parts of hydrodynamic pumps
Type of thesis	PhD
Department	Department of Experimental Physics
Supervisor	doc. Ing. Luděk Bartoněk, Ph.D.
The year of presentation	2021
Abstract	This work deals with shape optimization of hydraulic parts of centrifugal pumps, based on computationally expensive numerical simulations.
Keywords	Pumps, Shape optimization, CFD
Number of pages	81
Number of appendices	0
Language	English

Poděkování

Na tomto místě bych rád poděkoval svému školiteli, doc. Ing. Luděk Bartoňkovi, Ph.D., za pomoc se studiem a s přípravou dizertační práce.

Dále pak svým kolegům se společnosti SIGMA za rady:

- RNDr. Milan Sedlář, CSc.

- Ing. Lukáš Zavadil, Ph.D.

- Ing. Jakub Stareček

Prezentované výsledky byly dosaženy s pomocí následujících grantových projektů:

- TE02000232 Výzkumné centrum speciálních rotačních strojů

- FV10302-Progresivní řešení hydraulického designu čerpadel extrémních výkonů pro „Voda-Sucho“

- CZ.02.1.01/0.0/0.0/17_049/0008408 Hydrodynamický design čerpadel

- e-Infrastruktura CZ

Content

1. Introduction

Mathematical optimization is an important part of mathematical theory. It can be applied to many theoretical and real-world problems. With the rapid development of numerical modelling in last decades, engineering optimization has become a hot topic. By connecting the tools of mathematical optimization and numerical simulations, it is possible to guide many design and technical problems, such as shape / topology optimization, or inverse problems. In modern society, virtually every product has been optimized with respect to some objectives.

In this work, the goal is to present how the optimization can be used in pump industry. To succeed in the competition, every pump design needs to be carefully optimized. This is done on multiple levels – mechanical and material solutions, production technologies and costs and hydraulic shapes. The hydraulic design typically utilizes exact analytical formulas, combined with data and experience-based corrections. For this purpose, numerical modelling of fluid is commonly used. It is capable of describing the pump performance and operation with a high level of accuracy, and gives the designer a tool to fine-tune the pump and balance the trade-offs between various conflicting objectives.

Here we focus on shape optimization as a part of the hydraulic design. The pump design uses input in the form of numerical geometry parameters (dimension, blade angles etc.) and numerical output such as performance characteristics (efficiency at a given flow-rate etc.). Thus, in principle it is possible to view the hydraulic design as an optimization task. When performed by a hydraulic expert, it is approached as a manual, semi-intuitive optimization. However, numerical simulations can be automated, and connected with an optimization code. This way, we can utilize the full potential of modern optimization methods, developed specifically to help aiding the design process.

Still, there remain many challenges related to the practical incorporation of optimization into the hydraulic design process. This thesis deals mostly with the practical side of the pump design and optimization. On multiple selected examples, the goals and practical challenges are shown and explained.

The content is organized as follows:

1. Basic terminology and concepts of (centrifugal) pumps are introduced, and the development goals are described.
2. A brief introduction of hydraulic design and its goals. Connection between (shape) optimization and hydraulic design.
3. Basic description and classification of approaches and methods that are commonly used for simulation-driven, computationally expensive optimization, is given.
4. The real-world cases of shape optimization - these are the main part of this thesis. Each case is thoroughly described – introduction, goals of the optimization, parametric model and automation, optimization method, results and the outcome, including summary and discussion. The cases are ordered by increased complexity of the optimization.
5. In the end, the overall results and experience are summarized. Future plans and prospects, and advantages and limits of the shape optimization in the process of hydraulic design are discussed.

In next chapter, a brief introduction of the centrifugal pumps follows.

2. Centrifugal Pumps - basic concepts[1]

Centrifugal pumps are machines that convert mechanical energy to a kinetic energy of the moving fluid. It typically comprises of a rotating impeller, driven by an engine through a shaft, a stator part and a casing (hub and shroud). The impeller adds the kinetic energy to the fluid, and the stator helps to convert the kinetic energy into static pressure. They are utilized in many areas, such as water or petroleum pumping. Any pump can be described by its performance and operational characteristics.

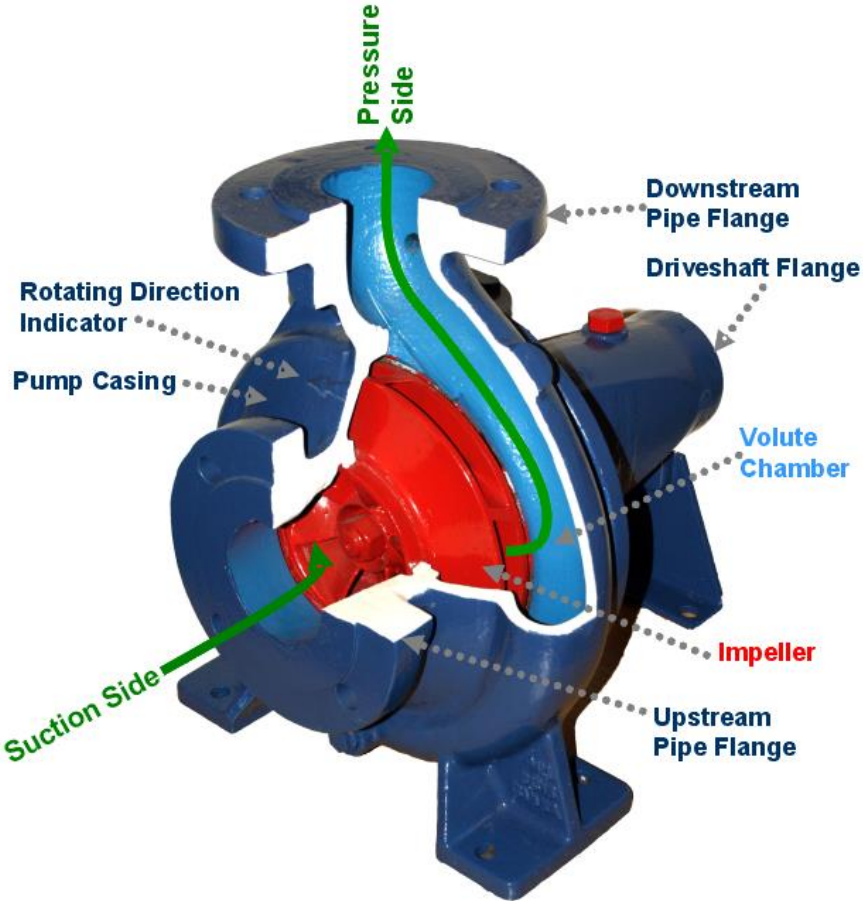

Figure 1: Centrifugal pump schematics. Source: By Fantagu - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4332102

The most important characteristics related to the hydraulic performance of the pump are:

1. flow rate Q - volume of fluid transported per time unit. The flow rate where the pump reaches peak efficiency is called optimum and denoted Q_{OPT}.
2. Head H – difference of the total pressure between the outlet and the inlet.
3. Power P – power needed for operating the pump, i.e., the mechanical energy spent on the pump operation.
4. Efficiency η - the ratio between energy transferred to the fluid kinetic energy and the power needed for the pump operation.
5. $NPSH_3$, the so-called suction ability of the pump – it denotes the Net Positive Suction Head (i.e., the inflow pressure), below which the pump performance (measured by head) is degraded by 3 percent.

A pump is typically operated in a range of flow rates, typically called working range. Thus, instead of a single value, the dependence on Q needs to be given for all the characteristic. For example, Q-H, Q-P and Q- η and $NPSH_3$ curves are often used as performance metrics.

[1] The naming and abbreviations found in this chapter follow (1).

2.1. (Centrifugal) pumps classification and specific speed

The size of centrifugal pumps can vary from a few watts up to hundreds of MWs. However, all these pumps can be classified by the so-called specific speed.

$$n_s = n \cdot \frac{\sqrt{Q_{OPT}}}{(H_{OPT})^{0.75}}$$

Where *n* is the pump speed (in *rpm)* and H_{OPT} is the head at Q_{OPT}. Pumps range from *low Q, high H* (low specific speed – radial pumps) to *high Q, low H* (high specific speed – axial pumps).

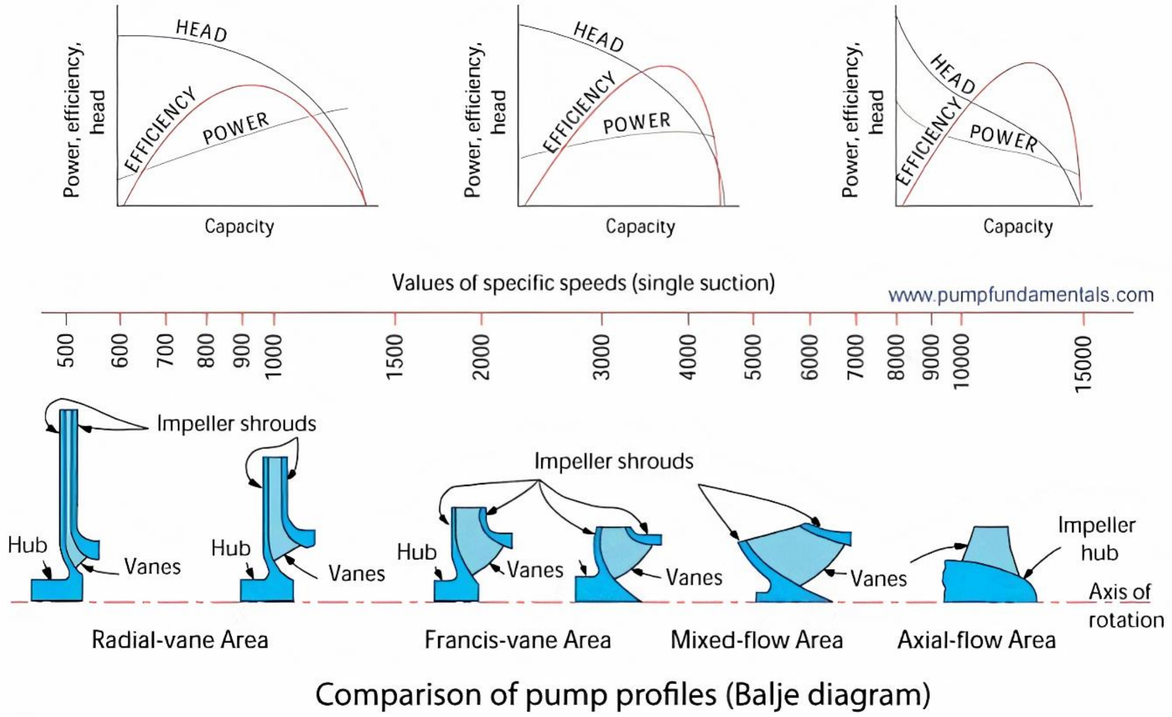

Figure 2: Pump types by specific speed. Source: https://www.pumpfundamentals.com/centrifugal-pump-tips.htm[2]

Two pumps are called hydraulically similar, if both the hydraulic shapes and the fluid dynamics are similar. I.e., there exists a real coefficient λ > 0 such that *the hydraulic shape of the second pump is* λ*-scaled geometry of the first pump (including the surface roughness).* If we follow the so-called *Affinity laws* $(Q_\lambda = \lambda^3 \cdot Q,\ H_\lambda = \lambda^2 \cdot Q)$, we get a formula:

$$n_{s,\lambda} = n \cdot \frac{\sqrt{\lambda^3 \cdot Q_{OPT}}}{(\lambda^2 \cdot H_{OPT})^{0.75}} = n_s$$

This means that the specific speed remains the same when scaling the pump. Of course, the conditions of hydraulic similarity are not met rigorously in real life. Obviously, gravity or speed of sound remain the same. But, for practical applications this can be safely ignored. This is useful for both so-called *model testing* (i.e., testing a *scaled-down* version of a large production pump) and also for the methods of hydraulic design. The reason is that it is not necessary to develop a new design for every pump size. Instead, it is sufficient to create a design for a given *specific speed,* and then it can be scaled *down* or *up* in a wide range of sizes. Of course, there are limits to this approach (1), as for large values of λ, the surface roughness or mechanical losses of the pump start playing an increasingly important role. Empirically, the *hydraulic similarity* between two pumps is considered reliable for the values of the scaling coefficient $\lambda < 10$.

[2] In this source, a different evaluation of *specific speed*, based on the *imperial system of units* (*ft*, *gpm*). The conversion rate is $n_{sI} = 51.65 \times n_s$.

2.2. Hydraulic design

The whole process of a pump design, production and installation (on site) has multiple steps – hydraulic design, construction, production and technological limits consideration, experimental performance verification etc. This work only deals with the (already very complex) problematics of the hydraulic design. By *hydraulic design* we mean designing the hydraulic (wetted) shapes of the pump. The pump development starts by the hydraulic design, then construction and technology considerations follow. All these stages need to be taken into account, but due to excessive complexity it is commonly approached as an iterative process. As the fluid flow is a very complex phenomena, no simple direct analytical method for *optimal* design is known. Instead, multiple "hybrid" (semi-analytical) design theories, each suitable or recommended for a certain interval of specific speeds, have been proven for the impeller design. "Hybrid" means that these methods combine exact analytical formulas with "engineering" approximations, based on experience and simplifications. The theory originates from Euler's pump equation and velocity triangles (between the circumferential and relative velocity components). The good practice is to approach the impeller design process as a sequence of steps:

1. Deciding the *Main dimensions*, i.e., the inflow and outflow dimensions. This is mostly dictated by the required performance parameters (flow rate and head).
2. Meridional shape of the impeller, i.e., the transition from inflow to outflow part. The goal is to distribute the fluid momentum change in the "optimal" way.
3. Blades – number of blades, relative position in the passage and leading and trailing angles. The blades are shaped at multiple *camber lines*, in case of an impeller there are usually at least three – hub, midline and shroud.

If the stator part contains blades (i.e., for example in a case of axial diffuser), the design methods are similar to the impeller. For different parts (volute, suction), different design methods exist. Generally, the designer tries to minimize the energy dissipation (= efficiency loss), caused by the whirls. I.e., from the geometrical perspective, the energy conversion needs to be as smooth as possible. When flow rate changes, the velocity triangle changes, too. Thus, any pump can only display good performance in a limited range of flow rates.

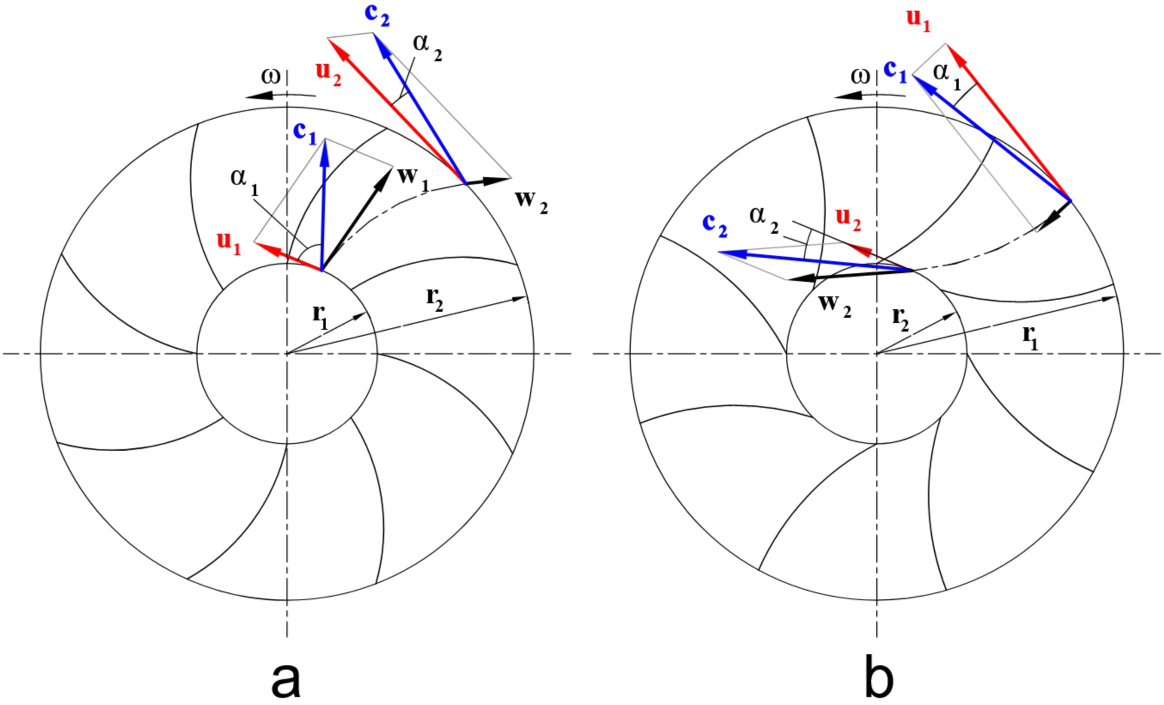

Figure 3: Visualisation of impeller velocity triangle. Source: By Kaboldy - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=9581391

All the design theories are very well elaborated and described in literature, such as (2) or (3). During decades (the modern hydraulic theory is already over 100 years old (4)), many experimentally-verified examples have been gathered and used for experience-based corrections of predictions provided by the hydraulic theories. These corrections play a significant role, as the hydraulic theory does not consider the three-dimensionality of the flow (and other physical phenomena, such as cavitation). As a result, for one particular flow-rate, it is typically possible to design a highly-efficient pump – very quickly and using solely analytical tools and experience. However, vast majority of pumps is not operated at a single point (flow rate). Instead, a whole working range has to be considered. As in the practical applications the pumps speed (rpm) remains fixed (and the same obviously holds for the shape), the velocity triangles change with the flow rate. As a consequence, the flow becomes more turbulent and "three-dimensional" when the is not working at its design point. Under such conditions, prediction accuracy of the analytical methods diminishes. This all means that pump design is a very complex process, that involves balancing multiple conflicting objectives and utilize many "layers" of information – from fast analytical methods to demanding and costly numerical simulations and experimental measurements. Here are some examples what needs to be considered by the designer:

- Peak efficiency (at the design point)
- Efficiency in the whole working range
- Head at the design point
- $NPSH_3$ (suction abilities of the pump)
- Technical limits (minimal thickness of the blades, manufacturability, dimension limits specified by customer, …)

And the tools available to the designer for estimating / evaluating the design performance:

- Analytical methods based on "one-dimensional" flow properties
- Engineering approximations and corrections based on previously measured designs
- Numerical simulations
- Experimental measurements, typically performed on model (scaled down) pump.

The analytical methods, together with the corrections, are very fast and efficient. But, as mentioned before, they have limited accuracy for non-optimal flow rates. The accuracy in non-standard situations (For example, limits on pump dimensions that do not allow for the "best practice" main dimensions, etc.), where the assumptions of the methods cannot be quite met for some reasons, can also be of a concern. In such cases, the numerical simulations and experimental measurements offers higher accuracy, but at higher cost. The general rule is that the designer tries to obtain as much information as possible from the cheapest method. The more demanding numerical simulations are used as verification and correction tool. And the most demanding and expensive experimental measurements are only used for the final verification of the design performance. The graphical visualisation of the workflow can be seen in Figure 4.

The major advantage of the analytical methods is that following the "good practice" produces well-tested designs, which will most likely prove to have properties desired for the later stages (construction, production, …) of the design process. This is an important factor, as many of the pump qualities (mechanical properties, vibrations, $NPSH_3$, …) are difficult (or even impossible) to obtain during the early stages of the (design) process. And the later there are changes, the more expensive they become. This also limits the contribution of the numerical simulations. While they are a powerful tool, it would be extremely difficult to model all relevant pump characteristics. This means that numerical simulations are very useful as an extension of the analytical methods (used for "tuning" the baseline design), but cannot be used as a sole tool. The same is true for the experimental measurements – the price and time demands are very high.

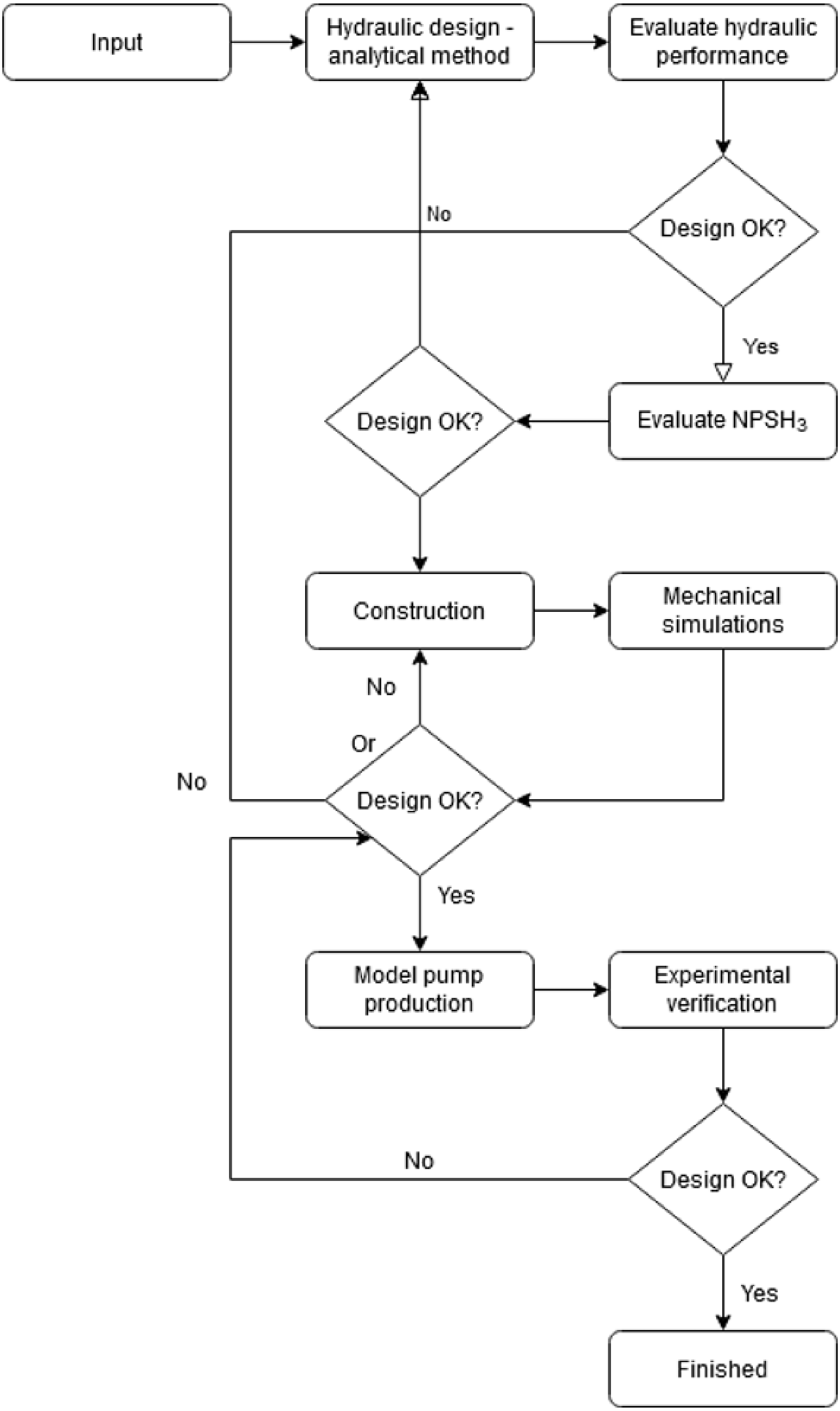

Figure 4: Scheme of hydraulic design

2.3. Hydraulic design and optimization

As was described in the previous text, the "good practice" approach to hydraulic design is, in principle, a many-parameter, multi-objective (and "multi-level"[3]) optimization. The pump geometry is always described by (a finite number) N parameters, and we also have a finite number M of objectives. Thus, the problem of hydraulic design can be viewed as:

$$minimize\ J: \Omega \subset R^N \rightarrow R^M$$

where Ω is a (bounded) set of N parameters describing the pump geometry, and J is an evaluation of pump performance. In the "classical", i.e., expert approach to the hydraulic design, this optimization is not performed in a strictly mathematical sense. Instead, it is combined with human expertise, where many of the objectives are not evaluated solely by numbers. Nonetheless, it is possible to perform a "true" optimization, but it brings considerable challenges:

1. It is necessary to fully automate the evaluation of the objectives. I.e., geometry creation, mesh generation, computational model assembly, running simulations and the post-processing. While this is no problem in theory, in real applications it is not an easy task.
2. In many cases, exact formulation of the objectives can be challenging. Human experts use many ways how to measure or estimate the quality of the design. In some cases, the evaluation is based on things like visual impression ("smoothness" of the hydraulic shapes) or experience with previous designs. It is often unclear, how to process such steps algorithmically, or how to assign an exact numerical value to such decisions.
3. Numerical simulations are computationally very demanding. Having a high-accuracy, multi-physics model is not a viable option for an optimization run. Instead, compromises and simplification are necessary. This, on the other hand, further (together with 2) limits the set of objectives that can be actually used for the optimization.

In theory, it is possible to make a general parametric model, that can cover all possible pump shapes. And to optimize it just for any physically possible pump performance. This is ensured by the fact that both the parameters and objectives are bounded, and their numbers are finite. However, such optimization would be extremely demanding due to large number of both parameters and objectives, combined with multimodality of the objective function(s). When practical limitations are considered, the optimization becomes not a replacement, but instead an addition to the design process performed by human experts. I.e., it is usually extending the earlier steps (as mentioned in the workflow description in the previous chapter) of hydraulic design process. This means that the initial "baseline" design is still performed by a hydraulic expert, and the consecutive optimization uses this design as a starting point. This approach uses the available information from the analytical methods and human expertise, and thus helps making the optimization easier – increasing chance of success and saving time and considerable amount of computational resources. And, since the fidelity of the numerical modelling is limited in practice, it lowers the chances that the optimization arrives at an impractical design due to not having complete information about some aspects of the pump performance.

In the next chapter, the most common optimization methods and approaches to the shape optimization, used in pump hydraulic design, are described.

[3] By "multi-level" it is meant that hydraulic experts work with different types of information – fast analytical formulas, computationally expensive simulations, experience-based judgments of the design quality etc. This means that part of the design process is also a proper utilization of these "levels", something, that is very difficult to include in an automated optimization.

3. Current approach to hydraulic shapes optimization

There are multiple ways of approaching the hydraulic design as an optimization problem. One important aspect of the optimization is how the parametric model is created. One option is to create a “complete” parametric model, i.e., the geometry is completely defined by the input parameters. For hydraulic shapes, the “complete” parametric model is a frequently used option. It is very convenient, as there is a relatively standardized set of rules, how to parametrically create a pump design, that is commonly used by the hydraulic experts. As pumps are axisymmetric, it is based on meridional shapes, blades position, angles and thickness.

Another option is to use the parameters to express a difference from a baseline design. This is typically realized by some kind of parameters-driven shape morphing technique, such as *Free-Form Deformation* (5), mesh deformation techniques based on *Radial Basis Function* interpolation (6) or *Hicks-Henne bump functions* (7), (8). These deformation techniques allow for very flexible control of the shape. It is also very easy to set the number of parameters, so it is possible to choose the details and difficulty of the optimization – something the “complete” parametric models do not allow. These methods are commonly used for wings optimization.

When comparing these two approaches, both have strengths and weaknesses. The first option (the “complete” model) is often relatively straightforward, as in engineering practice, CAE is a standard part of the workflow nowadays. This means that in many cases the parametric model already exists, and if automation of it is possible, then the optimization can directly follow. In such case, the values of the geometry parameters can also be easily interpreted by the experts. The disadvantage is, that if the parametric model is not available, then it can be difficult to create. The deformation techniques, on the other hand, are easier to apply on any geometry. However, sometimes it can be difficult necessary geometry conditions (such as rotational symmetry) are met after the deformation. And when deforming the computational mesh, the quality can suffer and limit the shape variations. Poor mesh quality can also lead to a phenomenon where the optimization actually searches for good mesh instead of good objective values (9).

The “complete” parametric model method was the choice for all optimization cases presented in this work. Its main advantage is that the parameters obtained by an optimization can be easily understood by the hydraulic experts. Also, the “good practice” approach to the geometry creation can easily guarantee that necessary geometry constraints (for example rotational symmetry) are met, as it is inherently included in the geometry generation. And, as mentioned before, it is suitable for wide range of pump designs. Something, that cannot be easily accomplished with the deformations.

Yet another option is to obtain the geometry via some kind of geometry optimization. In such case, the parameters used by the optimization of hydraulic shape are not used directly for geometry creation. Instead, they serve as an input for yet another geometry optimization. This last approach can be combined with the first two. The main advantage is that the geometry optimization can be a very complex, with large number of parameters. And thanks to this (fast) geometry optimization, fewer parameters are needed for the (slow) hydraulic optimization. However, this only makes sense if the geometry creation cannot be simplified in other way. Another big problem is, that the geometry optimization can produce the same (or very similar) results for multiple input parameters. I.e., the input parameters can be different, but the optimization will still converge to the same results. In such case, the hydraulic optimization can have troubles with sampling the objective space. On the other hand, it can help to improve the hydraulic shapes fast. In pump design, smooth shapes are desirable. But the optimization based on noisy and computationally expensive simulations can hardly converge to a smooth shape. But, the geometry optimization can manage this.

Another classification is based on the type of the optimization routine:

1. **Gradient-based methods**. These are the "classical" methods, such as *gradient descent*, *conjugate gradients* or *Newton's method*. These methods can converge (to a local minimum) very fast, even for large number of parameters. However, they suffer from serious drawbacks when applied to (computationally expensive) simulation-driven engineering problems. First, in many commercial simulation packages, the derivatives (of the objective function) are not available (to the user). The derivatives can be approximated numerically, but this considerably increases number of the expensive evaluations and lowers the accuracy. Second, these are *local* methods, and the objective functions in hydraulic design can be multi-modal. Another problem is, that the objective function evaluation can fail for various reasons, and such methods were not designed with such situation in mind.
2. **Global heuristic methods**. such as genetic algorithms, DE (10), PSO (11), etc (12). These are global methods, usually population-based, mimicking real-world behaviour of some systems. They are typically very robust, capable of dealing with very difficult objective functions. They also require relatively big number of the (computationally expensive) objective function evaluations. This makes their application for hydraulic design problematic, as the computational costs can be too prohibitive. On the other hand, these methods can handle penalization of objective functions relatively well.
3. **Surrogate-assisted optimization (SAO).** In this case, the objective function is replaced by an approximation, usually called a *surrogate* or *response surface*. The surrogate can help in guiding the search and speeding the optimization process. Of course, building and updating the surrogate cost additional computational resources, too. But in a case of computationally expensive objective functions the overhead is negligible. There are many different kinds of surrogates, such as *Radial Basis Functions*, *Kriging* or *Support Vector Machine* A survey of various methods performance is in (13).

When considering optimization based on computationally expensive simulations, multiple things need to be considered. First, to number of objective evaluations is limited. Thus, methods improving the objective faster are preferred. Gradient-based methods can converge really fast, if derivatives are available. But in such case, it is also possible to improve building of the surrogates. Second, the method needs to be able to cope with situations where the evaluation fails. This is another advantage of the population-based methods (with or without surrogates). As a new population is generated in every iteration, the error rate would need to be very high to cause any problems. Generally, various SAO methods are most commonly used for computationally expensive simulations (14). The typical good practice it an iterative approach. The method is start from the initial *sampling* on the given parametric space. Once the samples are evaluated, a response surface is fit to these points. Then new samples are generated (based on the information provided by the response surface) and evaluated, and the response surface is updated. Evaluating an approximated value is much faster than the objective functions and can thus speed up the optimization process. As the objective function and its approximation can generally be multi-modal functions, the search is usually based on the global optimization methods. A comparison between an optimization with and without surrogates can be seen in (15).

In this work, global SAO approach was used. The objective functions defined for hydraulic design are difficult to optimize, and derivative information was not available. More detailed description of these methods, and their practical applications on pumps are described in the following chapters.

4. Case 1 - Pump stator

Objectives: Pump efficiency and circumferential velocity at the outflow, at the optimal flow rate.
Solution: ANSYS Workbench parametric model driven through text script files, Transient Blade Row model. Best design selected from Latin hypercube sampling of the parametric space in second iteration.
Results: Efficiency increased from 3% up to 8% in the working range.

4.1. Introduction

The studied pump was a diagonal one, designed for specific speed $n_s = 240$, with adjustable blades of the impeller and a diagonal diffuser. The "adjustable blades" means the attack angle of the blades can be set within a range of values, and allows for operating the pump in wider range of flow rates. The hydraulic parameters were only mediocre, and as the CFD analysis revealed, this was mainly due to the diffuser. A well-designed diffuser can transfer most of the circumferential velocity (inevitably found at the outlet of the rotating impeller) to the desired forward movement. In our case, the residual circumferential part of the velocity was still significant, even at the optimal flow rate. As a result, the total (combined) velocity of the flow in the outlet parts was increased. Thus, the efficiency was lowered significantly due to the hydraulic losses in the elbow.

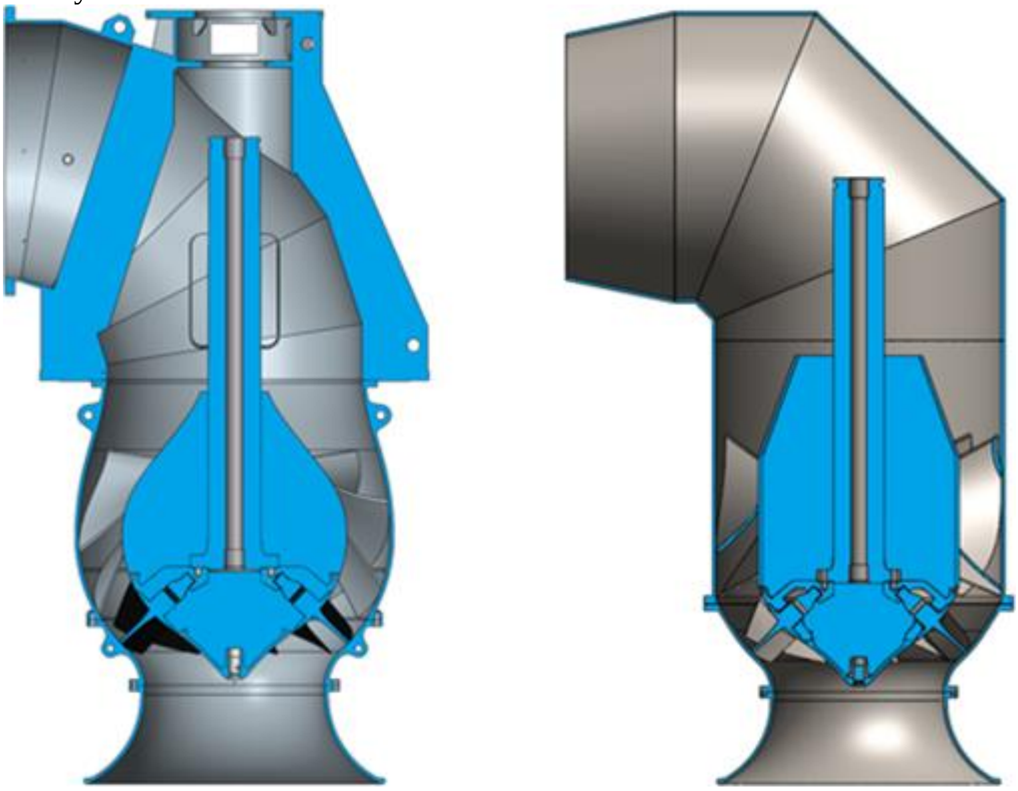

Figure 5: Original pump (left) vs the new design (right)

Because of this, the stator was deemed as the ideal starting point for pump optimization. This decision was based on following reasons:

1. The stator has negligible impact on cavitational performance. Because the cavitation starts at leading edge of the impeller and slowly progresses further into the passage as the NPSH drops, the moment it develops to the stator is far below operational limits of the pump, anyway. Since the numerical simulations of pump cavitational characteristics are both computationally very demanding and difficult to evaluate, optimizing only the stator is much easier.

2. The pump hydraulic performance is mostly decided by the impeller. By lowering the hydraulic loss in the diffuser, both the pump efficiency and head are increased. In such case, there will be an almost ideal linear dependency between efficiency and head (because both parameters are increased by the same mechanism - lowering the hydraulic loss). Thanks to this, only efficiency needs to be considered as optimization objective. This makes formulation of the optimization problem easier.
3. The shape of the stator is simpler and easier to control by parameters than the impeller. For example, stator blades are typically created using two camber lines only, compared to three or more used in case of the impeller. This means lower number of parameters and lower chance for geometry conflicts caused by different angle setting for each streamline.

Apart from improving hydraulic performance, another opportunity for the new design was lowering the manufacturing cost by making the design “simpler”. As revealed by the CFD analyses, the curved design of the original diffuser, based on theoretical assumptions related to optimal flow-direction and cross-section area, did not work well in practice. In fact, the flow was not fully following the geometry shapes. I.e., the complex hydraulic shapes of the original diffuser were not really guiding the flow as intended. For these reasons, considering a shape manufacturable by a process of metal-sheets-bending seemed to be a viable idea. Such method can be considerably cheaper and easier to implement, when compared to the casting used for the original diffuser. In Figure 5, a meridional comparison of the old and new designs can be seen. The longer and wider passage was designed with the intention to enable a more fluid flow, and to prevent whirls at the beginning of the diffuser.

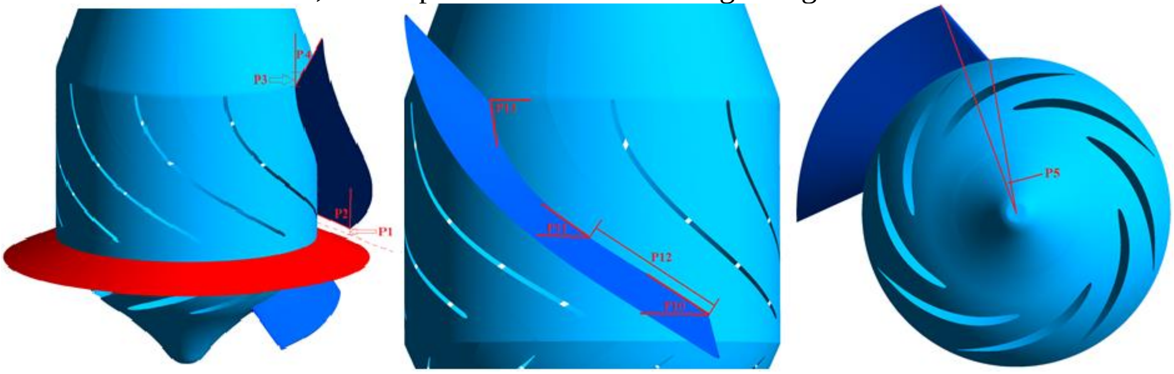

Figure 6: Graphical visualisation of the geometric parameters.

4.2. Parametric model and automation

ANSYS Workbench was used for creation of the computational models. Parametric diffuser model in DesignModeler was connected to automated structured (hexahedral) mesh generation in TurboGrid. Next, the diffuser mesh was updated in a premade CFX model. The other parts (Inflow, Impeller and Outflow) remained fixed. All the necessary steps (updating the diffuser geometry, mesh and CFX model) were recorded as a Workbench script. Next, the numerical values of the considered parameters were replaced by keywords (*par_1*, *par_2*, …) and the script file was used as a template. With Excel macros, a set of Workbench replay files could be easily created, based on a table containing a list of names and appropriate parameters. To avoid unnecessary complexity, only selected parameters were considered – basically blade position, angles and the so-called *sweep angle* (the right part of Figure 6).

In total, 13 geometric parameters were considered. The baseline design followed the “standard” methods for hydraulic design, and the parameter ranges were determined by an offset from these initial values. The parameters are described in Table 1, a graphical visualisation of these parameters is shown in Figure 6 and Figure 7.

Table 1: List of parameters

Parameter	*Description*
1	*Shroud-position of the leading edge of the blade. Defined by a distance to rotor-stator interface.*
2	*Angle between the shroud and the leading edge of the blade. This way the hub-position of the leading edge is also prescribed.*
3	*Hub-position of the trailing edge. Defined by the distance to stator-outflow interface.*
4	*Angle between the hub and the trailing edge.*
5	*Sweep angle of the leading edge. I.e., the angle between the hub-position and shroud-position of the leading edge, as seen from the axis-of-rotation direction.*
6	*Beta angle (of the blade) – shroud, leading edge*
7	*Beta angle – shroud, passage*
8	*Beta angle – shroud, relative position of the passage point*
9	*Beta angle – shroud, trailing edge*
10	*Beta angle – hub, leading edge*
11	*Beta angle – hub, passage*
12	*Beta angle – hub, relative position of the passage point*
13	*Beta angle – hub, trailing edge*

Since the intention was to only consider the design point Q_{OPT}, it was possible to use the so-called Transient Blade Row (TBR) method (16), (17). In this method, only one passage of the impeller and diffuser was considered. TBR can only be used on axisymmetric cases, and the elbow in the outflow part violates this condition. However, at Q_{OPT}, the elbow can be safely ignored and replaced by a straight pipe. Such change has an effect on the absolute values of head and efficiency. However, the general practical experience is that the relative comparison between various pump design remains very similar in the end. For optimization purposes, only the relative values matter. In case of non-optimal flow rates, this is generally not true. TBR is unable to capture the lower-frequency instabilities present in the non-optimal flow rates. But at the optimum, it provides accurate values of the efficiency.

Such approach (considering only Q_{OPT} efficiency as the optimization objective) can be risky, as improving the peak efficiency does not necessarily grant an improvement in the whole working range. Still, as the TBR model meant approximately ten times faster simulation in our case, it was assessed as the preferred option. The resulting mesh size was ca. 0.25 million of nodes (compared to 1.7 million for the "full" model), with simulation time below 2 hours. The comparison between the "full" and TBR models can be seen in Figure 8. For similar reasons, only one adjustment of the impeller blades was considered. Numerical simulations of multiple scenarios would increase the computational demands dramatically, and based on previous experience, it was expected that results obtained for one scenario are applicable for other settings, too.

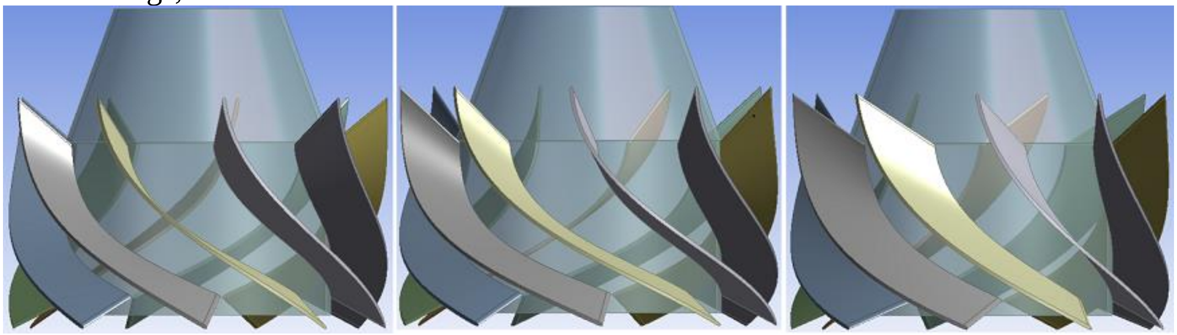

Figure 7: Examples of the stator geometry for various parameters settings

The whole analysis was performed as fully transient. Timestep was chosen with respect to count of impeller and diffuser blades. To maintain reasonable simulation accuracy, one timestep should represent rotation from 1° to 4°. In our case (8 blades on impeller, 9 blades on diffuser) $8 \cdot 9 \cdot 2 = 144$ timesteps per a rotation meet these criteria. One timestep represents 2.5° and has this value:

$$(1)\ \Delta t = \frac{1}{i \cdot n \cdot p} = \frac{1 \cdot 60}{18 \cdot 294 \cdot 8} = 0.00141723(s)$$

with *p* being the impeller blades count, *i* being the number of timesteps per passage and *n* (rpm) the rotor rotation speed (in *rpm*). As a turbulence model, SST (Shear Stress Transport) *k*-Ω model has been chosen. In general, *SST k*-Ω is the recommended choice for modelling fluids in geometries containing rotating parts.

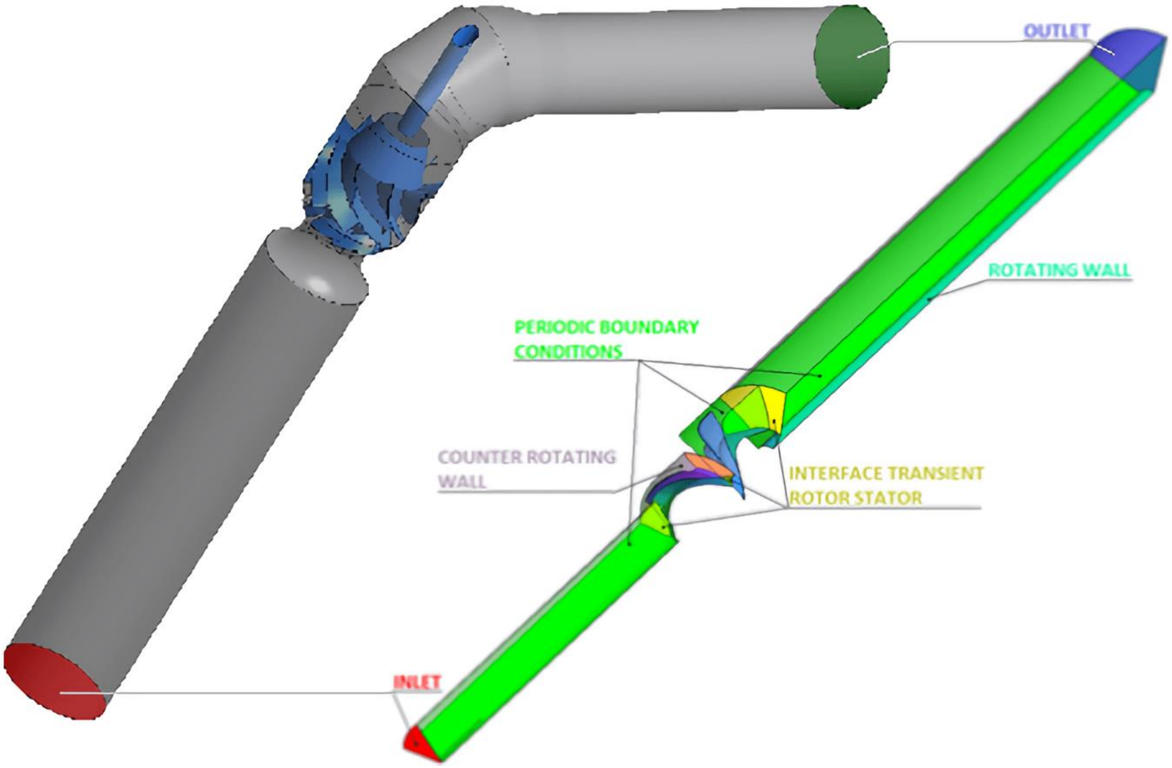
OUTLET
ROTATING WALL
PERIODIC BOUNDARY CONDITIONS
COUNTER ROTATING WALL
INTERFACE TRANSIENT ROTOR STATOR
INLET

Figure 8: Full (left) vs TBR (right) CFD model

4.3. Optimization and results

Using ANSYS DesignXplorer tool, an initial sampling for the hypercube (of the parametric space) was generated. In the first step, only 11 parameters were considered. Parameters 8 and 12 remained fixed at value of 0.35. The sampling had 151 items, of these 147 were successfully created. Next, for these 147 samples, the CFD simulations were performed. The simulations were run on a dual-socket machine, running in parallel as 2 x 8 cores. The total machine time was ca. 3000 CPU-hours, i.e., ca. 8 days of a real time. For each sample, two objectives were evaluated – efficiency of the pump and averaged circumferential velocity at the inflow part of the pump. Since the high residual circumferential velocity was one of the reasons for the low efficiency of the old design, it was also considered for the optimization process and analysed.

From all the computed designs, one seemed to be particularly promising. After an evaluation and analysis of the results, the parameters bounds were limited to focus more on the area close to the best sample of the first run. The comparison of the parameter ranges can be seen in Table 2. Then, a second sampling was created for this hypercube, with all 13 parameters (including P8 and P12) enabled. There were 105 samples in total, and only 77 of these were successfully generated and evaluated.

Table 2: Comparison of parameter bounds – 1^{st} vs 2^{nd} sampling.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
1^{st} sampling	1	45	5	45	-20	-75	-60	0.35	-3	-70	-60	0.35	-3
	15	70	100	90	-5	-45	-30	0.35	12	-40	-30	0.35	12
2^{nd} sampling	1	55	5	30	-15	-70	-65	0.25	-10	-70	-65	0.25	-10
	10	67	15	55	-5	-55	-50	0.5	3	-55	-50	0.5	3

Once again, all the CFD simulations were run in parallel on the dual-socket machine. The efficiency results for both runs can be seen in Figure 9. From the new samples, the best design was selected. It displayed both excellent efficiency and minimal residual circumferential velocity at Q_{OPT}. After minor modifications (mainly by rounding the values of the geometric parameters), it was used as the final design. For this design, the full CFD model (all passages, outflow elbow) was created, and complete performance characteristics were evaluated for three different settings (of attack angle) of the impeller blades – 0, 6 and 9 degrees. In total, 21 performance points (3 adjustments, 7 flow rates) were computed. The results can be seen in Figure 10 and Figure 11.

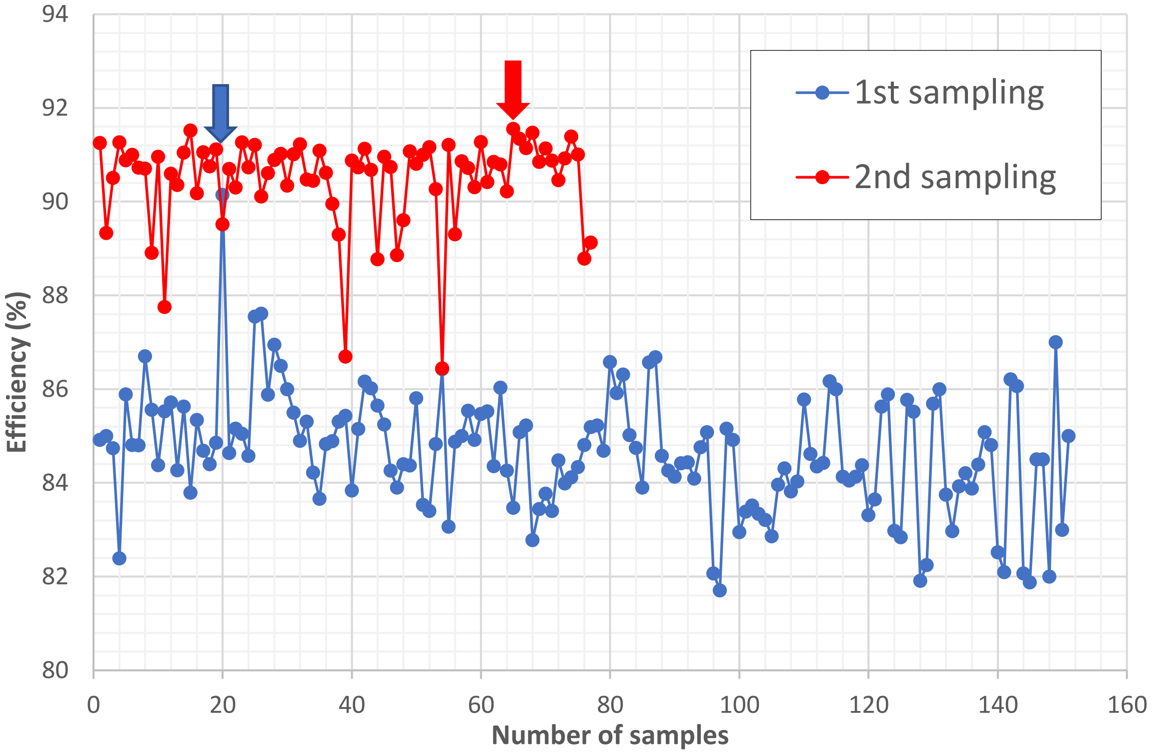

Figure 9: CFD results (efficiency) for the 1st and 2nd sampling. The arrows show the selected (s the best) designs for each sampling.

As can be seen, the optimized design clearly dominates the old one. In the whole working range, no matter what attack angle is set for the impeller blades, the improvement in efficiency ranges from 3% up to 8%. For a high-performance pump, this is a very significant improvement. The real product can vary in size a bit, as it can be scaled down or up in accordance with the laws of hydraulic similarity. The maximum power can exceed 5 MW in its largest form, and for such power consumption, every single per cent of efficiency matters when the pump is operated for a long period of time. The higher efficiency can help to save the electricity costs, and possibly save the cost for the electro engine. The head has improved by similar margin. This is only natural, as the impeller remained the same. The $NPSH_3$ characteristics were not simulated. As mentioned before, the diffuser does not have a significant impact on cavitation.

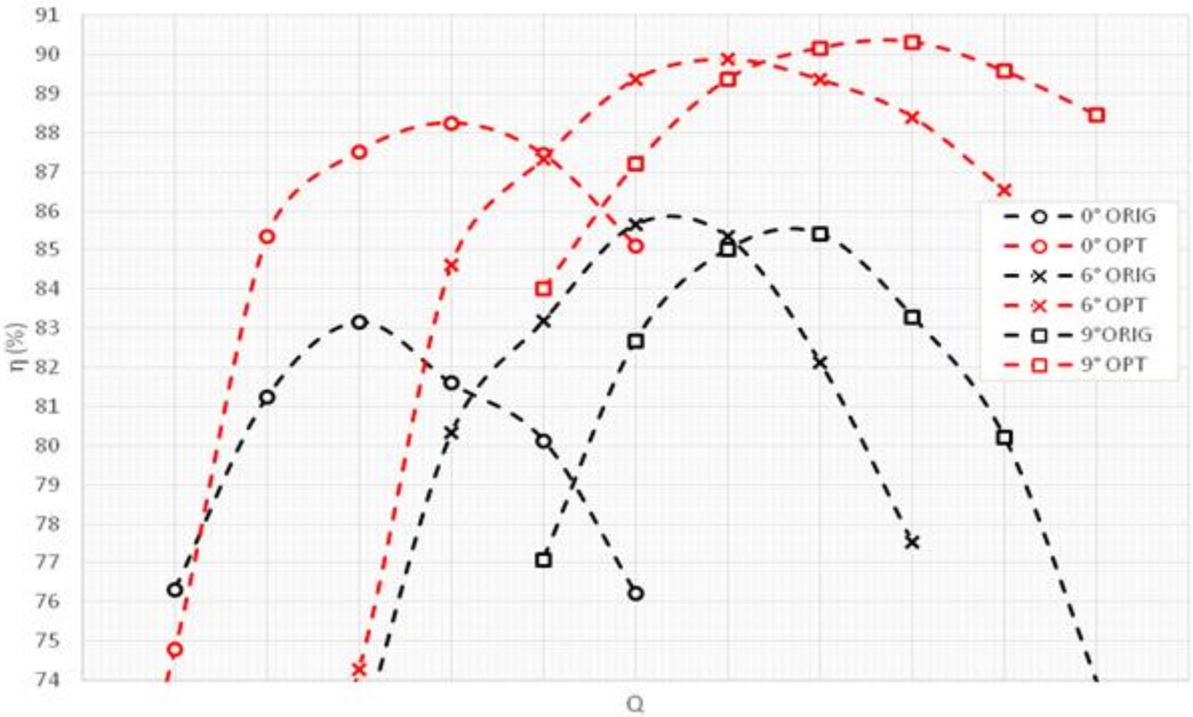

Figure 10: Efficiency for different blade adjustments. The original (black) vs optimized (red) design.

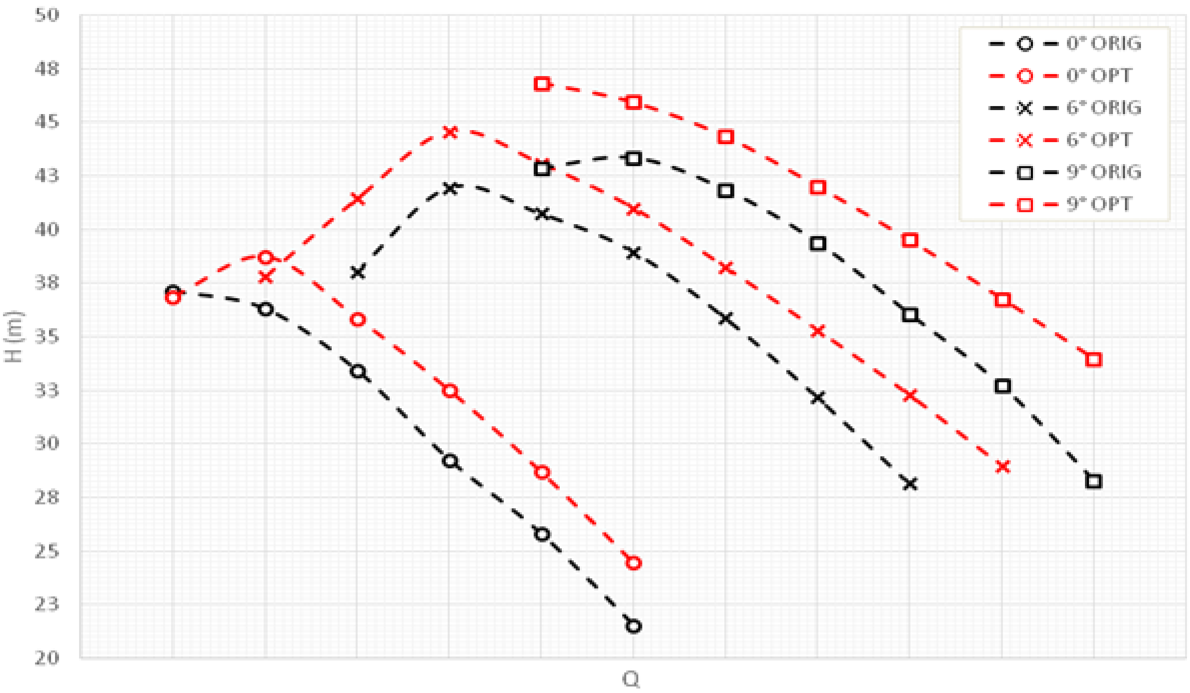

Figure 11: Head for different blade settings. The original (black) vs optimized (red) design

In Figure 12 and Figure 13, a comparison of residual circumferential velocity and backflow areas between the old and optimized design can be seen. There is an evident and significant difference. The wider and longer passage, together with the optimized blades, helped to reduce the circumferential velocity. The lower dissipated energy translates into higher overall efficiency of the newly designed pump. The new model also worked better despite the fact it was meant for the *sheet-metal bending* technology, which limited the parametrization options of the geometry.

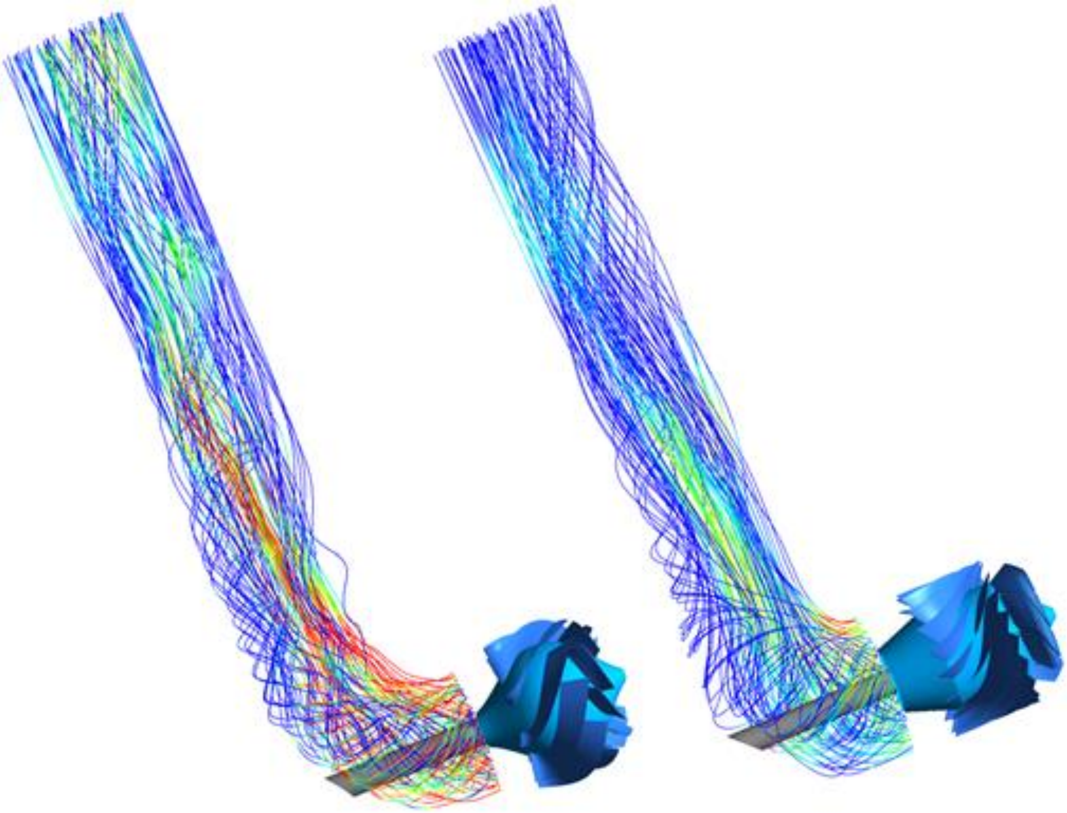

Figure 12: Circumferential velocity at the outflow for the original (left) and optimized (right) design. Excessive values are displayed in red colour.

4.4. Summary and conclusion

The automated parametric model was successfully assembled and used as a part of CFD-driven hydraulic design of a pump diffuser. The optimization was only performed with a simplified TBR model at the design point Q_{OPT}, but the outcome was very good nonetheless. The newly designed and optimized diffuser display performance superior to the original one – ranging from 3 % to 8 % in majority of the working range. It should be noted, however, that randomness can play certain role in the process, as the parametric space and initial sampling can influence the results. This can be also observed here, as in the first initial sampling, only one sample helped guiding the search and further narrowing the parametric space to the promising area. Overall, the selected approach has shown promising results and potential for the future tasks of hydraulic development.

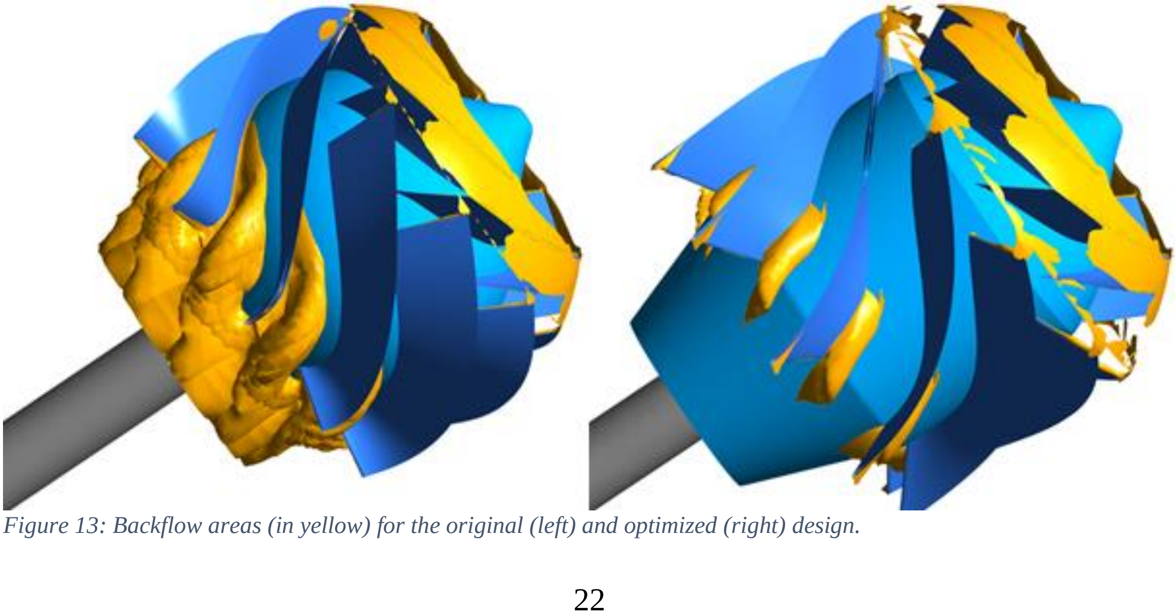

Figure 13: Backflow areas (in yellow) for the original (left) and optimized (right) design.

5. Case 2 - Pump suction

Objectives: Pressure loss and uniformity of velocity profile at the output of the suction part (at Q_{OPT}).
Solution: ANSYS Workbench parametric model driven through text script files, steady-state and transient simulations, Stochastic RBF optimization method
Results: Efficiency increased by 0.5%, $NPSH_3$ by 20%.

5.1. Introduction

In this case, a radial pump of *specific speed* $n_s = 135$ was considered. In order to improve the suction ability ($NPSH_3$ characteristic) of the pump for a specific application, an inducer[4] was developed. CFD analyses of the pump performance revealed, however, a problem at the inflow part of the pump. For optimal performance, the flow that comes into the rotor (i.e., the inducer in this case) should be perpendicular to the entrance and uniform. In this case, however, the CFD analyses revealed significant variations in the inflow profile, which is supposed to have detrimental effect of the pump performance. The visualisation of the velocity profile at the interface between the suction and the impeller is shown in Figure 16. In such case, the velocity profile at the leading edges (of the blades) is not expected, and thus the impeller is not working in the regime it was designed for. This means that the blade angles are not optimally selected for the direction of the fluid.

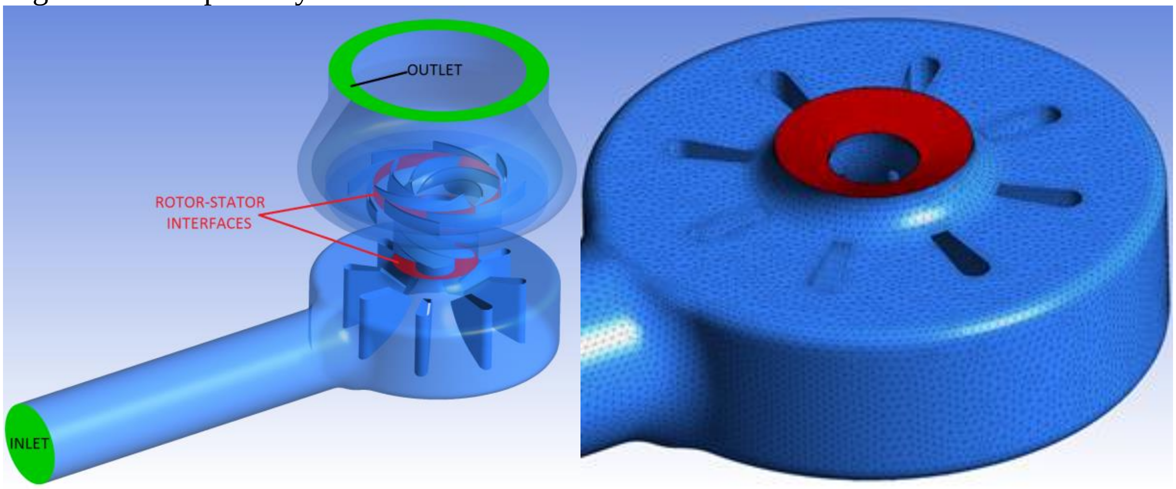

Figure 14: The pump CFD model and a detail of the suction geometry and mesh.

Based on the data, it was decided that optimizing the suction shape is a good opportunity for improving the pump performance. The expectation was that such optimization can be based on steady-state simulations of the suction only, significantly limiting the computational expense and efforts invested into the automation of the numerical modelling. The plan was to further develop the automation and connect it with a suitable optimization method. When compared to the impeller or diffuser, the suction of the shape is less complicated and easier to change via geometric parameters. So, this task was decided to be the optimal starting point for testing a fully-automated parametric model, connected to a suitable optimization code. As the computational cost of one steady-state CFD simulation of the suction part is not very taxing, it can be much easier to test geometry and mesh creation, and to control the simulations. Evaluating steady-state computation is easier, too, as it converts to a single number. For transient simulation, averages over a period of time need to be considered.

[4] An inducer is added in front of the impeller and shares the same shaft. Its blades are designed to generate static pressure and help feeding the impeller to help to delay the cavitation occurring there. In exchange, the efficiency of the pump is lowered by a few per cents.

5.2. Parametric model and automation

The parametric creation of the suction was done using ANSYS Workbench and DesignModeler. The approach to the hydraulic design and choice of parameters were mostly based on recommendations in (2). In total, there were 18 geometry parameters, which are listed in Table 3. Visualisation of selected parameters can be seen in Figure 15. The parameters and geometry model were selected as a compromise between a full control over the suction shape and the complexity of the optimization. There are also technical difficulties when creating such parametric model, as the freedom for creating a 3D shapes is more limited in the DesignModeler than it should be in theory.

Table 3: Parameters description

Parameter	*Description*
1	*Radius of the inner wall.*
2	*Radius of the outer wall.*
3	*Rounding radius of the diffusion channel.*
4	*Radius of the diffusion channel.*
5	*Width of the diffusion channel.*
6	*Height of the bottom part (of the suction).*
7	*Height of the top part.*
8	*Meridional width of the bottom part.*
9	*Meridional width of the top part.*
10	*Meridional width of the middle part.*
11	*Diffuser vane – leading edge - radius at the top*
12	*Diffuser vane – trailing edge – radius at the top*
13	*Diffuser vane – leading edge – position at the top*
14	*Diffuser vane – trailing edge – position at the top*
15	*Diffuser vane – leading edge - radius at the bottom*
16	*Diffuser vane – leading edge – radius at the bottom*
17	*Diffuser vane – leading edge – position at the bottom*
18	*Diffuser vane – trailing edge – position at the bottom*

Considering the shape of the suction, the computational mesh was created as unstructured, i.e., tetra + prism for the boundary layers. Creating a block structure would be overly complicated in this case. The numerical modelling was fully automated, using ANSYS Workbench, Linux shell and Python scripts and codes.

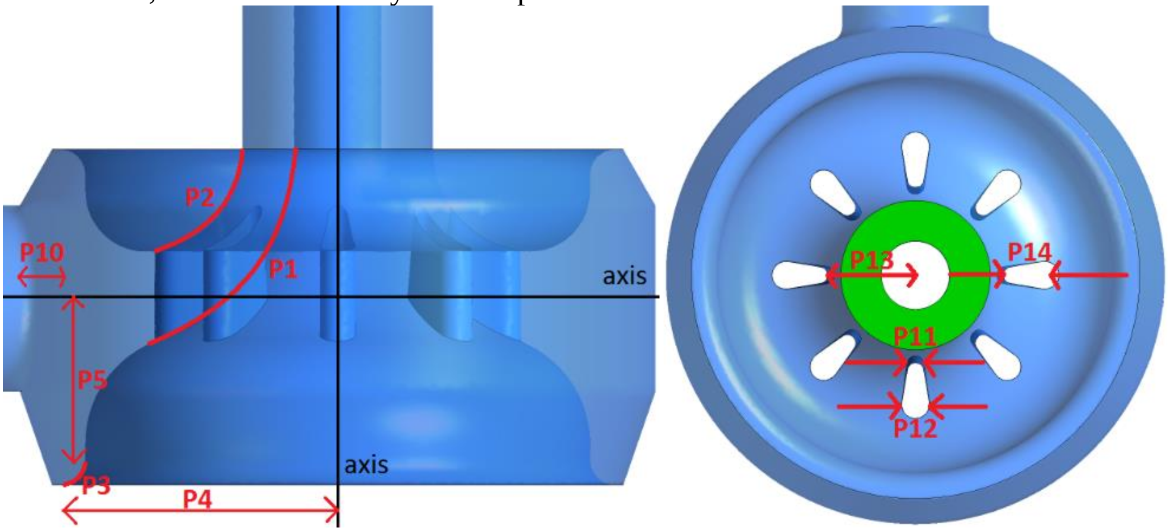

Figure 15: Visualization of selected parameters

First, the update process in Workbench (DesignModeler, Meshing and CFX/Pre updates) was recorded as a script. By replacing the file name and parameter values by keywords, a source template for the automation was then created. The workflow of the automation is as follows:

1. The Python routine accepts csv file with parameter values as an input.
2. The Workbench template is loaded, keywords are replaced by the actual values, and a new script is saved – named after the input csv file.
3. The Workbench script is executed in command line. As a result, a CFX input file is created.
4. For pre-set flow rates (and possibly other solver settings defined in a premade ccl files), the CFX input file is solved in batch mode.
5. For each result file, monitored variables (efficiency, head, …) are extracted and stored in a csv file (with appropriate name).
6. The csv files are processed by a Python script, and efficiencies (for each result file of a particular design) values are saved into a csv file. This csv file then acts as an input for an optimization routine.

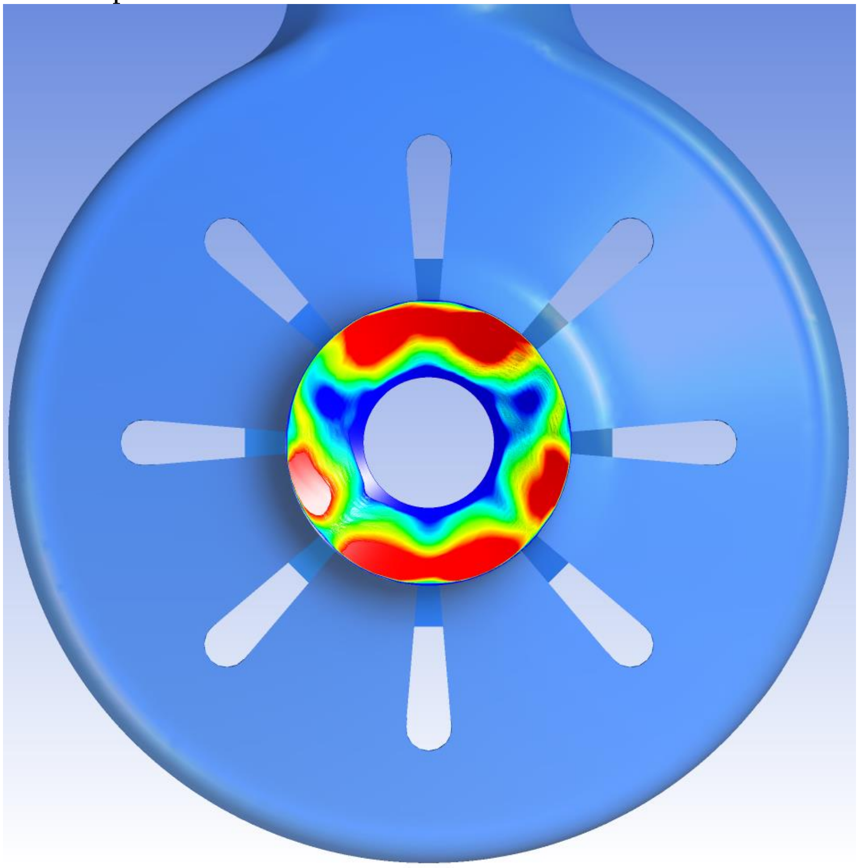

Figure 16: Velocity profile at the inlet to the inducer. Red or blue colour shows greater than 10% difference from the average value.

The step 4 (the solver) is typically performed on an HPC cluster via a PBS scheduler. I.e., the user does not have a direct control over the run. Instead, the task is sent to a queue, and the scheduler decided when it is run. Such decision is based on multiple factors – licences availability, computational resources availability, user priority etc. This brings many technical challenges to the automation procedure. Apart from difficulties with data transfers between the “global” shared storage and local computation nodes, the simulation can fail for various reasons (licensing problems etc.). To maximize the utilization of the available resources and to make the simulation “flow” more manageable, the CFX model creation and assembly is separated from the solver run. The scripts work like this:

1. For all *wbjn* files (the Workbench scripts) present on the shared storage, the corresponding jobs are created and send to the queue.
2. Once the “model creation and assembly” job starts, it copies all the necessary data to the assigned local storage and tries to execute Workbench in the batch mode. As the Workbench run can “freeze” frequently (for various technical reasons), special measures needed to be taken. Because the time spent for the generation of the solver file is known quite well (typically between five and ten minutes), the Workbench process is limited to 30 minutes (and killed after that, if necessary). At the end of the script run, a log file is created. Now three situations can occur:
    a. Neither the solver input file nor the log file is created. This means Workbench “froze” and got terminated by the timer. In this case, the job is sent to the queue again.
    b. Only the log file is created. This means the Workbench script was executed successfully, but the geometry or mesh generation failed. In this case, the design is written to a log file on the shared storage as “failed”.
    c. The solver input file is created. In this case for each pre-set solver setting (uploaded at the shared storage), the solver jobs are set to the queue (in parallel).
3. The “solver” job copies the necessary data to the local storage and runs the CFX solver with the appropriate model and settings in batch mode. As no problems with “freezing” were observed during the CFX runs, no measures similar to running Workbench needed to be deployed. One the simulation end (typically in ca. 10 minutes), the results are copied back to the shared storage and an “objectives evaluation” script is run.
4. The evaluation script is created in Python, and process the data extracted from the result file. The outcome is an objective function value stored in a csv file. If all the results for the design (i.e., all result files corresponding to the defined list of solver settings) are available, the scalarized objective is generated and written to a file. The design is also flagged as “finished” in another log file on the shared storage.

With this approach, it is possible to maintain sufficient control over multiple simulations running in parallel, without a central “driving” script. Such script would be technically very difficult to create, as both the user’s control over the scheduler and knowledge about running and finished jobs are very limited. Instead, any optimization code can just check for newly added “finished” jobs in regular intervals.

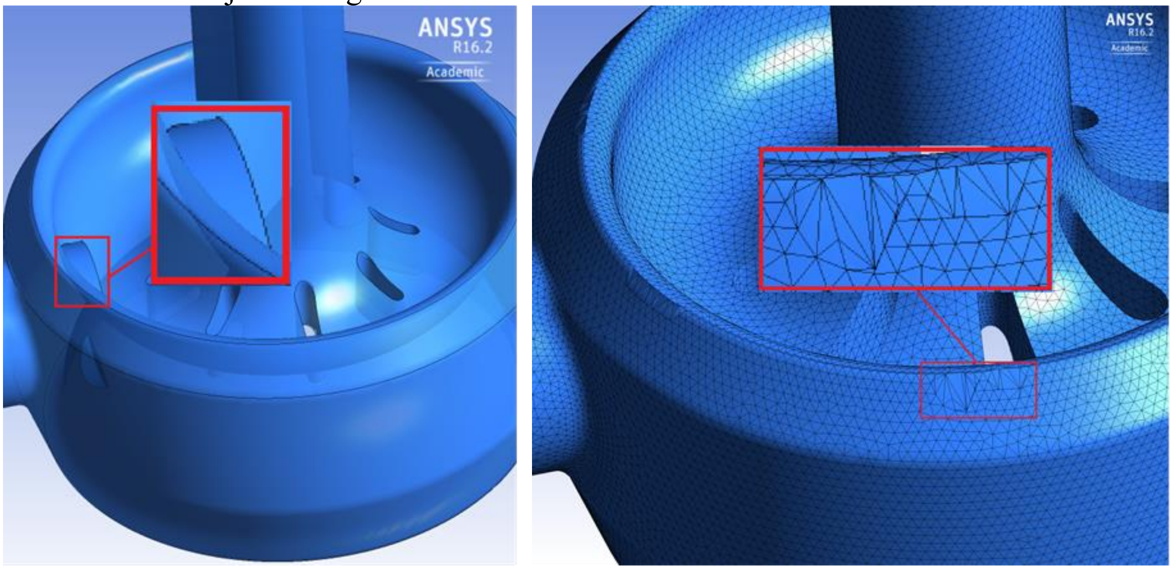

Figure 17: Geometry error (left) and mesh error (right).

5.3. Testing the parametric model

For the start, a relatively large sampling of size 100 was created and used for testing of the parametric geometry. Of these 100 samples, 43 failed due to various geometry or mesh errors. This is a common problem faced when creating the parametric model, as ensuring a consistent geometry generation for all possible combinations of the parameters is often a challenging task.

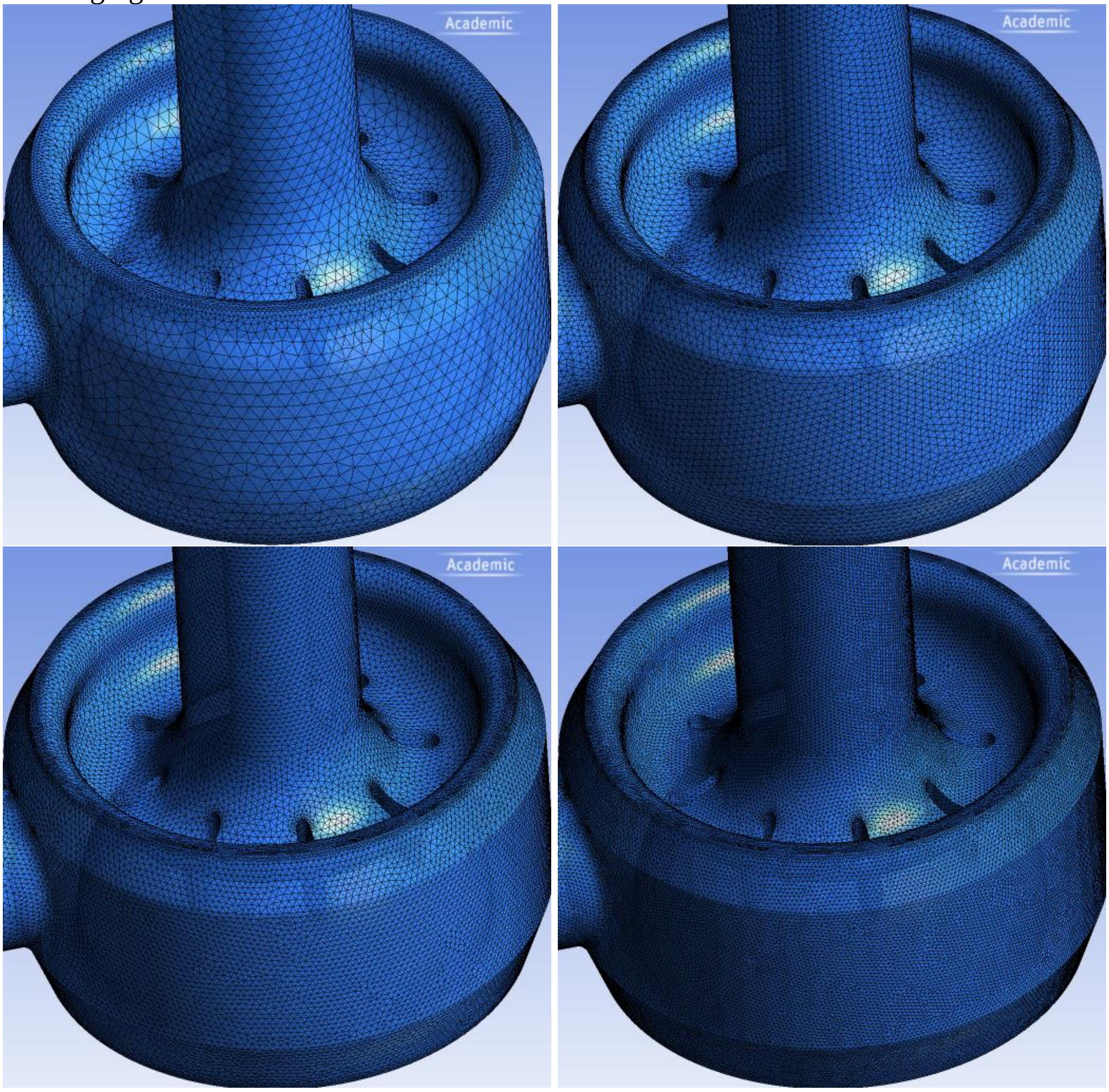

Figure 18: Four mesh sizes used for the mesh dependency tests. Ranging from ca. 100K to 2M nodes.

After improving the geometry creation, 88 samples (out of the 100) were successfully generated. This rate was decided to be acceptable. Next, a smaller sampling, intended for CFD evaluation and testing the parametric space, was created. For the 18 parameters, 38 samples were created and used as an input for the batch processing. For the lack of better information, the default size used in the Stochastic RBF method codes was used. It is defined as *2 * (N + 1)*, where *N* is the number of parameters.

Table 4: Parameter ranges

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
200	130	30	380	220	5	5	2	2	30	5	10	190	250	5	10	90	250
450	200	50	510	300	50	50	25	25	100	25	45	230	280	35	50	220	280

With this setting, 31 models were successfully generated and 7 failed because of geometry errors. These 31 samples were evaluated by CFD at Q_{OPT}. For the optimization, only the suction, together with prolonged inflow and outflow (for better numerical stability), were considered. SST k-ω model was used as the turbulence model. The original intention was to use both total pressure loss (denoted as *H*) and outlet velocity uniformity (denoted as *v*) as optimization objectives.

$$H = \frac{Total\ Pressure_{Inlet} - Total\ Pressure_{Outlet}}{\rho_{Water} \cdot g}$$

$$v = \int_{Outlet} \left| v - v_{avg} \right|$$

Unfortunately, upon more thorough testing the results proved to be significantly mesh and model dependent. In Table 5 and Table 6, the values for each mesh size / solver settings are normalized and coloured according the normalized value.

Table 5: Normalized values of H (left) and v (right) – mesh dependency.

	Mesh size				Mesh size			
Sample	rough	medium	fine	finer	rough	medium	fine	finer
1	1.00	1.00	1.00	1.00	0.45	0.56	0.54	0.65
2	0.76	0.83	0.82	0.87	0.80	0.73	0.70	0.88
3	0.43	0.64	0.55	0.66	0.76	0.57	0.57	0.65
4	0.61	0.70	0.68	0.78	1.00	0.95	0.94	0.87
5	0.68	0.76	0.77	0.75	0.95	0.93	0.70	1.00
6	0.38	0.53	0.37	0.56	0.33	0.58	0.42	0.37
7	0.50	0.59	0.56	0.69	0.84	0.64	0.67	0.82
8	0.40	0.54	0.57	0.63	0.81	0.57	1.00	0.57
9	0.43	0.37	0.45	0.39	0.51	0.30	0.22	0.33
10	0.42	0.67	0.64	0.78	0.34	0.13	0.11	0.07
11	0.41	0.54	0.57	0.62	0.85	0.61	0.40	0.63
12	0.57	0.55	0.54	0.57	0.71	0.43	0.37	0.43
13	0.24	0.25	0.30	0.24	0.31	0.25	0.20	0.20
14	0.17	0.23	0.25	0.18	0.52	0.53	0.25	0.27
15	0.30	0.49	0.46	0.55	0.73	1.00	0.51	0.55
16	0.18	0.20	0.14	0.37	0.36	0.28	0.13	0.03
17	0.19	0.21	0.23	0.16	0.55	0.37	0.29	0.14
18	0.18	0.19	0.24	0.11	0.71	0.40	0.43	0.35
19	0.59	0.44	0.48	0.49	0.37	0.40	0.36	0.35
20	0.17	0.33	0.41	0.40	0.80	0.41	0.26	0.24
21	0.14	0.19	0.23	0.21	0.28	0.12	0.02	0.05
22	0.38	0.42	0.37	0.47	0.31	0.33	0.32	0.36
23	0.14	0.22	0.43	0.25	0.36	0.08	0.14	0.08
24	0.24	0.22	0.13	0.25	0.48	0.50	0.27	0.52
25	0.32	0.37	0.51	0.36	0.00	0.00	0.00	0.22
26	0.43	0.32	0.36	0.31	0.18	0.34	0.14	0.06
27	0.27	0.41	0.43	0.39	0.51	0.27	0.12	0.29
28	0.09	0.16	0.10	0.11	0.54	0.34	0.03	0.00
29	0.13	0.33	0.33	0.35	0.89	0.42	0.15	0.69
30	0.00	0.00	0.05	0.00	0.54	0.53	0.48	0.23
31	0.02	0.06	0.00	0.19	0.32	0.36	0.11	0.00
32	0.14	0.19	0.37	0.20	0.38	0.44	0.07	0.34

The testing was done for four different mesh sizes and four solver settings. The mesh sizes ranged from ca. 100 thousand to 2 million nodes. There were differences in both surface elements sizes and prism layer settings. As there are no moving parts in the suction, the simulation was set as "steady-state", significantly faster than the transient option. The comparison of meshes for one geometry are displayed in Figure 18. Every design in the sampling was then evaluated for these eight scenarios. For an optimization, relative values of the objectives are more important than the absolute numbers. I.e., the statement *"design A is better than design B"* should be independent on the mesh or solver settings.

Table 6: Normalized values of H (left) and v (right) - solver settings dependency.

	Solver settings				Solver settings			
Sample	**v1**	**v2**	**v3**	**v4**	**v1**	**v2**	**v3**	**v4**
1	1.00	1.00	1.00	1.00	0.74	0.89	0.37	0.27
2	0.77	0.79	0.83	0.84	0.37	1.00	0.38	0.31
3	0.60	0.63	0.64	0.68	0.35	0.27	0.39	0.12
4	0.63	0.74	0.70	0.83	0.18	0.79	0.09	0.76
5	0.67	0.68	0.76	0.73	0.59	0.93	0.21	0.78
6	0.46	0.52	0.53	0.58	0.67	0.79	0.38	0.20
7	0.53	0.58	0.59	0.62	0.24	0.03	0.22	0.27
8	0.49	0.51	0.54	0.56	0.02	0.00	0.04	0.08
9	0.31	0.31	0.37	0.32	0.56	0.37	0.71	0.69
10	0.61	0.66	0.67	0.74	0.21	0.31	0.48	0.05
11	0.48	0.53	0.54	0.54	1.00	0.72	0.71	0.39
12	0.50	0.50	0.55	0.58	0.34	0.59	1.00	0.91
13	0.21	0.22	0.25	0.26	0.01	0.00	0.00	0.00
14	0.21	0.26	0.23	0.25	0.47	0.40	0.41	1.00
15	0.41	0.44	0.49	0.47	0.45	0.04	0.33	0.13
16	0.15	0.16	0.20	0.21	0.10	0.43	0.29	0.50
17	0.18	0.18	0.21	0.22	0.11	0.15	0.11	0.12
18	0.15	0.15	0.19	0.19	0.00	0.04	0.00	0.02
19	0.39	0.39	0.44	0.39	0.51	0.33	0.75	0.44
20	0.30	0.31	0.33	0.38	0.19	0.37	0.38	0.20
21	0.18	0.18	0.19	0.22	0.34	0.47	0.25	0.77
22	0.35	0.36	0.42	0.39	0.45	0.43	0.31	0.67
23	0.19	0.20	0.22	0.24	0.07	0.04	0.21	0.12
24	0.19	0.21	0.22	0.25	0.79	0.74	0.44	0.69
25	0.32	0.33	0.37	0.36	0.17	0.20	0.56	0.20
26	0.27	0.28	0.32	0.32	0.29	0.34	0.25	0.33
27	0.35	0.36	0.41	0.40	0.51	0.68	0.65	0.81
28	0.14	0.13	0.16	0.17	0.47	0.10	0.20	0.43
29	0.29	0.29	0.33	0.35	0.18	0.11	0.28	0.23
30	0.00	0.00	0.00	0.00	0.33	0.07	0.10	0.31
31	0.02	0.03	0.06	0.11	0.72	0.18	0.16	0.36

It is clear that the velocity profile depends significantly on the simulation settings and mesh size. The probable cause of this is that the local values of the velocity field are much more prone to errors and inaccuracies than the "averaged over area" head values. As a consequence, *v* was decided to be too unreliable to be used as an optimization objective. Thus, only the pressure loss at the design flow rate was selected as the cost function.

5.4. The optimization method

When selecting an optimization method for a CFD-driven optimization, one has to consider multiple criteria:

1. The numerical simulations are computationally very expensive, this severely limits the maximum number of the objective function evaluations. This means that a method that can improve the objective(s) fast is preferable to a method that can do better, but only after very high number of (objective function) evaluations.
2. It needs to be a derivative-free method. In special cases and with in-house or open source (such as OpenFOAM) codes, it is possible to use the so-called *adjoint method* (18). But generally, especially for commercial codes such as ANSYS CFX, no derivatives are available. Numerical differentiation is not a viable option here, due to accuracy limits of numerical methods and high number of parameters.
3. The objective functions are supposed to be multimodal, i.e., global optimization methods are preferable.
4. Failed simulations / crashes can occur for various reasons. I.e., for some input values, the objective function returns no output. The non-existent values can be “faked”, but this can distort the optimum search. Thus, having a method that can handle such situations easily is preferable.

Point 1 is a problem for classical population-based methods, points 2, 3 and 4 practically rule gradient-based methods (steepest descent, Newton's method, etc.) out. Currently, the so-called Surrogate-Assisted Optimization (SAO) is the preferred option for computationally expensive engineering optimization. In SAO, the approximation of the objective function (surrogate) is used instead of the objective function itself. In every iteration, the optimization is performed on this (computationally cheap) surrogate. Once the new point for evaluation is decided, it goes through the computationally expensive simulations and the surrogate is updated. SAO has been studied extensively in literature, such as (13), (19) or (20).

For the suction optimization, the so-called Stochastic Radial Basis Function (Stochastic RBF) method, described in (21), (22), was selected for this task. There were two main reasons for the choice:

1. According to the testing done by the authors, it is very competitive when maximum number of evaluations is a concern.
2. Matlab codes were freely available at (23), and easy to understand and modify. This was a major advantage, as it is necessary to modify the codes to deal with the specialities of CFD-driven optimization

Stochastic RBF is a single-objective, stochastic SAO method. It works by a following scheme:

1. Generate initial sampling and evaluate the samples.
2. Use already computed samples and fit the response surface.
3. Generate large number (ca. ten thousand) random testing points to cover the response surface uniformly.
4. From these testing points, select *N* new candidate points for evaluation. The selection is a compromise between exploiting local minima of the response surface and exploring areas further away from the already sampled points.
5. Evaluate the candidate points.
6. Repeat (2) until ending criteria are met.

The number *N* can be selected arbitrarily; the authors recommend either 4 or 8 for optimal performance. To deal with the technical difficulties in real-world application for the shape optimization (namely failed samples and heterogenous HPC cluster environment with a scheduler), the original method was modified to work with something that could be called “pipeline”.

1. Generate initial sampling and evaluate the samples.
2. Use already computed samples and fit the response surface.
3. Generate large number (ca. ten thousand) random testing points to cover the response surface uniformly.
4. Select *N* new candidate points for evaluation and add them to the queue.
5. Wait until at least one computation finishes. Then wait for a few more minutes and read all available results. (Due to the nature of the scheduler system, the results often come within very short interval.)
6. Select k new points, where *k = number of samples computed + number of samples failed*.
7. Add the newly selected samples to queue.
8. Repeat (5') until ending criteria are met.

This way, the failed samples are simply ignored, and the "slower" ones are just used once they finish. Because of this, there are no bottlenecks caused by waiting for "stuck" computations. One can assume that such approach has influence on convergence speed, but no thorough testing has been performed with the computationally expensive functions. Obviously, there is no automated solution for a case where all new samples fail. In practice, such situation never occurred, though. Plus, as the method always work with "all data available", it is possible to insert new samples manually. It is thus for example possible to manually "tune" a selected design (an experienced hydraulic expert can often do this) and add it to the already evaluated data.

5.5. Optimization and results

The optimization continued from the results of the initial sampling. The number of newly generated samples in every iteration was set to *N = 4*. The optimization was ended manually once no significant improvement in the objective have been observed for multiple iterations. The results can be seen in Figure 19. In this particular case, the optimization has hardly shown an improvement over the random sampling, performed over the parametric space.

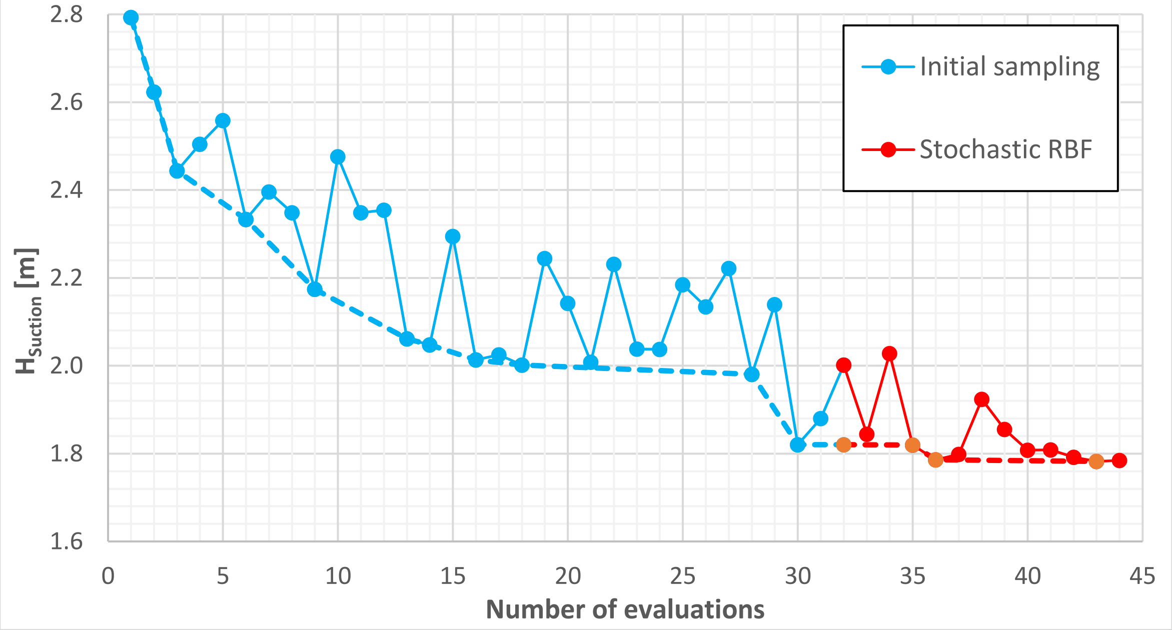

Figure 19: Objective function progress. The bold line connects the best values reached.

Based on the optimization results, the complete CFD model of the pump (suction + inducer + impeller + stator + outflow) was assembled. For comparison, both best samples from the initial sampling and Stochastic RBF were considered. Next, a full set of CFD simulations (performance and *NPSH₃* characteristics) were performed. Comparison to the original design can be seen in Figure 20. The differences in efficiency and head are negligible, basically within the accuracy limits of the numerical simulations. More significant improvement, up to almost 8 %, can be observed for the suction ability. While the improvement in hydraulic losses is negligible in the context of the whole pump, it obviously gives more headroom at the inducer inflow, delaying the cavitation creation. As the results were not convincing, though, another, more complex parametric model was created.

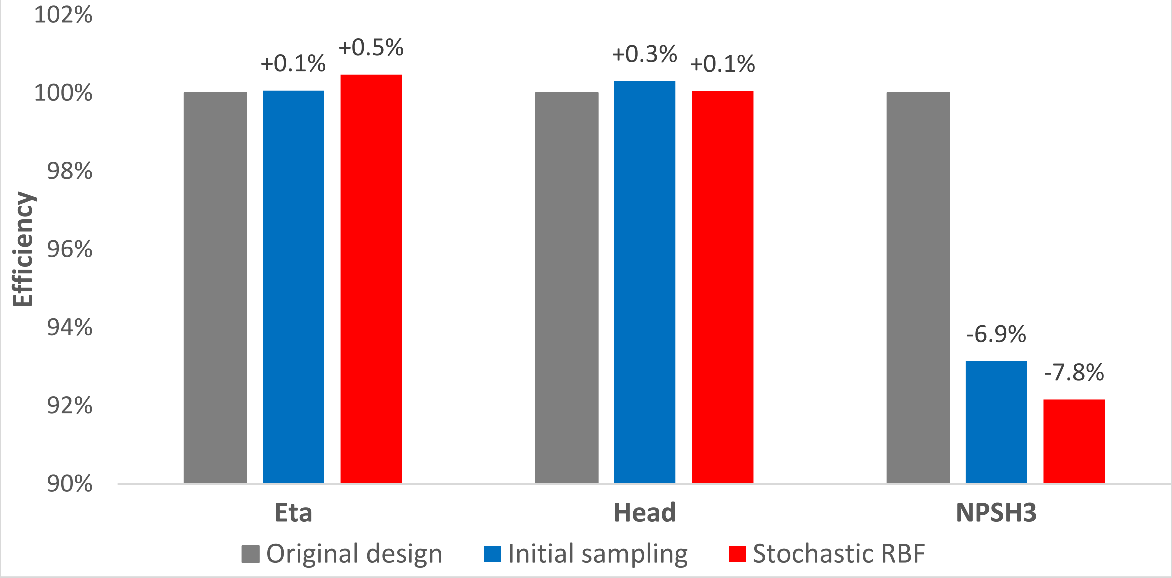

Figure 20: Hydraulic and NPSH₃ performance of the optimized designs, compared to the original (in grey).

5.6. Parametric model – suction and inducer

After analysing the outcome of the optimization, the inducer was added to the parametric model. As the next step, a different way of evaluating the velocity uniformity was chosen. The inducer was added to the model, and the simulation was set as transient. I.e., the inducer domain rotated around the *z* axis, with *rotor-stator* interface to the static suction domain. These changes were supposed to help in capturing the velocity objective *v* with higher accuracy. Inflow and outflow parts were again modified to improve the numerical stability. The boundary conditions and turbulence model remained the same as in the previous case.

The transient simulation gives more accurate results, as the rotor / stator interaction and inertia forces are considered. However, it also brings a steep increase in computational time and adds complexity to the automation process and post-processing. In a steady state simulation, only one “converged” value is used for the objective function. In the transient case, average values (typically over two last rotations of the impeller) need to be used. However, this approach allowed obtaining data for meridional velocity profile at the inlet part of the inducer blades. Using ANSYS CFD-Post built-in functions, the Hub-to-Shroud line was defined and meridional velocity v_m was exported in multiple points. The objective function was then defined as:

$$J = \sum_{i=1}^{N} \frac{\left(v_m^{(i)} - \bar{v}_m\right)^2}{\bar{v}_m}$$

where $\bar{v}_m = \frac{1}{N} \cdot \sum_{i=1}^{N} v_m^{(i)}$ is an average meridional velocity over the selected *Hub-to-Shroud line*. Number of points *N* was selected as *32*. Unlike head, the *v* values were not averaged over time interval. Instead, only the last value was considered. This was a limitation forced by technical reasons, as processing the data with CFD-Post every timestep would be computationally too taxing.

With objective function defined this way, the intention was to ensure as uniform fluid entrance to the inducer as possible. Supposedly, this should lead to better pressure distribution along the blades and slower creation of the cavitational areas. The assembled CFD model and inducer details, including the *Hub-to-Shroud line*, can be seen in Figure 21.

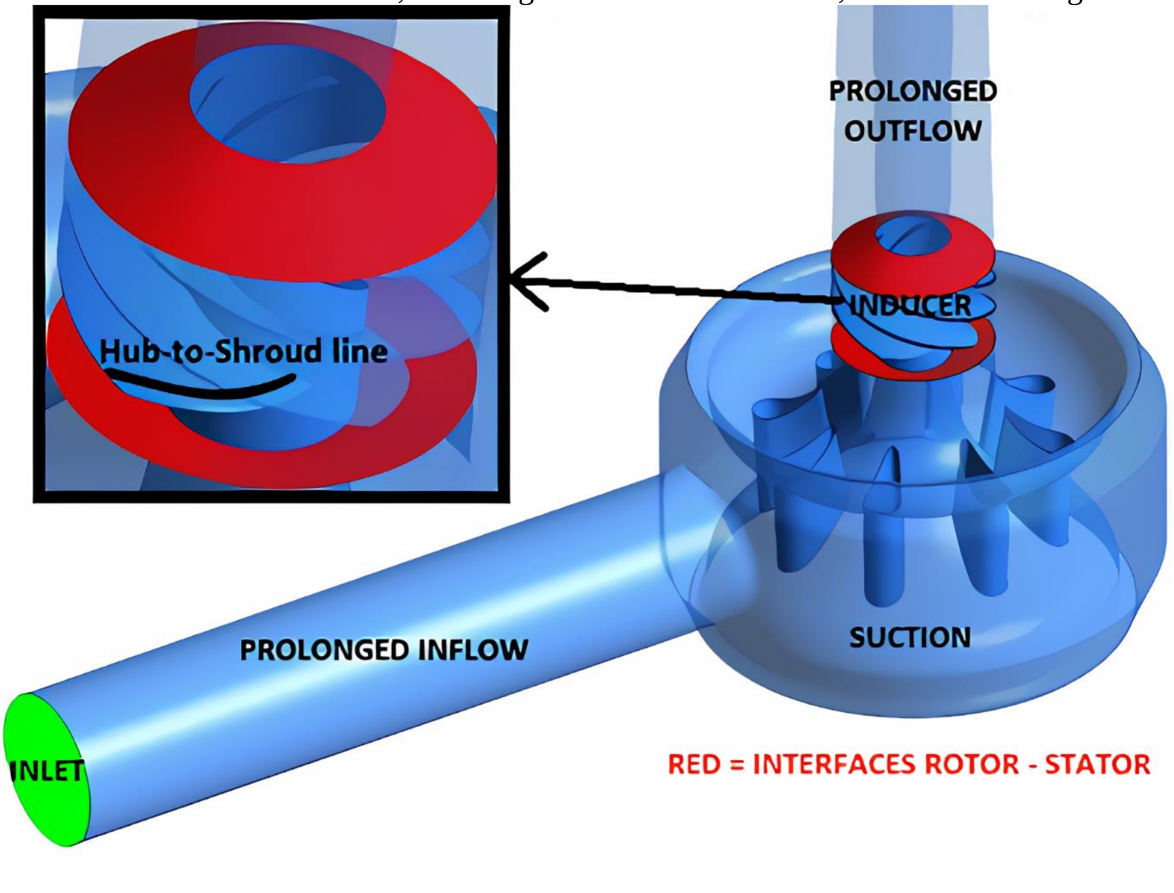

Figure 21: Parametric CFD model with the inducer and a detail of the Hub-to-Shroud line.

The mesh dependence was again tested with the aforementioned four different suction mesh settings, this time only with one selected design. The inducer mesh remained the same for all the variants. The results are displayed in Figure 22. As expected, the transient analysis has proven to be more accurate, and the result differences were negligible.

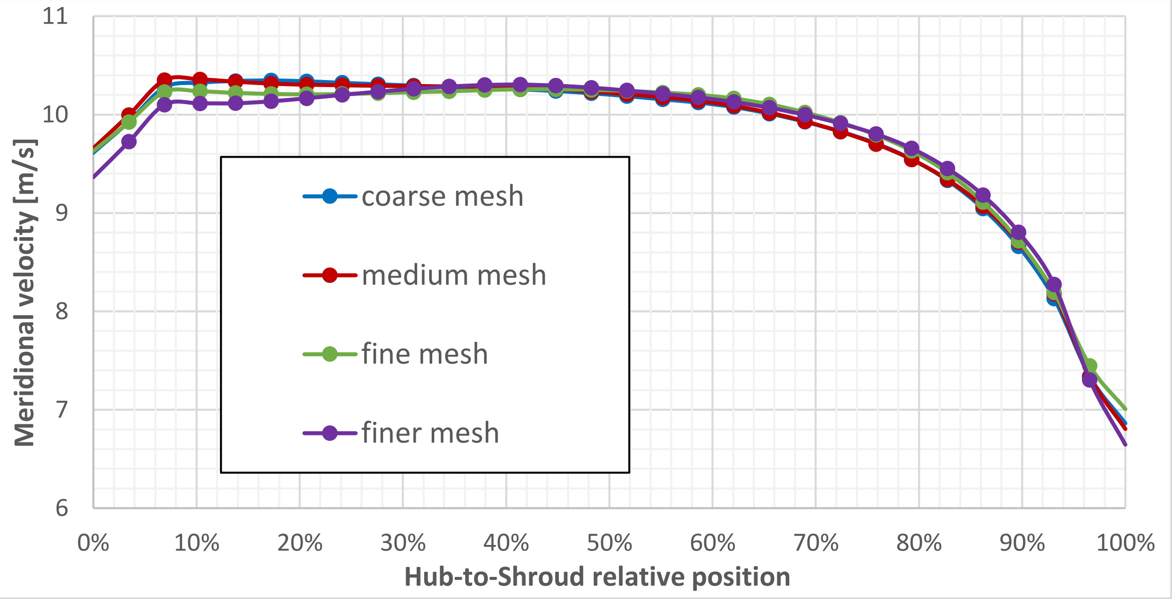

Figure 22: Meridional velocity profiles on the selected Hub-to-Shroud line.

5.7. Optimization and results – suction plus inducer

The optimization was started from the initial sampling. The number of samples remained the same, i.e., *38*. The geometries for all the initial sampling were generated and computed on an HPC cluster. Then, using these results, the Stochastic RBF optimization was performed. The procedure was set to generate 4 new samples in each iteration (same as in previous case), and run in an HPC environment. The progress of the objective function *J* during the optimization is shown in Figure 23.

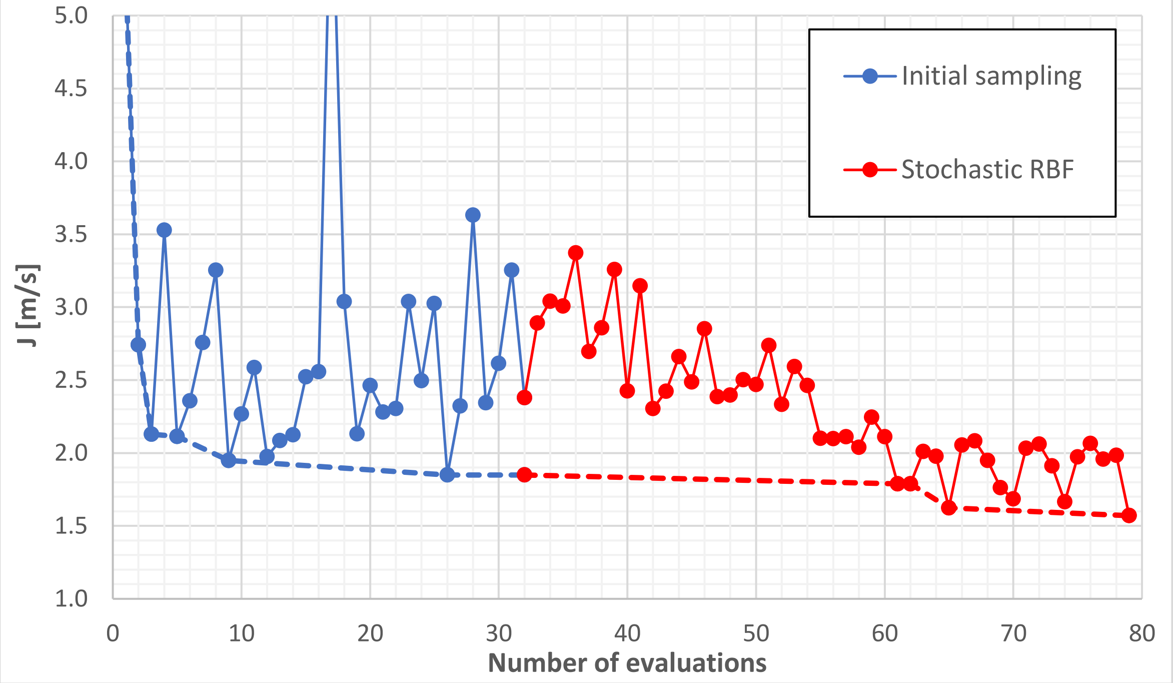

Figure 23: Optimization progress. The dashed line connects the best values.

For the best design selected from the optimization run, CFD model of the complete pump was assembled and full set of simulations was performed. The geometries and results comparison between the original and optimized designs can be seen in the following section. Quite in the contrary to the original expectations, this objective function yielded inferior results to the pressure drop version. Further analysis of the results has revealed that the suction shape has very little to no effect on the velocity profile in the middle and trailing parts of the inducer blades. Due to this, the shape optimization with respect to the meridional velocity uniformity has no significant effect once the cavitational areas start to develop. Lowering pressure loss in the suction, on the other hand, gives more *NPSH* reserve.

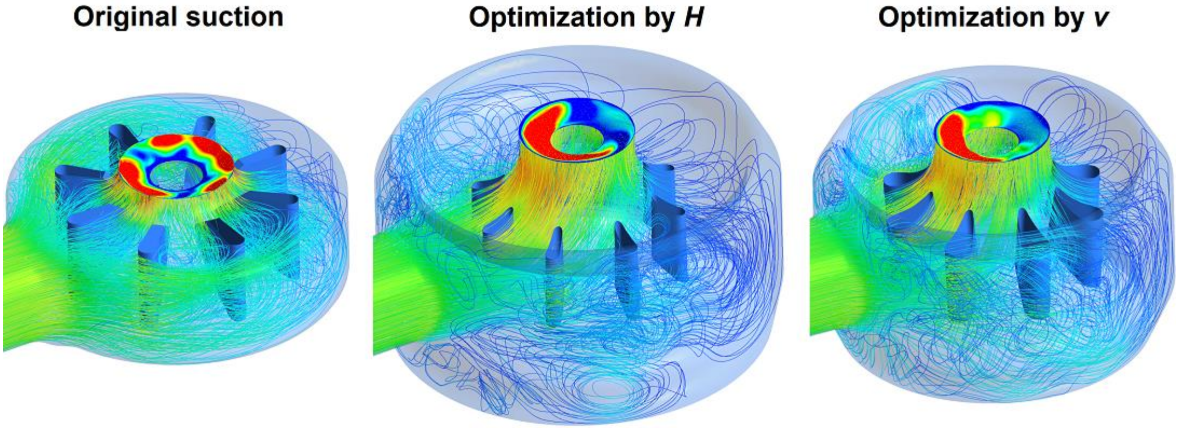

Figure 24: Suction designs and results of CFD analyses. The colours show differences in velocity profile at the suction outlet.

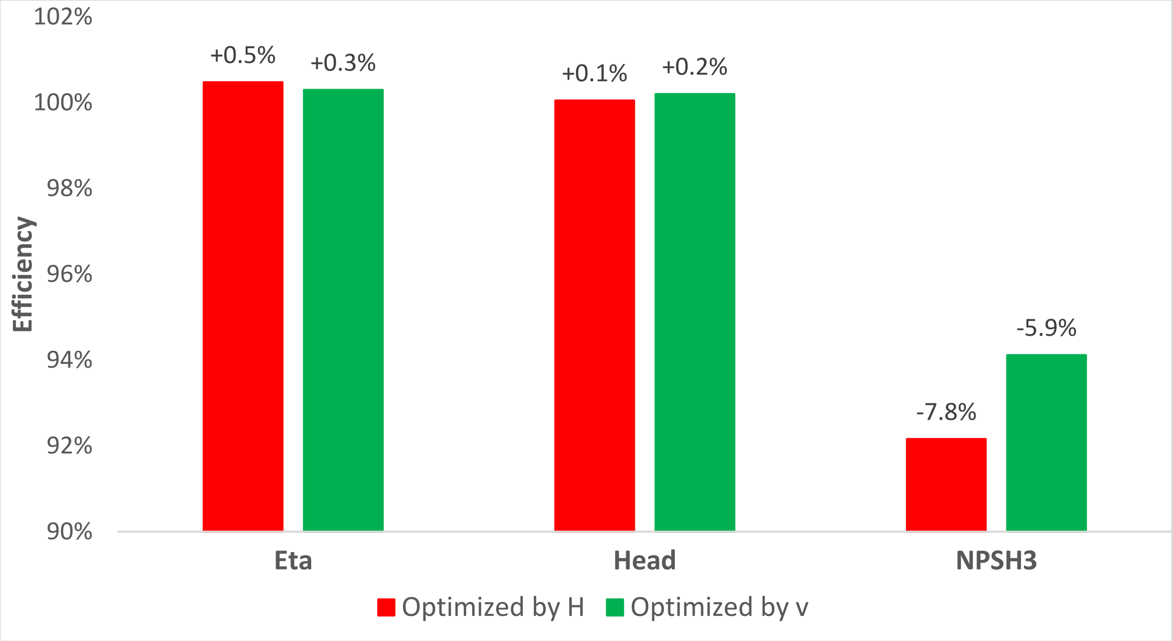

Figure 25: Hydraulic and cavitational performance of the optimized designs. The results are related to the original suction design.

To further analyse the accuracy of the numerical simulations, the geometry from the first optimization (with respect to the $H_{Suction}$) was tested on different meshes. The full set of both hydraulic and cavitation simulations was performed. Three new variants of the CFD model were prepared – one with coarse mesh, one with similar nodes count as the original one, but with different topology, and one refined. The nodes count ranged from 1.3 to 7 million, compared to the 1.9 million for the original mesh. The mesh dependence of $NPSH_3$ is displayed in Figure 26. In Figure 27, the performance characteristics (compared to the original design) can be seen.

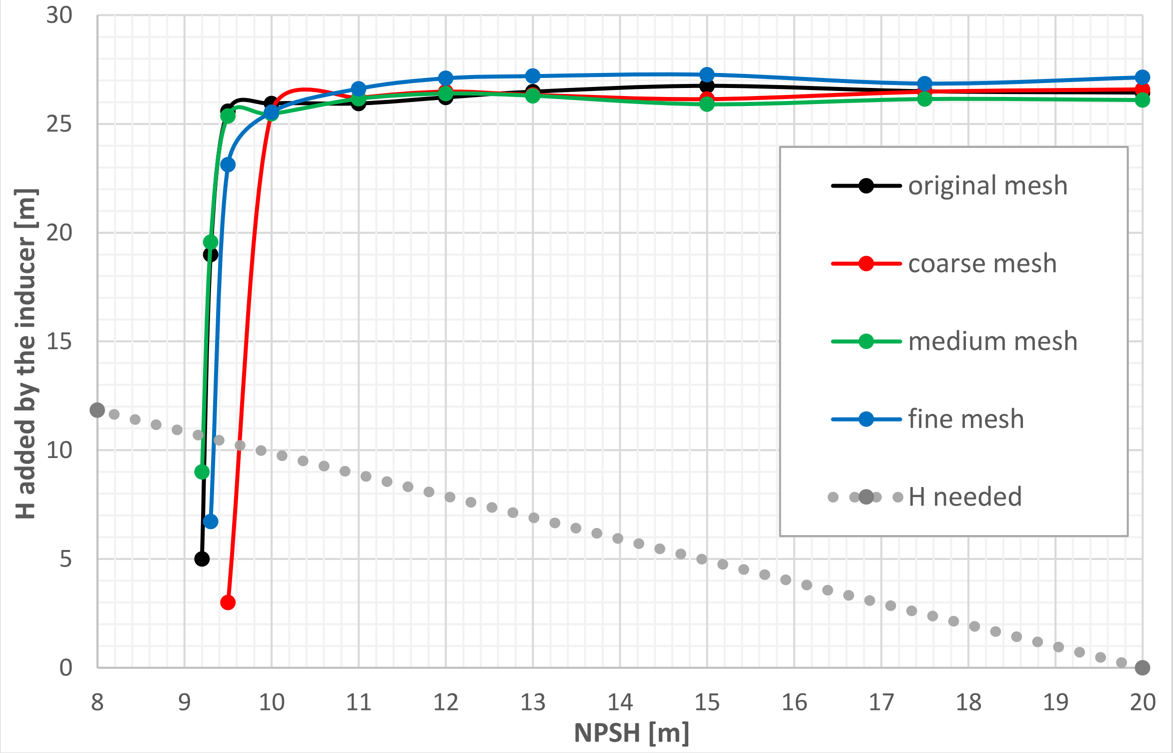

Figure 26: Head-drop curve mesh dependence. The dotted line shows the head boost required for the impeller to work properly. Once the static pressure generated by the inducer drops below this value, the cavitational breakdown of the pump occurs.

As can be seen in the figures, a relatively small mesh dependence can be observed. It is obvious that for efficiency and head, the improvement (over the original design) lies withing the interval of CFD inaccuracies, and is thus inconclusive. Only for *NPSH*$_3$, the difference is comfortably greater than the errors. Also, for this particular pump, the head drop is very steep, as can be seen in Figure 26. This leaves smaller range for possible errors and has a positive effect on *NPSH*$_3$ computation accuracy.

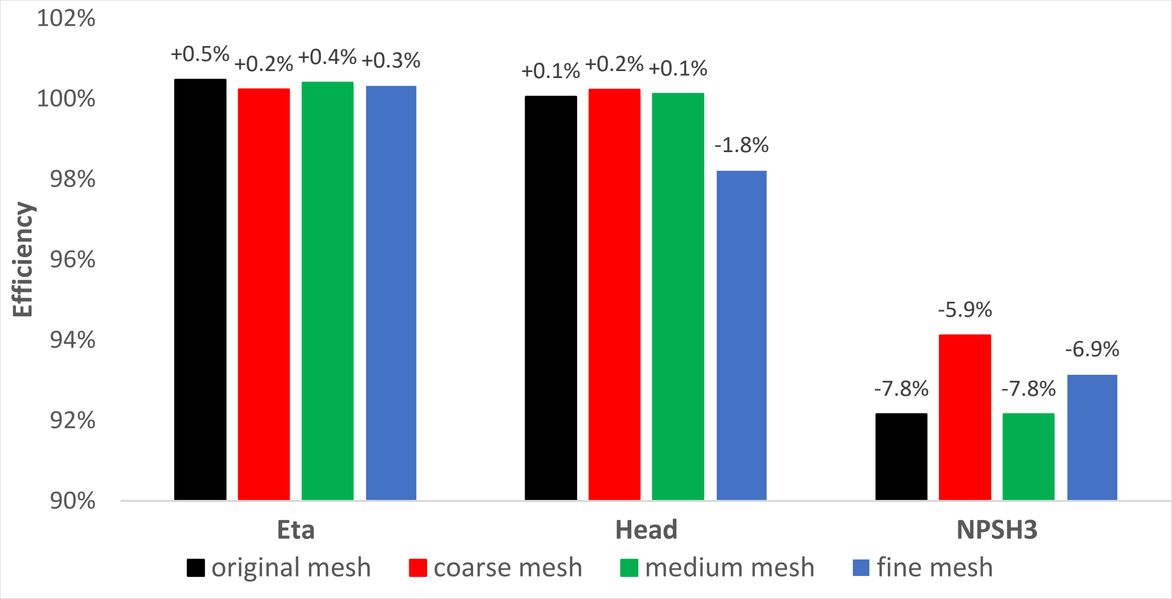

Figure 27: Mesh dependence of H-optimized design (in comparison with the original design).

5.8. Summary and conclusion

The automated parametric model was successfully created, tested and connected with a selected optimization method. Two different approaches were tested. First, a simple model with the suction only and set as computationally (relatively) inexpensive *steady-state* simulations. Second, a more complex model with the rotating inducer included, and using the computationally demanding transient simulations. In the end, the simpler model worked out better. The results of the numerical simulations of a complete pump suggest the model with suction optimized for $H_{Suction}$ displays both the best efficiency and cavitational properties. For hydraulic parameters the differences are too small to rule out the mesh-related errors, though. For the *NPSH*$_3$ characteristics, on the other hand, the changes are more substantial. The objective defined by velocity uniformity did not work very well, possibly because of lower accuracy on the rotor-stator interface. Overall, results of this optimization were not convincing. The parametric geometry and CFD simulations of the suction has proven to be much more problematic than was originally expected. The biggest concern was the mesh-dependence of the results.

However, this case was successfully used for testing of the Stochastic RBF method implementation for the purpose of hydraulic shapes optimization. For this, modifications of the original Matlab code were necessary, to make it capable of dealing with failed samples and “restarts”, which was important for running the optimization under a real-world condition on an HPC system with a PBS scheduler. It was also a good opportunity for testing all routines related to automation of the numerical simulations – parametric geometry creation, mesh generation, dealing with failed designs, CFD model assembling, running the CFD simulations and evaluating and post-processing the results. Valuable experience was obtained as part of the process, as there are many technical challenges related to the practical realisation of the shape optimization. This experience was then used in the next optimization cases.

6. Case 3 - multi-objective optimization of a stator

Objectives: Pump efficiency at working range.
Solution: ANSYS Workbench parametric model driven through text script files, transient simulation (full geometry). Three objectives (efficiency at *90%*, *100%* and *110%* of Q_{OPT}). Stochastic RBF optimization method + scalarization.
Results: Efficiency increased from 2% up to 5% in the working range.

6.1. Introduction

The next case for stator optimization was a diagonal pump with adjustable blades, of specific speed $n_s = 360$. It was an already existing, but outdated pump from SIGMA. When compared to the newer designs, the efficiency was not competitive. The numerical simulations identified the diffuser to be responsible for significant part of the efficiency loss. The impeller efficiency was very high (over 90 %), but the circumferential component of the velocity remained very high through the diffuser. This increased the hydraulic losses in the outflow part, lowering the hydraulic performance of the pump. Based on previous experience with diffuser optimization, it was decided to do another diffuser optimization. Unlike in the previously described "*Case 1*", the plan was to use the Stochastic RBF method and perform an automated and more complex optimization at multiple flow rates, to ensure a good performance at the whole working range. This means that the TBR method could not be used, as the inflow and outflow shapes could not be ignored at the non-optimal flow rates.

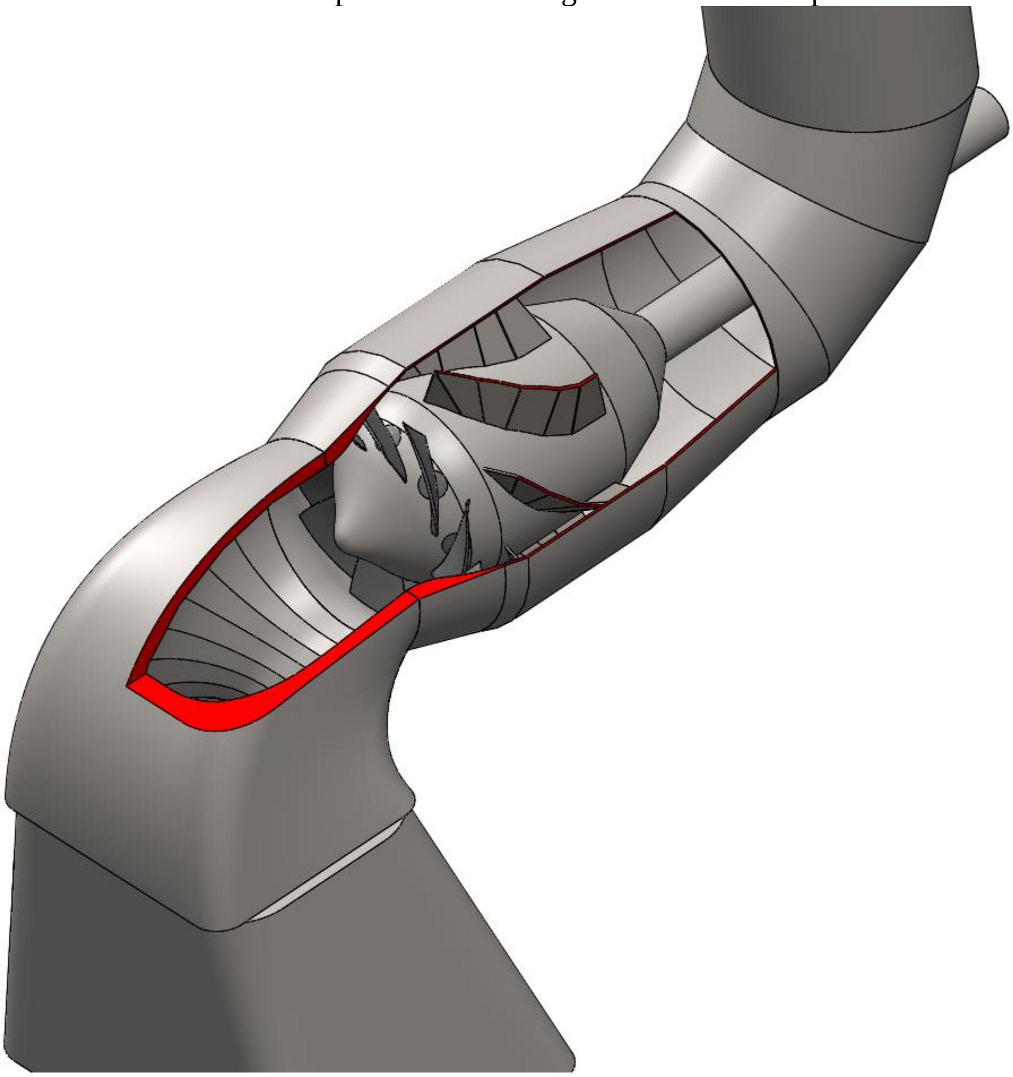

Figure 28: 3D model of the pump with bent diffuser blades.

Just as in the previous case, the plan was to produce a design that can be manufactured by the so-called *metal sheets bending*. I.e., relatively simple blades (with constant thickness, defined on two camber lines) and "straight" meridional profile. It should be noted that in the ideal case, the actual bent shape should be considered for optimization. The bending in general has a small negative impact on the hydraulic performance, as sharp edges are created on the blade. For each blade shape, there are practical limits for the bending. As a result, it can happen that one shape of the blade is deformed more than another, resulting in bigger performance drop. However, for the optimization this had to be ignored, as doing the optimization with respect to the bent shape would pose a major challenge. This is because the software tools used for the parametric model creation are not suitable for such option. I.e., both geometry creation and mesh generation cannot be automated in the bent shape. Thus, the optimization was based on the assumption that the resulting design will display performance good enough to have a reserve for the decrease caused by the bending.

Another empirically verified assumption was that only one selected impeller blades adjustment can be considered for the optimization. I.e., if the pump displays good performance for one setting, then it performs well for a range of adjustments. Then it is sufficient to only perform the optimization for one setting (of the impeller blades).

6.2. Parametric model and automation

For geometry and mesh parametric creation, a Workbench project was created. The process of setting the parameters, geometry and mesh creation, and consequent CFD model assembly was recorded as a template script. Parameter values recorded in the script are then replaced by *keywords* (*par_1*, *par_2*, …). When the code responsible for the evaluation of the objective function is called, it simply replaces keywords in the template by the input parameters, calls Workbench in batch mode, performs the CFD simulations, extracts and evaluate the results and sends the output information (either objective values, or "*failed design*" flag) back to the optimization routine. For the scripts related routines, experience from the previous "*Case 2*" was used. The geometry shape was driven by 18 parameters, listen in the following Table 7, some of the parameters are shown in Figure 30.

Table 7: Parameters

Parameter	***Description***
1	*Beta angle - hub - leading edge*
2	*Beta angle - hub - relative value at 25%*
3	*Beta angle - hub - relative value at 50%*
4	*Beta angle - hub - relative value at 75%*
5	*Beta angle - hub – trailing edge*
6	*Beta angle - shroud - leading edge*
7	*Beta angle - shroud - relative value at 25%*
8	*Beta angle - shroud - relative value at 50%*
9	*Beta angle - shroud - relative value at 75%*
10	*Beta angle - shroud – trailing edge*
11	*Meridional length – hub*
12	*Meridional length – shroud*
13	*Angle between hub and rotation axis*
14	*Position of leading edge – hub*
15	*Position of leading edge – shroud*
16	*Position of trailing edge – hub*
17	*Position of trailing edge – shroud*
18	*Sweep angle (defined at the shroud)*

The beta angles (at both hub and shroud) were defined at leading edge, trailing edge and three interior points (located at 25%, 50% and 75% of the blade length). As it is often recommended to “hold” the angle towards the blade end, the value at 95 % was considered the same as at the trailing edge. To ensure desired shapes (i.e., monotonicity), the inner values were set as relative with respect to the neighbourhood points:

$val_{95\%} = val_{100\%}$
$val_{50\%} = val_{00\%} + coeff_{50\%} * (val_{95\%} - val_{00\%}) * 20/19$
$val_{25\%} = val_{00\%} + coeff_{25\%} * (val_{50\%} - val_{00\%})$
$val_{75\%} = val_{50\%} + coeff_{75\%} * (val_{95\%} - val_{50\%}) * 10/9$

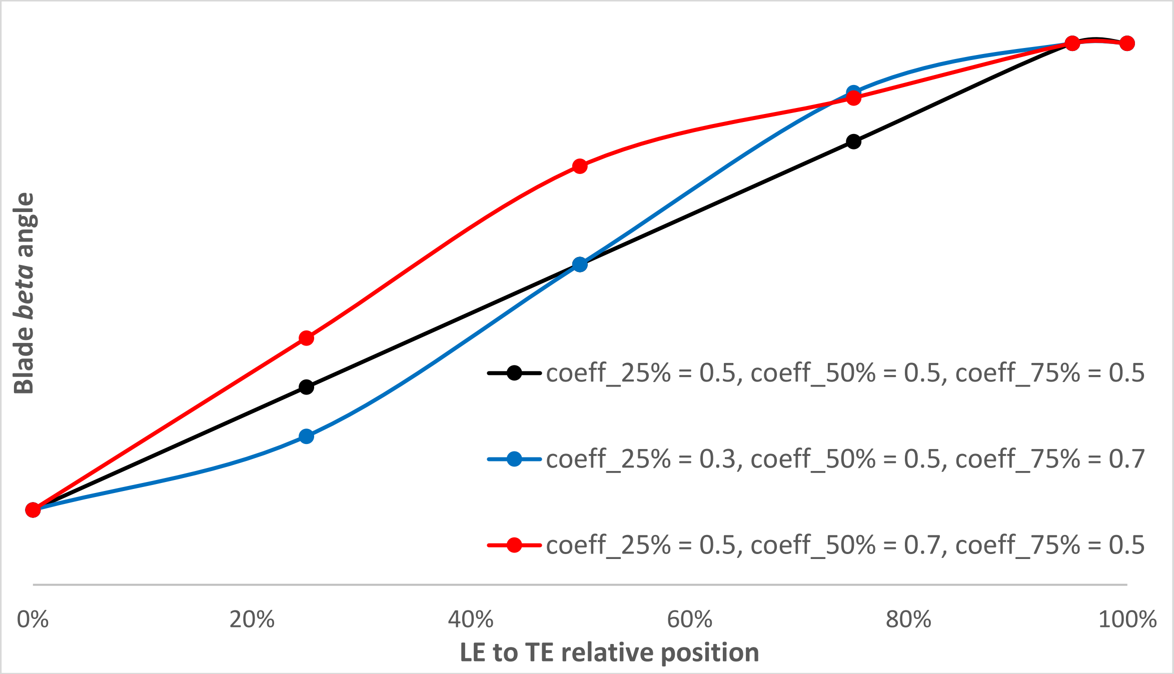

Figure 29: Examples of angle distribution for selected coefficient settings.

ANSYS DesignModeler was used for diffuser and outflow geometry creation. The structured impeller mesh was generated by TurboGrid, and ANSYS Meshing was used for elbow meshing. The mesh size was ca 1.5 mil. of nodes. The simulation was set as transient, with rotating impeller and the solution time was ca. 20 hours on a 16-cores HPC cluster node. *SST k-Ω* turbulence model was used. With a Workbench script, the CFD model is then updated and solver input file created. The solver is run for each desired setting, defined in a ccl file. This file contains the expression language understood by CFX, and can be used as an input for a command line run. It is thus possible to set arbitrary parameters of the CFD model, such as flow rate, pump *rpm*, fluid medium properties, maximal allowed number of iterations etc. Three ccl files with the respective settings were created, and the simulations were run on an HPC cluster. For optimal utilization of the available resources, the duration was limited to 2 hours (the fastest queue on the cluster). Once the run finishes, the results are extracted and processed. If the simulation is decided as “finished” by the routine, the results are saved to a csv file and used by the optimization routine. If not, then the simulation continues, i.e., another 2-hours run is sent to the PBS scheduler.

Table 8: Parameter ranges

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1st sampling	30	0.3	0.3	0.3		15	0.3	0.3	0.3		700	700	15	20	20	-10	25	25
	40	0.7	0.7	0.7		35	0.7	0.7	0.7		1300	1300	25	200	200	15	150	150
2nd sampling	31	0.3	0.25	0.25	80	17	0.5	0.4	0.5	80	1100	1100	15	20	20	-10	25	25
	37	0.5	0.45	0.4	90	23	0.7	0.6	0.7	90	1300	1300	25	200	200	15	100	100

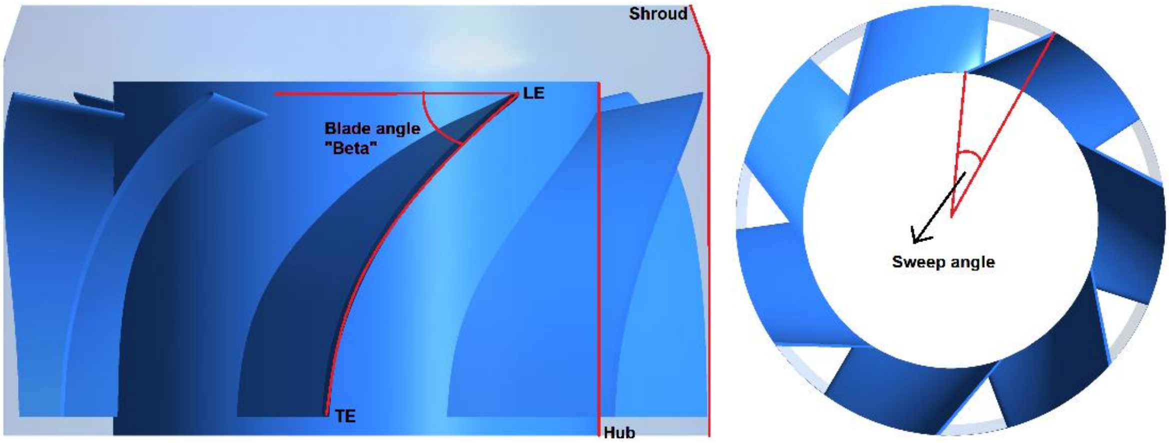

Figure 30: Visualisation of selected parameters.

6.3. Testing the parametric model

First, the trailing edges were set as fixed to 90 degrees. For the 16 remaining parameters, the initial sampling was created. The default (sampling) size of the Stochastic RBF method was used, i.e., $2 * (N + 1) = 34$ samples, where $N = 16$ is the number of parameters. Out of these 34 samples, 25 were successfully created. The remaining ones failed for geometry reasons. All the samples were evaluated at three different points (flow rates) – *90%*, *100%* and *110%* of the design point Q_{OPT}. After evaluation and careful consideration of the results, the parametric model was slightly adjusted. First, the beta angles at trailing edge were enabled as parameters, instead of being fixed to a constant value. Second, the ranges of the parameters were modified, mostly narrowed. The comparison can be seen in Table 8.

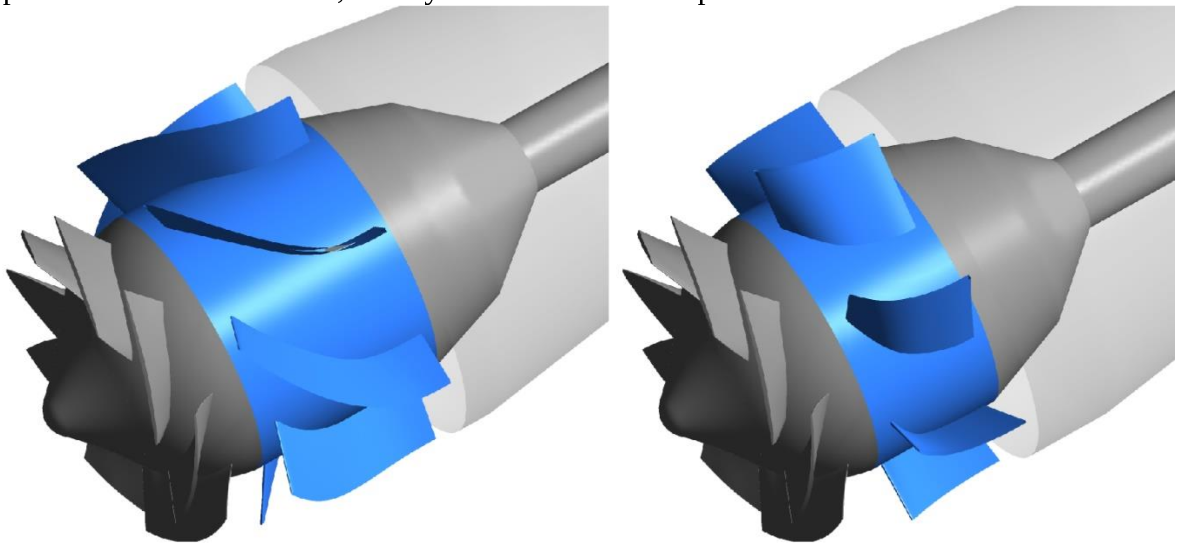

Figure 31: Examples of diffuser geometry (in blue).

With these changes, a new sampling of a size 38 was created. With the new settings, the error rate of the parametric model lowered - out of the 38 samples, 36 were successfully generated, and only two failed. Next, the simulations were again run on the HPC cluster, and the results were extracted and evaluated. Average efficiency value of two last impeller rotations was used as the objective value. This is the commonly used practice for evaluating the results of transient simulations. The CFD results of both samplings are compared in Figure 33. For better clarity, the numbers are shown in the so-called *parallel coordinate plot*. As can be seen, the second sampling is more focused on the high-efficiency area, and contains some more promising designs. Thus, data from the second initial sampling was used for starting the optimization.

6.4. Optimization and results

The same method as in the previous case (“Case 2”) - *Stochastic RBF* – was used. As it is a single objective optimization method, the scalarization of the three objectives (efficiencies at *90%*, *100%* and *110%* of Q_{OPT}, further denoted as η_{90}, η_{100} *and* η_{110}) had to be used. Based on recommended scalarization methods in (24), the objective function was defined like this:

$$J = max(90 - \eta_{90}, 91 - \eta_{100}, 88 - \eta_{110}) + \frac{\eta_{90} + \eta_{100} + \eta_{110}}{1000}$$

90, *91* and *88* were “thresholds”, i.e., efficiencies required to be exceeded (at the respective flow rates). By considering the maximum of the “missing efficiencies”, the optimization is basically forced to try to meet all the required thresholds. Unlike a scalarization by summing the objectives, where gain in one objective can outweigh the loss in another. The small additive member in the scalarized function is usually recommended (24) to help guiding the optimization in a case when an objective is improved, but the maximum remains the same.

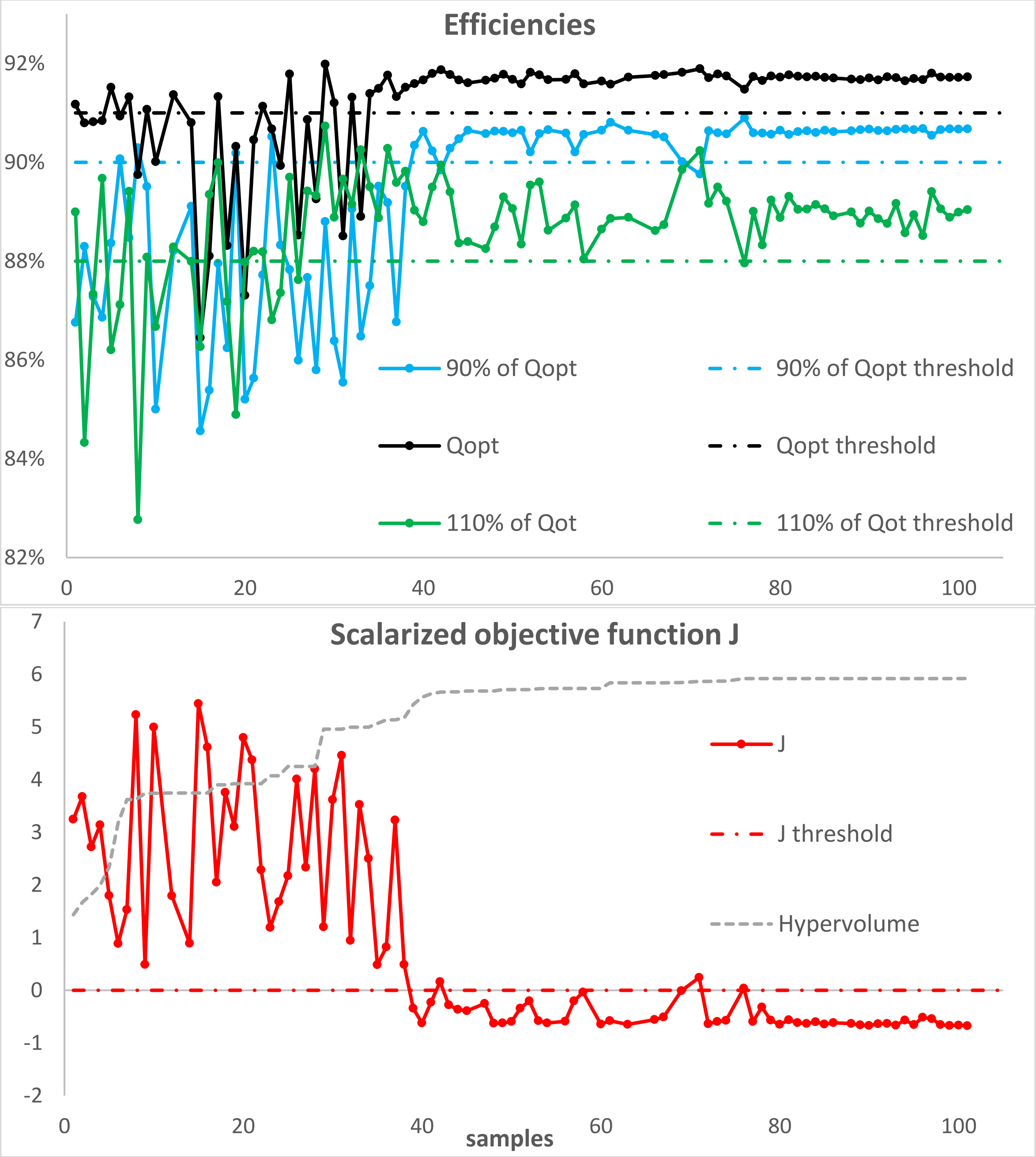

Figure 32: The objectives and scalarized function during the optimization. The grey dashed line shows the hypervolume of the evaluated designs in the objective space.

The optimization was performed at an HPC cluster. The limiting factor for the parallelization was the number of available ANSYS CFX licences, there were 25 of them. Thus, to obtain the results as fast as possible, Stochastic RBF was set to generate 8 new samples in every iteration. For three working points per sample, this means 24 CFD simulations are needed per optimization iteration – each utilizing 16 CPU cores. Most likely, increasing the number of newly created samples per iteration can possibly lower the optimization speed and "efficiency" (when measured in terms of objective function evaluations). This is on the other hand offset by the parallelization (if the resources are available) and improved exploration. There were 8 Stochastic RBF iterations in total. Out of the 64 samples, 11 failed and 53 was successfully generated and evaluated. The optimization was stopped once there were no significant improvements over successive iterations. The optimization record (efficiency objectives and the single scalarized objective) can be seen in Figure 32.

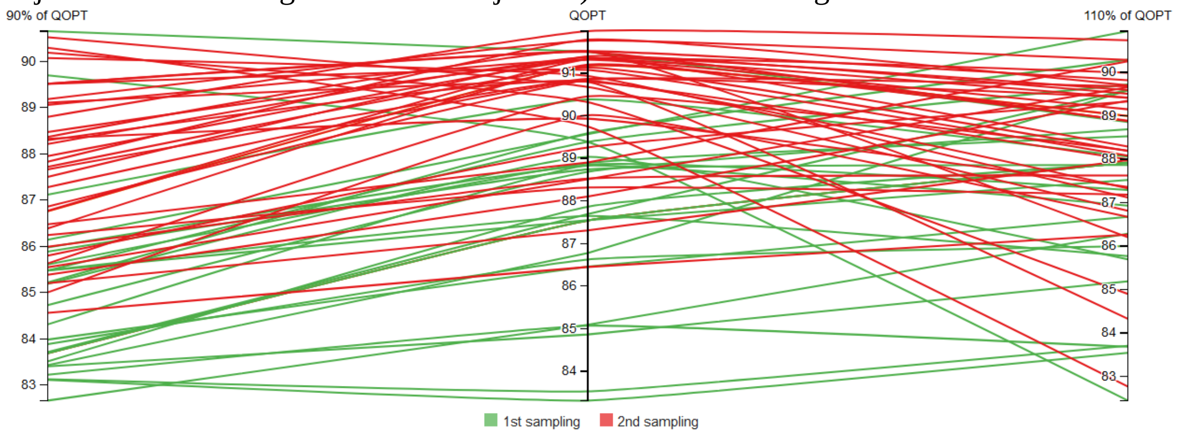

Figure 33: Parallel coordinates plot of CFD results – 1st and 2nd sampling. The y-axis is for the efficiency.

The optimization successfully exceeded the efficiency thresholds in all three flow rates. For the final comparison, the design found by the Stochastic RBF was then modified for the manufacturing (by sheet metal bending technology). For all considered blade adjustments (3, 0, 3, 6, 9 and 12 degrees), the meshes were created, and full sets of CFD simulations were performed. The comparison of *Q-Efficiency* characteristics between the pump with original diffuser and optimized diffuser with the bent blades can be seen in Figure 34.

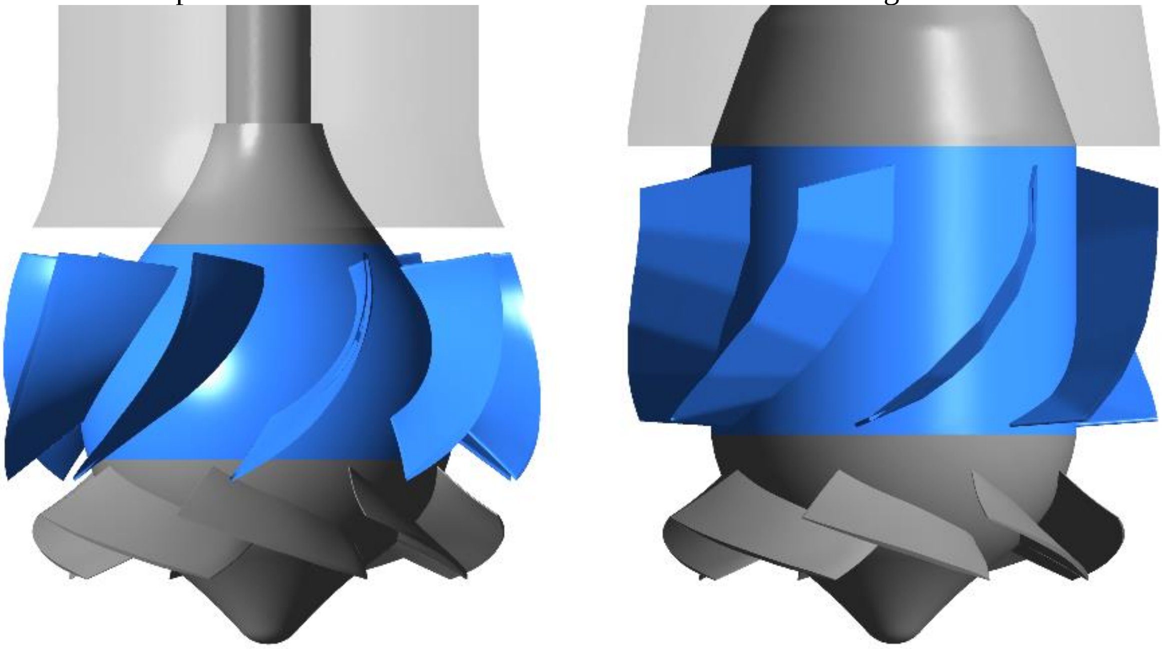

Figure 34: Comparison of original (left) and optimized (right) design.

The improvements in efficiency vary from 1% up to ca. 7%. The new optimized design has a bit narrower operation area with high efficiency. For a pump with adjustable blades this is not a problem, though, as the setting can be adjusted to the optimal configuration for any scenario. Thus, in practical application, the performance improvements are in a range from 4 to 7%. For large pump with MWs of power, this is a significant improvement.

6.5. Conclusion

The automated optimization was successfully used as part of the hydraulic design process. The parametric model was created, connected to a Matlab optimization code and run in an HPC environment. The scalarization approach worked reasonably well, and the efficiency was improved for the whole relevant working range. The final sets of numerical simulations for various impeller blades adjustments (and with the bent diffuser blades) confirmed the simplifications of the CFD model (not considering the bent blades, only one impeller blades adjustment) as justified. Overall, we can conclude such approach to diffuser optimisation works reasonably well, if the parametric model is well created and necessary computational resources are available.

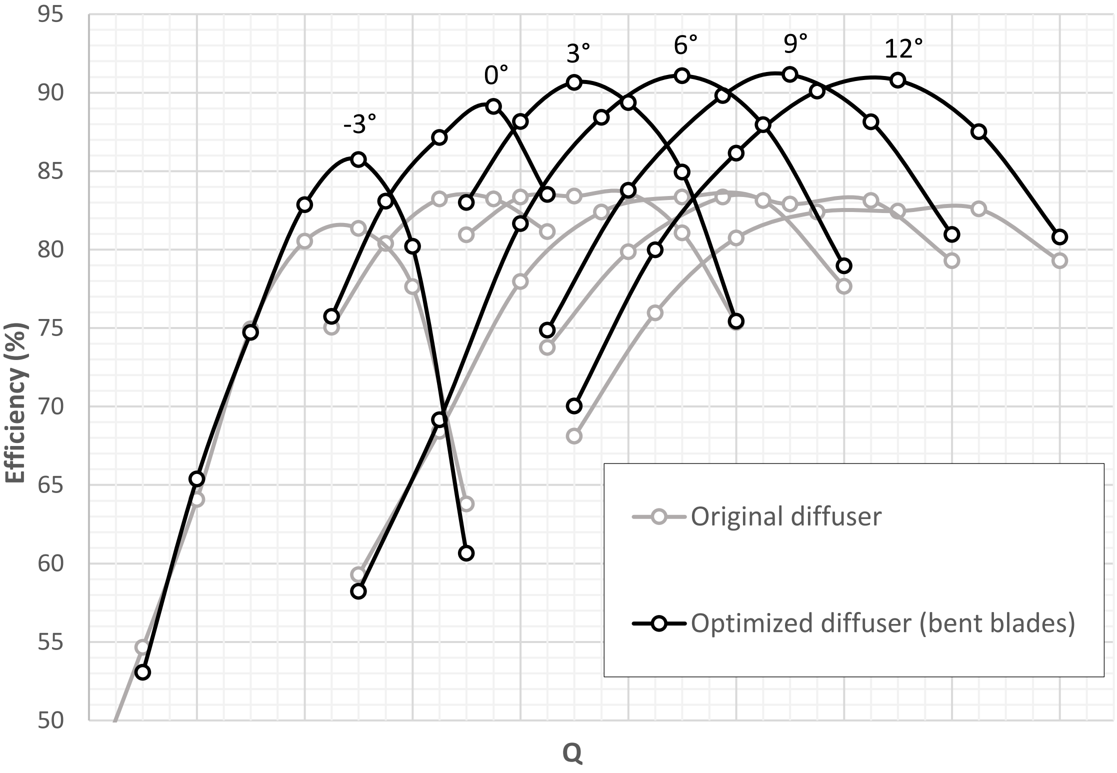

Figure 35: Efficiency comparison with the old design.

Another approach that was successfully tested in this case was the parallelization of the optimization. The ability to create multiple new samples in every iteration is a significant advantage of the modern optimization methods. As long as the computational resources are available, this can significantly increase the amount of information obtained per time unit. Without extensive numerical testing (which would be unacceptably expensive for CFD-based objective function) it is not possible to support this claim for the hydraulic shapes optimization. But for fast test functions, such behaviour was observed in (22). In “serial” optimization, every consecutive sample is generated with the knowledge of objective functions of the previously generated samples. When generated multiple samples, these samples lack the information from the actual information. Thus, diminishing returns of the parallelization can be expected.

7. Case 4 - multi-objective optimization of a stator

Objectives: Improving efficiency for the sub-optimal flow rates, maintaining it at the higher flow rates.
Solution: ANSYS Workbench parametric model driven through text script files, transient simulation (full geometry). Three objectives (efficiencies at *80%*, *100%* and *120%* of Q_{OPT}). Stochastic RBF optimization method + scalarization, K-RVEA.
Results: Efficiency increased from 2% up to 5% in the working range.

7.1. Introduction

Another case for stator optimization was a diagonal pump with axial diffuser, of a specific speed $n_s = 200$. The hydraulic design was challenging, as the operation site constraints forced relatively short total length of the pump. This caused some troubles, as the short meridional length did not really fit into the recommended ranges for this specific speed. The customer also required high efficiency in a wide working range. The resulting design, created by an experienced hydraulic expert, met these demands. However, the CFD simulations suggested a sharp decrease of performance at ca. *80 %* of the design point Q_{OPT}. It was thus decided to perform an optimization of the design. As the stator optimization is easier than optimizing impeller, it was the preferred option.

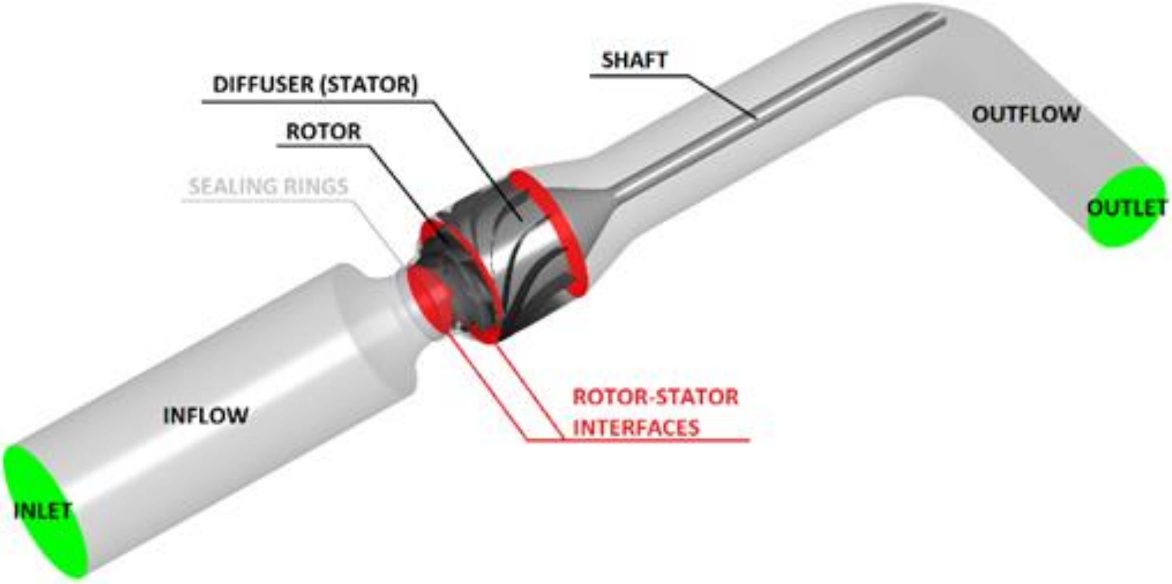

Figure 36: CFD model of the pump.

Based on the previous experience, the optimization was focused on the diffuser. Efficiencies at three different points were chosen as the objectives. These points followed the experience from the previous cases. What was different was the fact that the baseline design was not outdated, but a newly created one, with good performance (given the operation site limitations) and difficult to improve. Also, the influence of the sealing rings is considered and discussed here, which is not common, as the sealing rings are difficult to include in the CFD model. First, the already tested Stochastic RBF method and scalarized objective function were used. Next, a new, modern method, called K-RVEA, was used. It is a SAO, Kriging-based multi-objective method, that effectively uses available information from both parametric and objective spaces. This optimization case was an opportunity for testing the method, and to compare its performance to Stochastic RBF. Designs found by both methods were extensively evaluated and compared to each other and to the baseline design. In the conclusion, the results and the advantages of the multi objective approach are discussed. The influence of the sealing rings on the optimization is also considered.

7.2. Parametric model and automation

The parametric model was created in a similar way as in the previous diffuser optimization case. I.e., an ANSYS Workbench project, driven by Python-controlled scripts. As can be seen in Figure 37, the sealing rings were also included in the model. The flow from the sealing rings can enter the passage in a direction perpendicular to the "main" flow, and influence the rotor-stator interaction. The sealing rings are rarely included in CFD simulations, as the geometry and mesh creation are challenging. Instead, the effect on pump performance is estimated empirically.

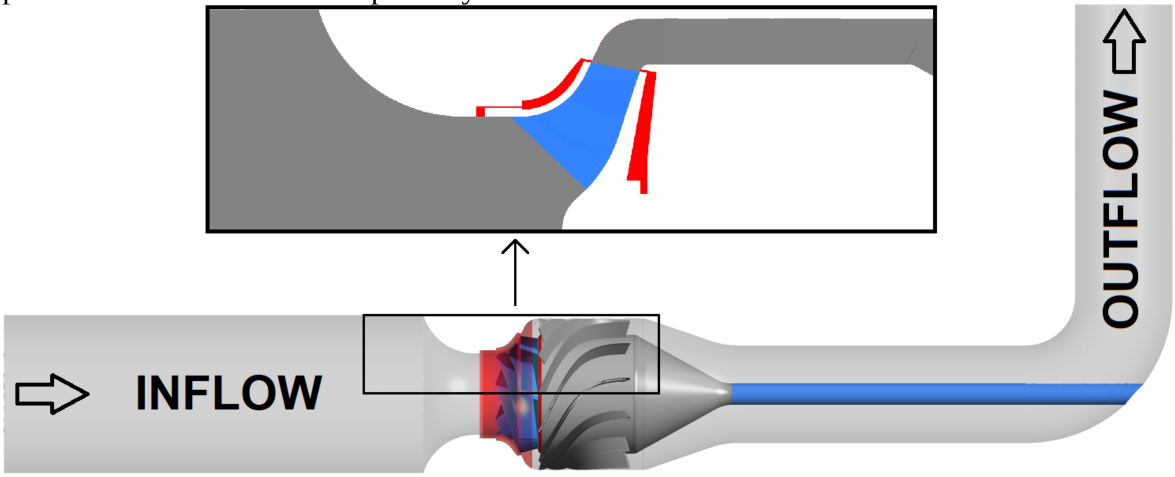

Figure 37: Meridional section of the CFD model. Rotating parts (impeller and shaft) are displayed in blue, the sealing rings are in red.

For the parametric model, the leakage part had to be omitted. Creating a parametric model of the leakage and its automation would be very difficult, as TurboGrid cannot be used for this. It was also assumed that this would not affect the comparison of different design significantly. For these reasons, the CFD model for optimization was simplified in this way. The simplified CFD model with diffuser detail can be seen in Figure 38, and the comparison of results between the models with and without sealing rings can be seen in Figure 39.

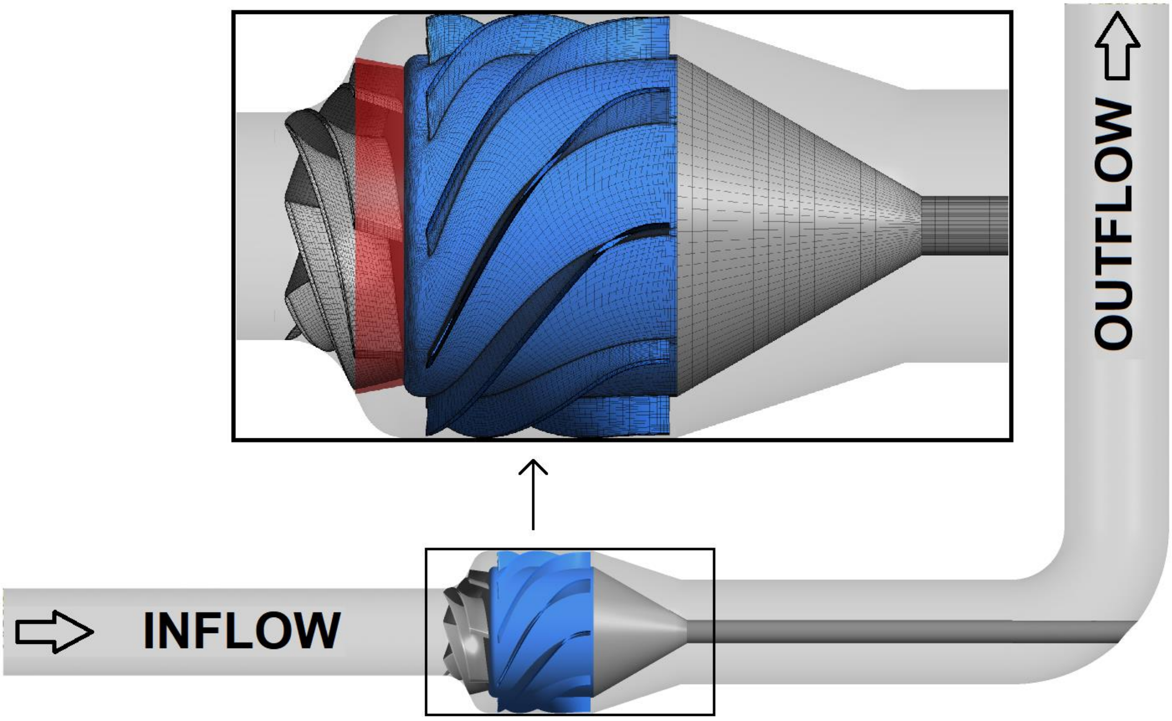

Figure 38: CFD model used for optimization. The diffuser is in blue, impeller-diffuser interface is in red.

The CFD model was set as fully-transient, with 154 time-steps per impeller rotations. This equals to 2.34 degrees of an impeller rotation during each time-step. The pump has 995 rpm; thus, the time-step value was $\Delta t = 60 / (154 \cdot 995) = 0.00039157s$. Boundary conditions (BC) were set as *Mass Flow Rate* at the outlet and *Total Pressure* at the inlet. As the turbulence model, *SST k*-Ω was used. The mesh size was ca. 1.5 mil. of nodes for most of the designs. With simulations running for 10 up to 20 impeller rotations, the solution time was usually 16 to 20 hours to termination on a 16-cores HPC node.

The parametric geometry model has been an evolved version of the previously used models, slightly enhanced by adding a few more parameters. In total, there were 22 geometric parameters, listed in Table 9. Some of these parameters are displayed in Figure 40. The blade angle beta was defined at six points for both hub and shroud. To prevent “waving” of the angle curves, the inner values were set “relatively”, in the same way as in the previous case:

$val_{95\%} = val_{100\%} - diff$

$val_{50\%} = val_{00\%} + coeff_{50\%} * (val_{95\%} - val_{00\%}) * 20/19$

$val_{25\%} = val_{00\%} + coeff_{25\%} * (val_{50\%} - val_{00\%})$

$val_{75\%} = val_{50\%} + coeff_{75\%} * (val_{95\%} - val_{50\%}) * 10/9$

By setting the angle-shaping parameters like this, monotonicity of the curve is usually met, because if one of the points (that is used by the other points – for example $val_{50\%}$) changes, then angle values at the following points (25 %, 75 %) change, too, despite the fact that the parameters $val_{25\%}$ *and* $val_{75\%}$ remain the same. If the parameters were set independently, then for high $val_{25\%}$ and low $val_{75\%}$ it might happen that the beta value at 25 % would actually be higher than at 75 % - resulting in an undesired shape of the curve. The biggest concern with this approach, on the other hand, is that it can make the relations between the parameters more complex. I.e., for example, the angle value at 25 % does not only depend on $val_{25\%}$, but is also influenced by $val_{50\%}$. This can make the optimization problem more complex.

Table 9: Parameter description

Parameter	***Description***
1	*Beta angle - hub - leading edge*
2	*Beta angle - hub - relative value at 25%*
3	*Beta angle - hub - relative value at 50%*
4	*Beta angle - hub - relative value at 75%*
5	*Beta angle - hub - value at 95 % - difference to trailing edge value*
6	*Beta angle - hub – trailing edge*
7	*Outlet diameter – hub*
8	*Meridional length – hub*
9	*Leading edge position - hub - distance from the inlet*
10	*Trailing edge position - hub - distance to the outlet*
11	*Sweep angle (defined at the shroud)*
12	*Beta angle - shroud - leading edge*
13	*Beta angle - shroud - relative value at 25%*
14	*Beta angle - shroud - relative value at 50%*
15	*Beta angle - shroud - relative value at 75%*
16	*Beta angle - shroud - value at 95 % - difference to trailing edge value*
17	*Beta angle - shroud – trailing edge*
18	*Meridional length – shroud*
19	*Leading edge position - shroud - distance from the inlet*
20	*Trailing edge position - shroud - distance to the outlet*
21	*Outflow angle – hub*
22	*Outflow angle - shroud - relative to the hub angle*

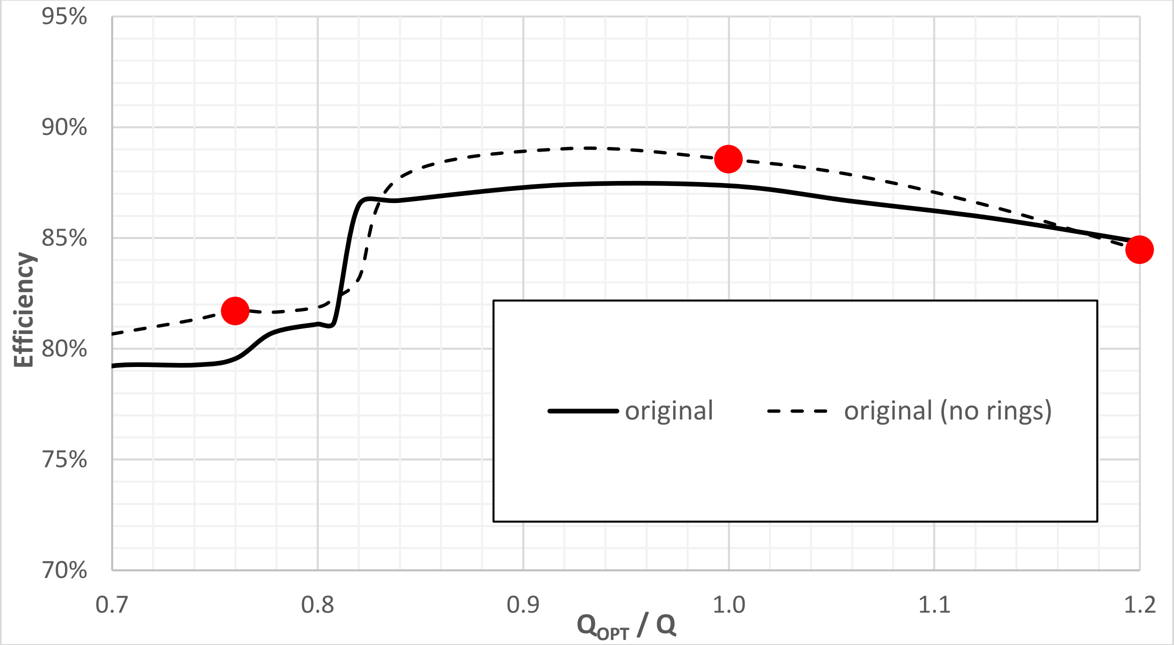

Figure 39: CFD results – with and without rings. The red dots show the three flow rates at which the efficiency was optimized.

7.3. Testing the parametric model

After deciding the parameter ranges, the $2 * (N + 1) = 46$ samples were generated, and the geometry and mesh creation was tested. Out of these, 37 was created successfully, and 9 failed. To further enhance the parametric model, two different settings of hexa-mesh generation were compared – a premade topology and so-called ATM method, both available in TurboGrid. The premade topology approach is faster and ensures consistent mesh topology through the optimization. The mesh quality, however, can suffer when the geometry differs too much from the geometry used for the template creation. ATM method, on the other hand, uses an automatic selection from library of topologies and then optimizes the position of the vertices. The mesh is generally bigger, but more likely to maintain high quality for a wide range of geometries.

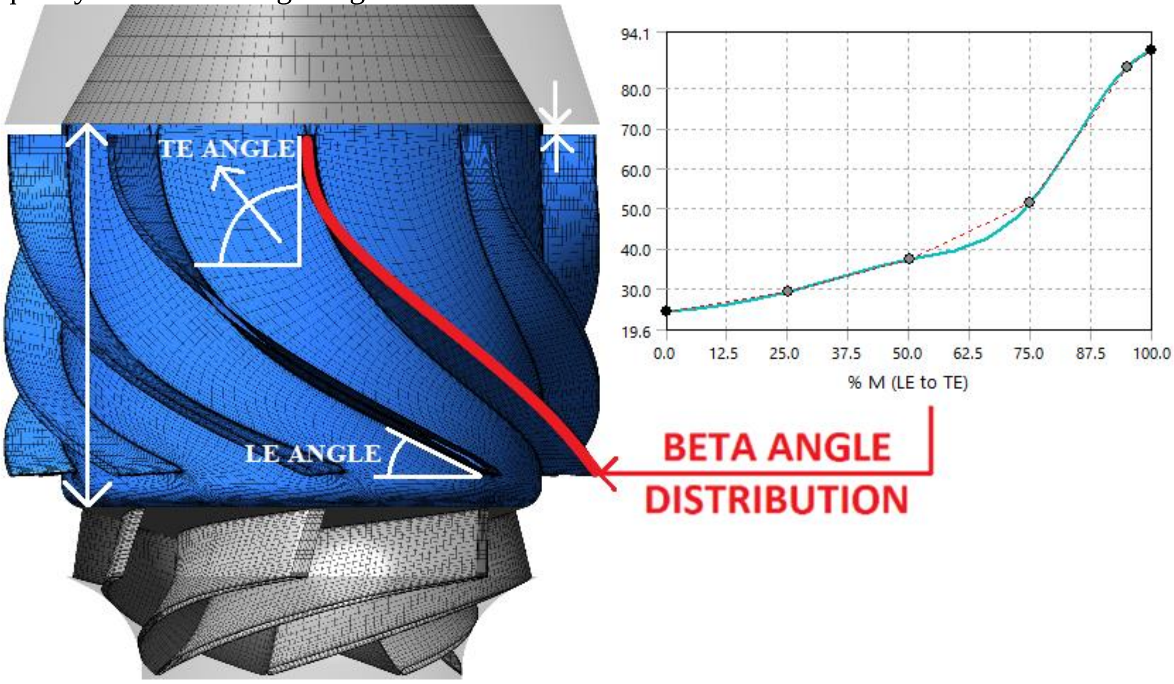

Figure 40: Parameters driving the diffuser shape.

Table 10: Comparison of pre-made topology and ATM meshing in ANSYS TurboGrid

Sample	**Min angle**		**Mean angle**		**Mesh size (number of nodes)**	
1	16.5	40.6	49.7	55.5	8.26E+04	1.19E+05
2	28.3	47.9	65.1	74.9	8.29E+04	1.07E+05
3	24.6	31.6	43.6	50.2	7.29E+04	1.15E+05
4	20.7	21.9	48.3	57.4	8.26E+04	1.17E+05
7	27.0	32.4	51.4	58.0	7.26E+04	1.15E+05
9	20.2	23.0	48.0	60.5	7.51E+04	1.11E+05
10	22.3	45.4	57.9	62.4	8.26E+04	1.08E+05
11	22.0	44.3	58.6	64.5	8.04E+04	1.12E+05
12	21.7	46.2	57.5	64.2	7.26E+04	1.12E+05
14	9.7	9.9	40.1	44.6	7.69E+04	1.12E+05
15	27.6	34.0	59.8	65.0	7.26E+04	1.13E+05
16	27.1	43.7	60.5	67.0	8.04E+04	1.11E+05
17	27.4	39.7	62.7	72.4	7.51E+04	1.11E+05
20	23.6	39.3	65.0	71.6	7.29E+04	1.19E+05
21	19.4	28.3	60.0	69.3	8.01E+04	1.17E+05
22	26.0	44.3	60.7	74.1	7.54E+04	1.10E+05
23	17.5	37.0	66.9	76.0	7.29E+04	1.17E+05
24	31.7	46.3	61.3	70.9	8.26E+04	1.11E+05
25	28.1	32.5	53.2	61.9	8.01E+04	1.09E+05
26	32.7	48.5	61.1	66.1	7.54E+04	1.13E+05
27	27.6	38.1	51.1	57.8	8.29E+04	1.09E+05
29	25.8	31.6	51.0	58.1	7.54E+04	1.12E+05
30	26.5	37.8	50.2	56.6	8.04E+04	1.12E+05
31	25.6	32.6	58.8	65.2	7.29E+04	1.14E+05
32	24.8	32.1	50.5	56.2	8.29E+04	1.12E+05
34	18.3	16.5	49.8	56.8	7.51E+04	1.18E+05
35	31.1	41.9	62.8	68.1	8.29E+04	1.11E+05
36	29.2	34.6	62.7	70.8	7.29E+04	1.08E+05
37	28.6	36.5	50.5	55.3	7.29E+04	1.17E+05
39	19.1	37.1	55.7	66.6	7.51E+04	1.13E+05
40	26.1	36.0	62.1	65.5	8.04E+04	1.13E+05
41	27.2	23.9	51.2	65.1	7.76E+04	1.08E+05
42	22.9	27.8	49.5	58.9	7.51E+04	1.10E+05
43	23.2	45.7	51.6	60.7	7.29E+04	1.17E+05
44	29.3	33.0	66.8	74.3	8.26E+04	1.11E+05
45	23.0	36.6	57.7	66.1	7.29E+04	1.12E+05
46	27.6	32.3	47.3	51.1	7.29E+04	1.11E+05
Mean	**24.6**	**35.4**	**55.7**	**63.2**	**7.72E+04**	**1.13E+05**
difference		*+44.1%*		*+13.5%*		*+45.8%*

As can be seen in Table 10, the ATM method is indeed considerably better in some cases. However, it is at the expense of the mesh size. As the mesh size and simulation times are very important for optimization, the premade-topology approach was preferred in the end. Still, such decision needs to be considered carefully, as in some cases the inferior mesh quality might lead to a situation where the optimization avoids good designs, because the results are skewed by the poor mesh quality.

7.4. Optimization run 1 – Stochastic RBF

Next, the testing was followed by the optimization. For the designs of the initial sampling (created by the “premade-topology” approach), the CFD simulations were run for the three selected flow rates (*76 %*, *100 %* and *120* % of Q_{OPT}). Just as in the previous case, the three objectives were selected as reasonable compromise between computational costs and ensuring good performance (of the pump) in the whole working range. 76 % of Q_{OPT} was placed right below the problematic in-stability flow rate of the original design. After checking these initial results, the optimization was continued with a scalarized objective function defined like:

$$J = max(84 - \eta_{76}, 90 - \eta_{100}, 84 - \eta_{120})$$

The *threshold* values *84*, *90* and *84* (per cents of efficiency) for the respective flow rates were believed to be enough to improve the original design. During the course of the optimization, some modifications were made to the settings. First, some of the parameters seemed to converge to the boundary values – indicating the results could be further improved outside of the originally specified hypercube. As the Stochastic RBF method updates the surrogates every iteration, extending the bounds is relatively straightforward. Of course, it can be assumed that RBF approximation of the extended areas is sub-optimal. Still, for a lack of better information such option was preferred to running a new initial sampling and optimization. Overall, the parameter ranges were modified twice during the course of the optimization. The visualization can be seen in Figure 41.

Further, analysing the available results, the values of the objective *thresholds* were modified, too. The objective function was then changed to:

$$J_{(2)} = max(86 - \eta_{76}, 90 - \eta_{100}, 87.5 - \eta_{120})$$

Also, as an additional information, a manually tuned sample was inserted to the list of computed samples. This can be seen in Figure 42. The sample was added when the hydraulic expert decided to analyse the results of the ongoing optimization, and saw a room for an improvement in the actual best design. This is an example of another major advantage of the Stochastic RBF method (and similar methods in general), when compared to the “classical” gradient-based methods. Here, all available information is always utilized, no matter what source is it obtained from.

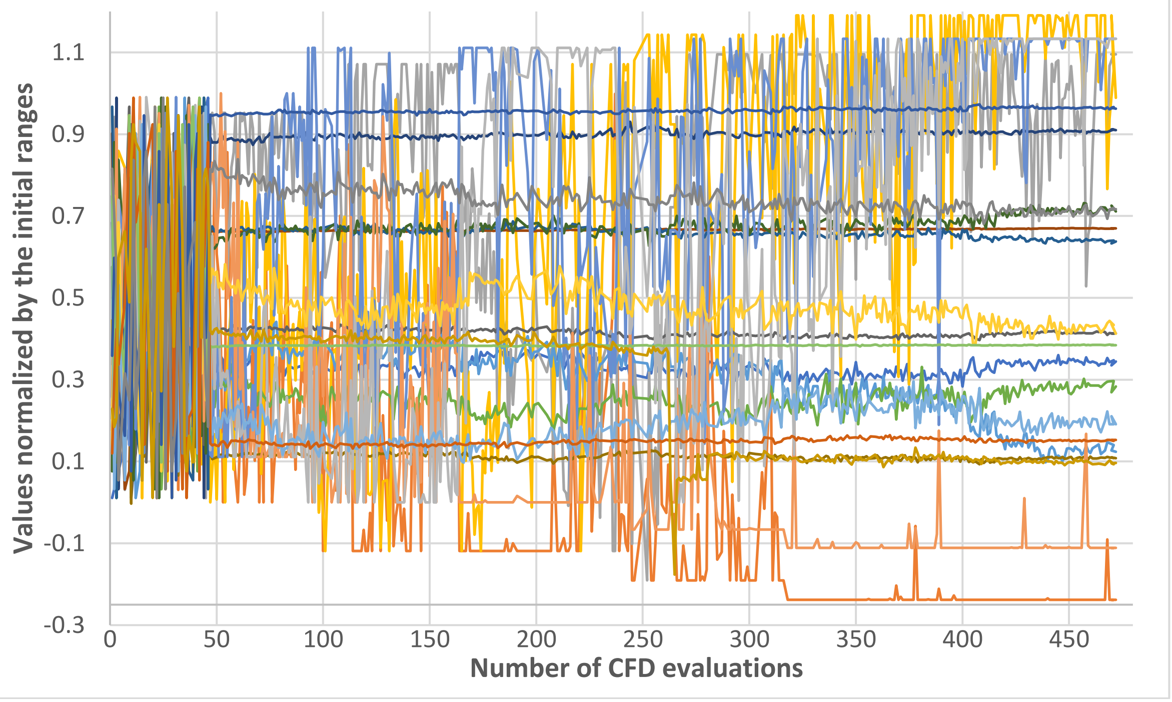

Figure 41: Parameter ranges normalized by the starting values.

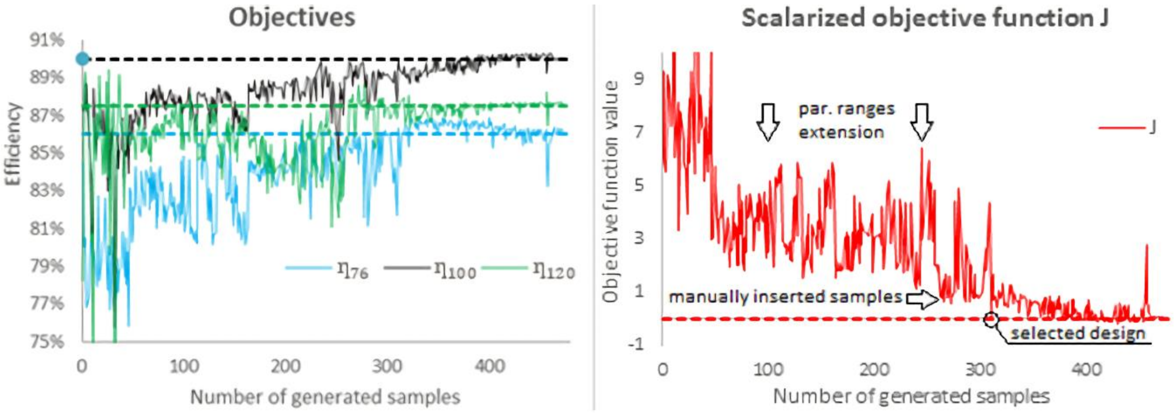

Figure 42: Objectives during optimization. The dashed lines show the optimization thresholds.

The optimization run took many iterations, before the set thresholds were exceeded. Especially improving the efficiency for the lower (sub-optimal) flow rates has proven to be difficult. The progress of all the objectives and of the scalarized objective function can be seen in Figure 42. In Figure 43, the hypervolume[5] of the evaluated samples (including the initial sampling) in the objective space is displayed. It can be clearly observed that the Stochastic RBF optimization managed to further improve on the results of the initial sampling. As the final design, sample number *311* was selected. The sealing rings were added to this model, and efficiency was evaluated at 13 different flow rates in the working range. The simulations were performed for both the full (with the sealing rings) and simplified geometry. The comparison with the baseline design can be seen in Figure 44. There it is also possible to compare the difference caused by the sealing rings. When excluded from the model, the efficiency is mostly increased, as the side-flow from the sealing rings cause whirls and disturbs the flow characters.

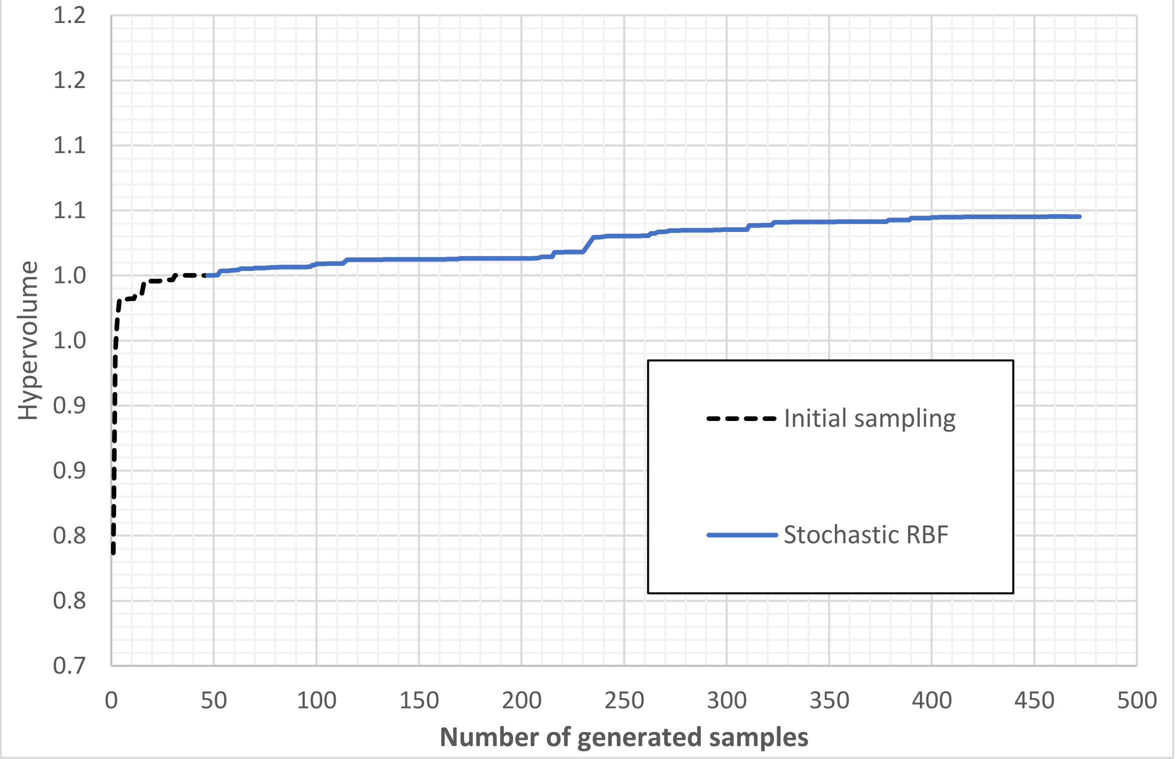

Figure 43: Hypervolume in the objective space. The values are normalized by the hypervolume of the initial sampling.

[5] The *hypervolume*, or *hypervolume indicator*, is commonly used for judging the ability of multiobjective optimization methods to capture a rich set of solutions (78).

The Stochastic RBF optimization took more iterations than expected, but the results were promising. In the end, the optimization managed to meet the objectives defined at the beginning. I.e., to exceed *84 %* of efficiency in the problematic area (*76 %* of Q_{OPT}), and to maintain the high efficiency at Q_{OPT} and at the higher flow rates. The newly found design also dominated the old one, and the more thorough CFD simulations confirmed the results.

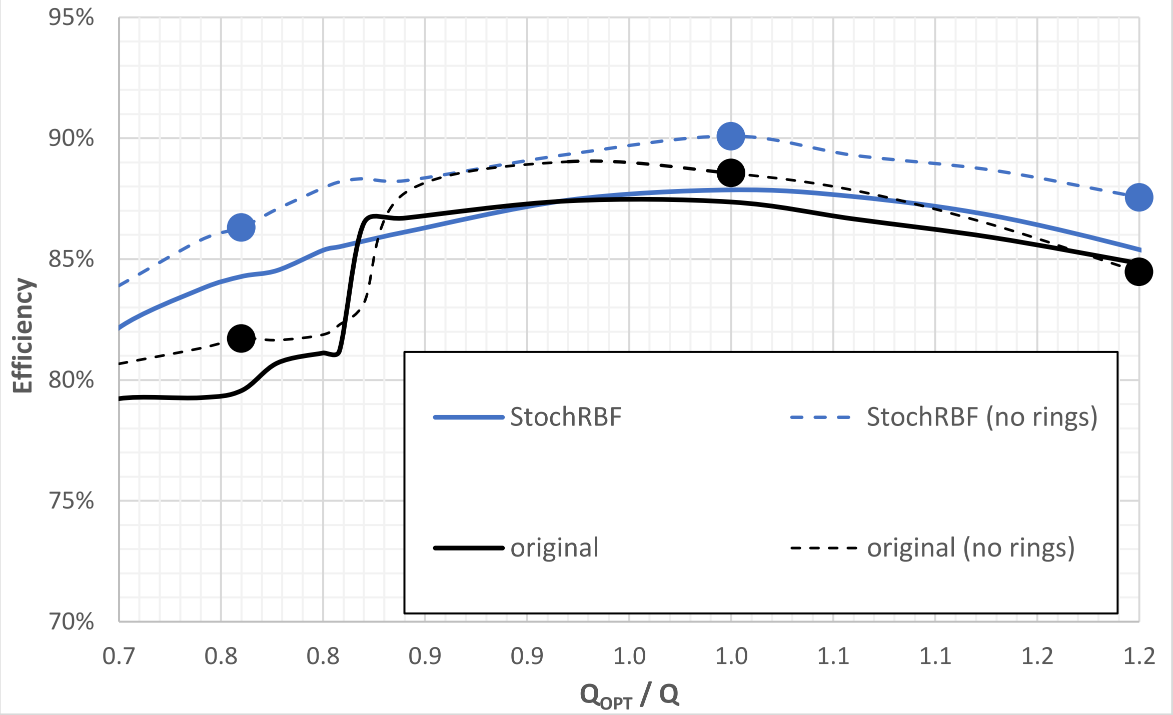

Figure 44: Performance of the original and Stochastic RBF - optimized designs. The dots show optimization objectives.

7.5. Optimization run 2 – K-RVEA

As the next step, a more advanced, truly multiobjective optimization method was used. Kriging-assisted evolutionary multiobjective optimisation (K-RVEA) was introduced in (25). It is an optimization algorithm designed to solve computationally expensive multiobjective optimization problems. It is based on RVEA, introduced in (26). RVEA uses the reference vectors in the objective space in order to guide the population development. The goal of this is to promote exploration in the search, and to ensure an efficient search of the Pareto front. In K-RVEA, this approach is further enhanced by Kriging model for the surrogates, which allows to use the uncertainty information to assess the quality of the approximations and confidence intervals. The objective spaces are approximated (separately for each objective) and RVEA operates on these approximations. This helps to speed up the optimization process, and less computationally expensive evaluations of the objective functions are then required (27).

For a better comparison, the optimization was started from the same initial sampling. In each iteration, there were three new samples. Thanks to its more sophisticated design, K-RVEA displayed superior performance to Stochastic RBF. As can be seen in Figure 47, the hypervolume of objective values (of the design found by the optimization) grows much faster for K-RVEA. This means that K-RVEA is able to guide the search much more efficiently, probably due to more sophisticated way of maintaining the balance between exploitation and exploration, as it uses the reference vectors, i.e., information from the objective space, to its advantage. Stochastic RBF, on the other hand, only works with the information about the samples in the parametric space. Thus, it needs more time to get from a *local-minima trap*. Of course, with the scalarization approach to the objective function it is also more dependent on the scalarization coefficients.

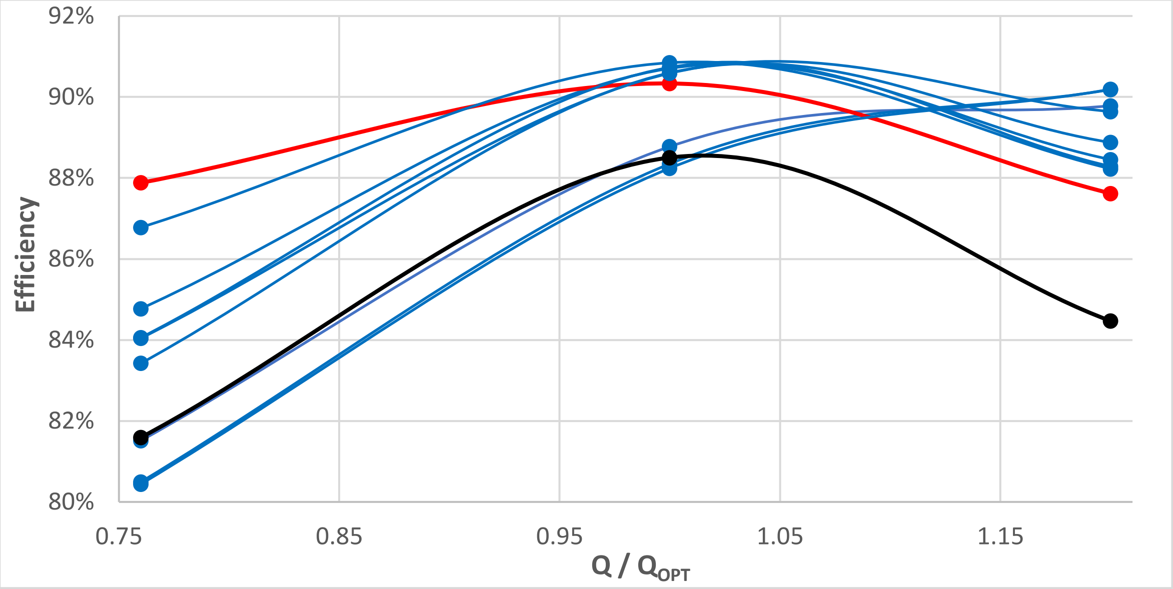

Figure 45: Nondominated solutions found by K-RVEA. The original design is in black, the selected one in red.

For the pump case, the K-RVEA was set to generate three new samples every iteration. As with Stochastic RBF, this was to promote the utilization of the available parallel resources, and also to minimize the problems caused by the failed designs. It took *45* iterations, i.e., *135* samples, before the optimization run was terminated. Out of these 135 samples, 99 were successfully generated and evaluated, the rest has failed. In some iterations it even happened that all three new samples failed, meaning that K-RVEA had to be restarted with a bit different setting a few times. The nondominated solutions found by K-RVEA are shown in Figure 45. Multiple designs displayed performance superior to the old one. In Figure 47, the comparison of hypervolumes from both Stochastic RBF and K-RVEA runs are shown. It can be seen that K-RVEA clearly outperform Stochastic RBF. For the same number of computationally expensive evaluations of the objective function, it is capable of obtaining more information, saving computational resources.

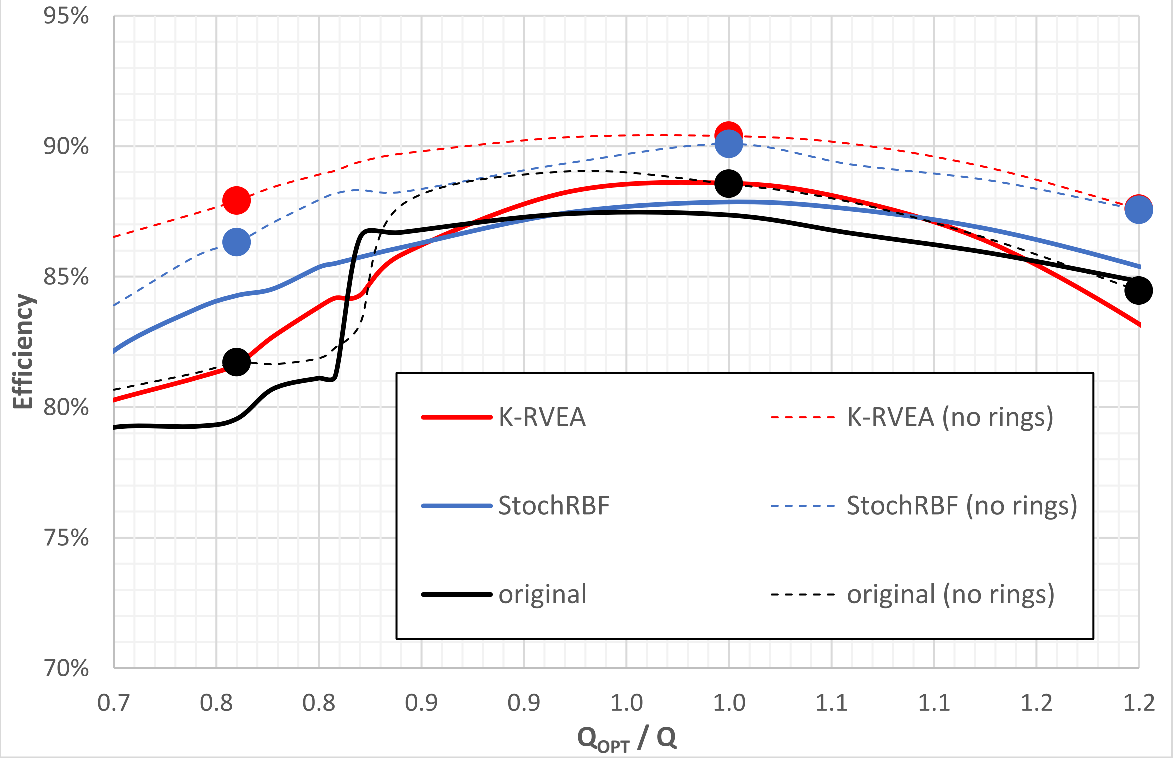

Figure 46: Performance comparison between original and optimized designs. The dots show optimization objectives.

As the main goal of the optimization was to improve the efficiency of the pump in the sub-optimal flow rates (and to at least maintain the efficiency in the remaining parts of the working range), the design with the highest efficiency at *76 %* of Q_{OPT} was selected for further investigation. As far as the results from the simplified optimization CFD model are considered, it clearly dominates the old design. The efficiencies at the two higher flow rates, at Q_{OPT} and at *120 %* of Q_{OPT} are excellent, too.

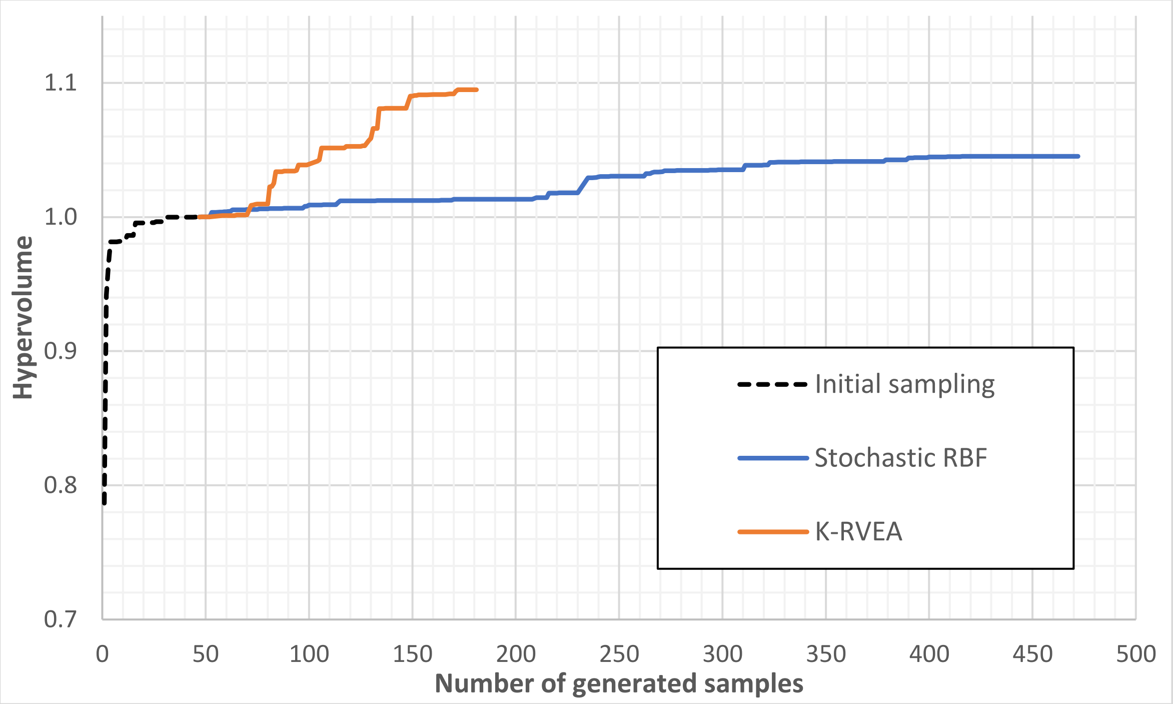

Figure 47: Normalized hypervolume in the objective space - Stochastic RBF vs K-RVEA.

For this design, the complete CFD model, including the sealing rings, was assembled. Next, the full set of CFD simulations for multiple flow rates was performed. The results can be seen in Figure 46. Unfortunately, the results did not meet expectations. The performance with the effect of the sealing rings is considerably worse than expected. While it is still higher than the performance of the original design, the efficiency drop in the sub-optimal flow rates is severe (when compared to the simplified CFD model). While the K-RVEA design dominated the solution found by Stochastic RBF, it does not hold true for the more complex model. The peak efficiency at Q_{OPT} is still higher, but overall, the Stochastic RBF design is preferable. The probable explanation is that that the “side-flow”, caused by the leakage at the impeller inflow and outflow, can disturb the flow in the passage and change the angle, at which the fluid is entering the blades. To further test the performance of the optimization methods, another two larger samplings (with sizes comparable to Stochastic RBF and K-RVEA runs) of the parametric space was created. For one set it was 181 samples - the number of created samples in the K-RVEA run (including the initial sampling). For the larger one, it was 472 samples. Of these, 140 and 390 was successfully created and evaluated on the HPC cluster. The intention was to test whether running the optimization is more efficient than a simple random sampling. The results can be seen in Figure 48. The sampling of the size 172 did not even surpassed the hypervolume of the original sampling of size 46. This is a good example of problems faced when dealing with simulation-based, computationally expensive objective functions. For the heuristic optimization methods, there is always an element of randomness, especially when combined with numerical simulations. Of course, it is not possible to make conclusion from one such comparison. The results are in a good agreement with the expectations, though. Thanks to utilizing additional information, such as the response surface(s), the optimization can easily outperform the random approach.

7.6. Conclusion

The pump performance was successfully improved by shape optimization of the diffuser. Two methods were employed for the task – the previously used (single-objective) Stochastic RBF, and more modern and sophisticated K-RVEA, designed for computationally expensive multi objective optimization. Due to technical limitations, the sealing ring were not considered during the optimization. With this optimization CFD model, K-RVEA displayed superior performance, clearly outperforming Stochastic RBF. Both methods also found an improvement over the original design. However, the original assumption, that the leakage can be ignored for the optimization, has been proven to be incorrect. Instead, the superior (by the simplified CFD model) design, found by K-RVEA, did not perform well once the leakage was added to the model. This clearly demonstrates one of the biggest concerns when dealing with the optimization of the hydraulic shapes. For the lack of better options, the amount if information that can be used for the objective function(s) is limited. I.e., it is not possible to perform a “complete” simulation, with a model that would provide all characteristics relevant for judging the pump performance. Instead, simplifications need to be considered, to make the automation and computational costs acceptable. This on the other hand pose a risk, that the optimized design displays an excellent performance according to the simplified objectives, and poor real-world performance.

Due to this, optimization as a stand-alone tool for hydraulic design is still problematic. The better option is to utilize it to extend the scope of tools hydraulic experts have at their disposal. Overall, this optimization case was still successful, as the baseline design was improved, nonetheless. The K-RVEA method was tested, and displayed performance superior to Stochastic RBF. As it was designed as a truly multiobjective optimization method, it is also capable of finding designs with various trade-offs between the objectives. In our case this means finding pump designs with performance tuned more to lower flow rates and pumps performing better at higher flow rates. Even when doing so, K-RVEA was still able to surpass K-RVEA performance with a lower number of samples.

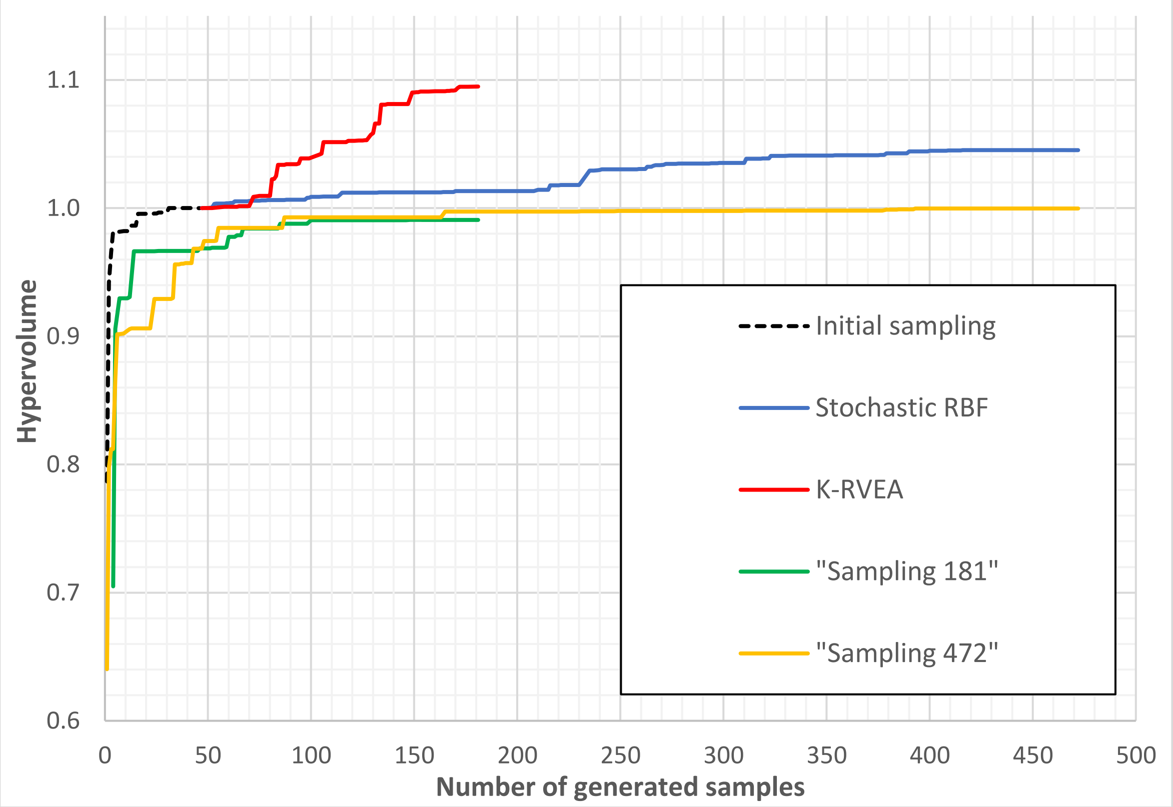

Figure 48: Normalized hypervolume in the objective space - Stochastic RBF vs K-RVEA vs larger sampling.

8. Case 5 - multi-objective optimization of a complete pump

Objectives: Improving the pump performance, meet prescribed head at Q_{OPT}.
Solution: Python codes + premade templates (ANSYS BladeGen, TurboGrid and CFX), transient simulation (full geometry). Six objectives (efficiency at Q_{OPT}, efficiencies at lower flow rates, efficiency at higher flow rates, head at design point and estimation of cavitation performance). DYCORS optimization method + scalarization, K-RVEA.
Results: Hydraulic design performed by "classical" methods was not surpassed.

8.1. Introduction

Based on the previous positive experience with diffuser optimization, it was decided to perform a more difficult optimization of a complete pump. While the basic principles of the optimization remain the same, the complexity of such task is considerably increased. First, the DesignModeler was already at the limit of its capabilities when dealing with a diffuser parametric model. There were problems with creating the geometry and managing the parameters, especially the more complex ones. ANSYS Workbench is also problematic for such tasks – slow, clumsy and prone to errors. The plan was to build a completely new framework based on ANSYS BladeGen and TurboGrid. BladeGen is a very versatile tool for *Turbomachinery*, fast, stable and completely script-based. It is thus very easy to generate complex hydraulic designs and to change the parameters. A considerable advantage is a fact that thanks to the scripts, it is possible to use BladeGen with parameters, that are already a result of executing a complex code. In Workbench this is not quite possible, as the parameters are basically limited to internal scripting capabilities.

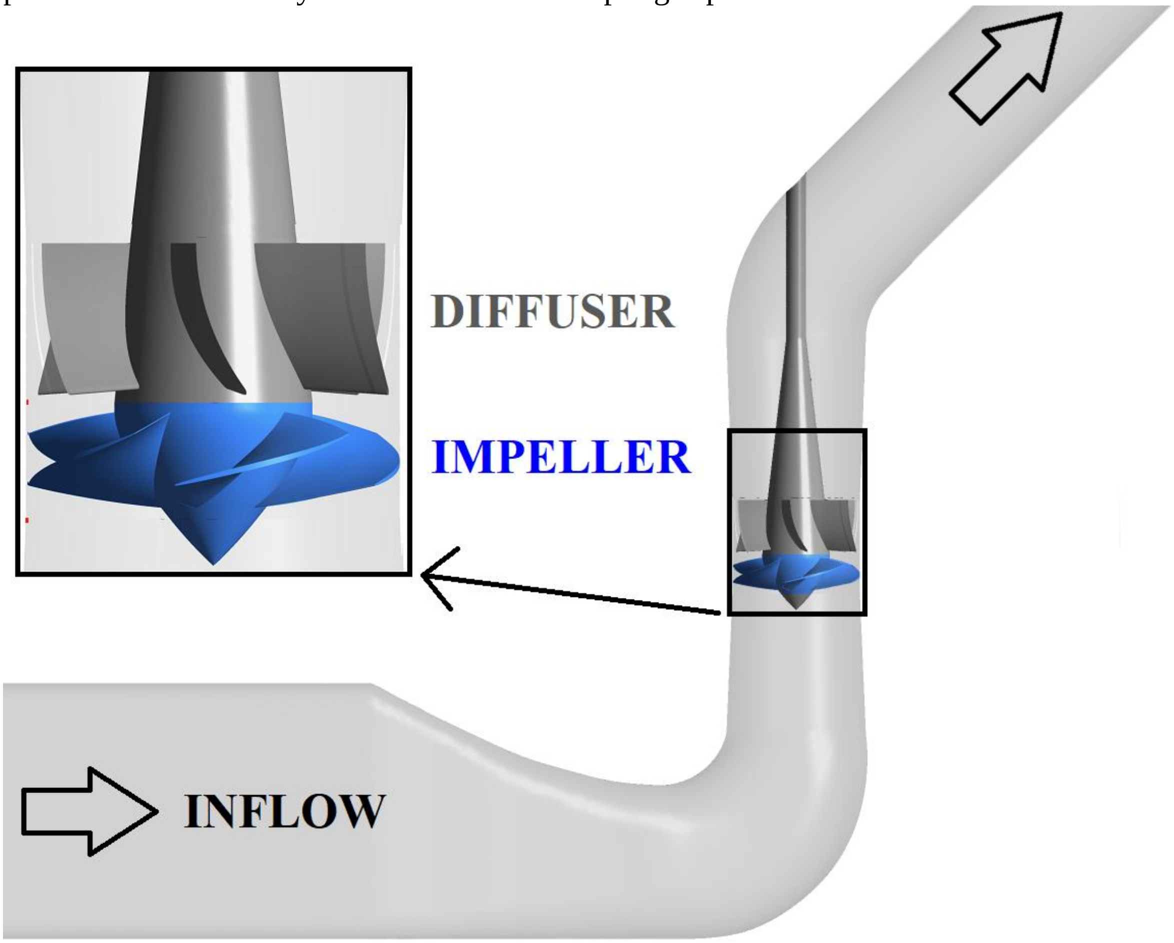

Figure 49: The optimized pump geometry, with a detail of the impeller and diffuser.

For example, when dealing with the beta angle curve, prescribing its shape with expression in Workbench is already a problematic task, prone to errors. But with BladeGen scripts, it is possible to create external codes, responsible for dealing with the optimization parameters, and to manage the geometry creation in more complex and efficient way – smoothing the curves, adding camber lines, managing the number of blades etc. A Python-based framework for managing the automation of whole workflow (geometry and mesh creation, CFD model assembly and computations and results evaluation) also give considerably more freedom in how the optimization problem can be handled.

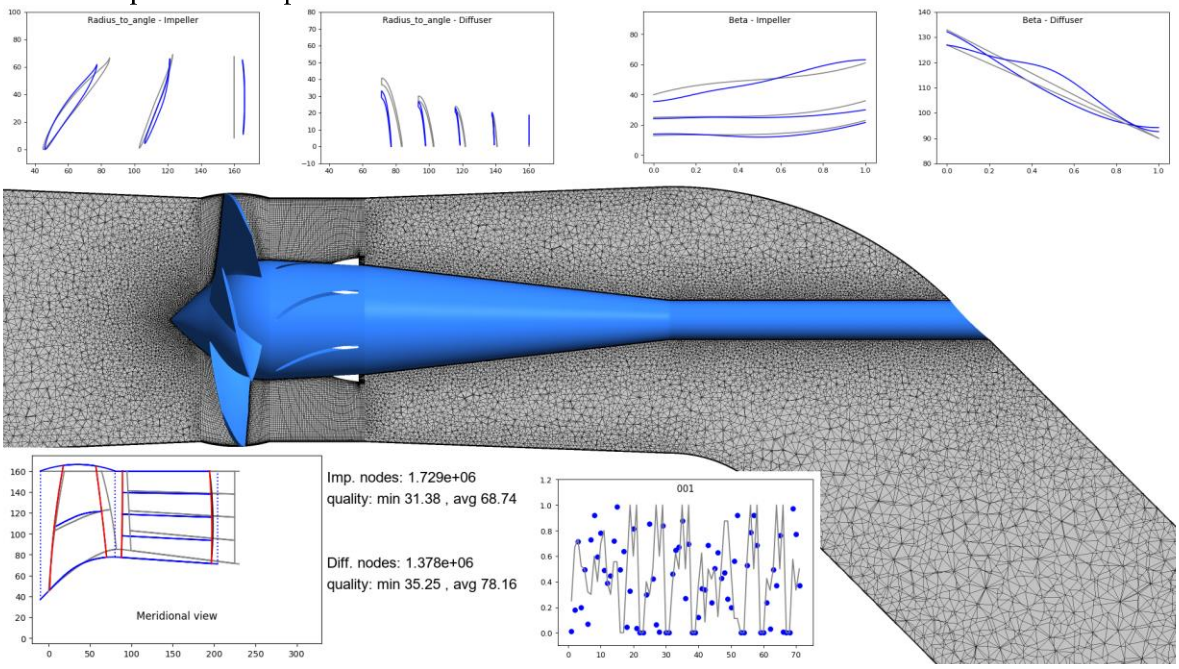

Figure 50: An example of graphical output of the parametric model

An axial pump with diagonal diffuser (specific speed n_S=*680*) was considered for this case. The 3D model of the pump can be seen in Figure 49. The goal was optimizing both rotor and stator, while considering hydraulic performance in multiple flow rates. The remaining parts of the pump remained fixed, to keep the complexity within reasonable limits. Impeller optimization increases the optimization complexity considerably, as – unlike in the stator optimization – the cavitation properties cannot be ignored anymore. As the multiphase head-drop curves simulations (necessary for $NPSH_3$ curve determination) are too computationally expensive, an estimation of cavitation performance, based on the so-called “Blade Loading” (static pressure distribution along the impeller blades) was used. Another challenge is ensuring the head value at the design point Q_{OPT}. This cannot be granted by the parametric model itself, as changing the shapes can have considerable impact on the head value. Still, the pump has to meet the value, and for optimization this poses a challenge. Penalization of the objective function(s), based on (not) meeting the head value, is not well suited for the SAO methods. The penalization changes the objective function, and as a result, it also changes the shape of the response surface. This has a negative effect on the quality of the predictions given by the response surface. Considering the head as an objective is a problematic solution, too, as will be shown later in this chapter.

As part of the solution, a complex parametric model, with a parametric description of both impeller and diffuser, was created. Next, multiple objectives formulations and optimization approaches were used. First, an improved version of Stochastic RBF, named DYCORS, was used. Next, the K-RVEA method was utilized. In the end, the resulting designs are compared against each other and against a classically created hydraulic design (by human expert). In the conclusion, the results are discussed.

8.2. Parametric model and automation

The parametric model was completely new and different from the previous cases, based on premade ANSYS templates and various Python codes.

The model creation works as follows:

1. Generate the meridional section and beta angles distribution (for both rotor and stator).
2. Pass the data to a pre-made BladeGen *bgi* templates and create the inputs for TurboGrid.
3. Generate the meshes calling TurboGrid in batch mode.
4. Call CFX-Pre with a premade model and replace the rotor and stator meshes. Positioning the inflow and outflow meshes is also required in order to match the interfaces. Then create a solver input (.def) file.
5. Call CFD-Post and create graphical / text outputs – meridional section, mesh appearance and quality criterions.

The Python codes allow for a lot of variability. Overall, 71 parameters were used for generating the geometry. Due to various reasons, some of these were kept fixed, and 59 were “active” and used for the optimization. Some of these were passed directly to the BladeGen template, other serve as an input to subroutines. In the next section, a detailed description of all the parameters, follows:

1. Impeller meridional shape:

The goal when creating the hub curve was to have a control over the transition of the shape from inflow to outflow. The geometry was created from boundary angles, relative distance for keeping the angles and one interior point. Shroud curve was created with a semi-adjustable variant of the pump in mind, i.e., as a part of a circle with the appropriate dimension. Three camber lines were used in total, the middle one (defined at 50 % of the Hub-to-Shroud distance) was created as a linear approximation between the hub and shroud. The parameters served as an input for a Python routine, which created the curves.

- “***Impeller_length***” – meridional length of the impeller. As both of the rotor-stator interfaces (inflow-impeller and impeller-diffuser) are created as perpendicular to the axis of rotation, one value, same for hub and shroud, is sufficient.
- “***Impeller_Hub_R_OUT***” – radius of the hub at the impeller outflow, i.e., at the interface between the impeller and the diffuser.
- “***Impeller_Hub_angle_IN***” – hub meridional angle at the impeller inflow, defined to the axis of rotation (i.e., 0°… axial, 90°… radial direction).
- “***Impeller_Hub_hold_IN***” – the distance (defined as a fraction of the hub curve length), for which the inflow angle (defined by *Impeller_Hub_angle_IN*) is held. If this parameter is greater than zero, the appropriate section of the curve is created as linear one, i.e., as a straight line defined by the angle and length.
- “***Impeller_Hub_angle_OUT***” – hub meridional angle at the impeller outflow, defined by a difference from the following diffuser hub angle.
- “***Impeller_Hub_hold_OUT***” – the distance (defined as fraction of the hub curve length), for which the outflow angle (defined by *Impeller_Hub_angle_OUT*) is held. If this parameter is greater than zero, the appropriate section of the curve is created as linear one, i.e., as a straight line defined by the angle and length.
- “***Impeller_Hub_arc_offset***” – additional parameter for driving the hub curve shape. First, the section between the linear inflow and outflow parts is created as a spline defined by two points and two angles. Next, a point is taken in the middle of the spline, and moved by the offset value in a direction perpendicular to the spline. Finally, a new spline is created from the two edge points and angles and from this middle offset point.

2. Impeller blades position and shape:

- "**Impeller_LE_relative_position_Hub**" – leading edge relative position at the hub (defined as a fraction of the hub curve).
- "**Impeller_LE_relative_position_Shroud**" - leading edge relative position at the shroud (defined as a fraction of the shroud curve).
- "**Impeller_LE_arc_offset**" – “curvature” of the leading edge, defined by the arc distance from the straight line in the middle of the curve between hub and shroud.
- "**Impeller_TE_relative_position_Hub**" - trailing edge relative position at the hub (defined as a fraction of the hub curve).
- "**Impeller_TE_relative_position_Shroud**" - trailing edge relative position at the shroud (defined as a fraction of the shroud curve).
- "**Impeller_TE_arc_offset**" - “curvature” of the trailing edge, defined by the arc distance from the straight line in the middle of the curve between hub and shroud.
- "**Impeller_HubLE_EllipseRatio**" – aspect ratio of the ellipse driving the shape of the leading edge at the hub.
- "**Impeller_ShrLE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the leading edge at the shroud.
- "**Impeller_HubTE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the trailing edge at the hub.
- "**Impeller_ShrTE_EllipseRatio**" – aspect ratio of the ellipse driving the shape of the trailing edge at the shroud.

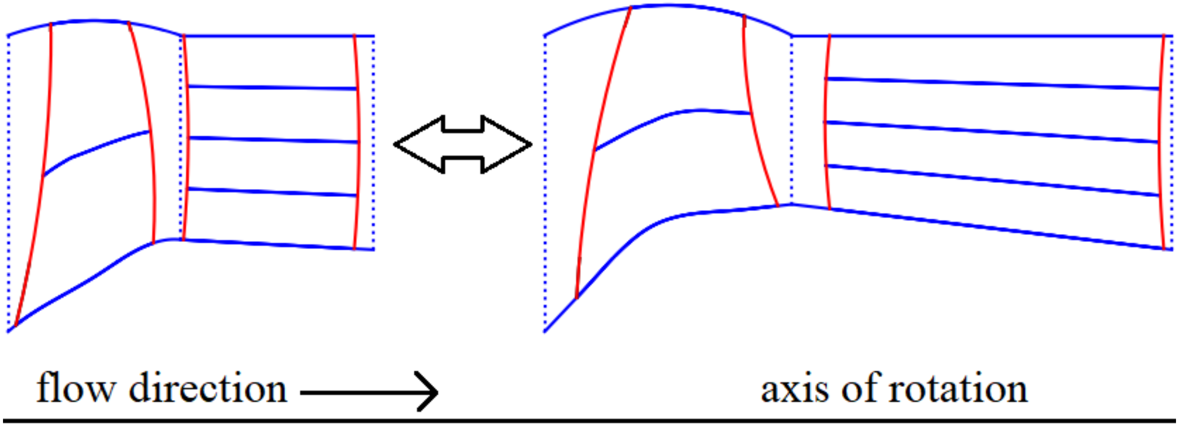

Figure 51: Visualisation of selected parameters (meridional shapes).

The shape of the impeller blades in described by the angle, defined at three different camber lines – hub, shroud, and in the middle. Each of these three curves is defined by end point, constant segments and a segment defined by four points and derivatives. The goal was to maintain a smooth transition between leading and trailing edges, and to have control over the character of the change. Some examples are given in Figure 53. The values at 25 % and 50 % of the curve length are important for the pressure generation, i.e., related to the value of head. The beta angle curve is generated from the parameters by a separate Python code, and inserted to the BladeGen script.

- "**Impeller_BetaHubLE**" – beta angle of the impeller blades, defined for the hub curve at the leading edge.
- "**Impeller_BetaHubTE**" - beta angle of the impeller blades, defined for the hub curve at the trailing edge.
- "**Impeller_BetaHubLE_derivative**" – coefficient of derivative of the hub beta angle curve at the leading edge. Its value can be between 0 and 1, When creating the appropriate segment of the curve, first a spline is fit to the four already known points.

Next, the value of derivative at the end points is multiplied by the appropriate coefficients, and a new spline is created.

- "**Impeller_BetaHubTE_derivative**" - coefficient of derivative of the hub beta angle curve at the trailing edge.
- "**Impeller_holdBetaHubLE**" – length of the straight segment at leading edge, where the angle value is held constant. Its value is considered as relative to the length of the whole curve.
- "**Impeller_holdBetaHubTE**" - length of the straight segment at trailing edge, where the angle value is held constant
- "**Impeller_diffBetaHub25**" – increase of beta angle at 25 % (of curve length) relative to the value at leading edge.
- "**Impeller_diffBetaHub50**" – a value of beta angle “added” to the curve at 50 % of the curve length. When using a negative value, this control point helps creating the desired “saddle” shape of the curve
- "**Impeller_BetaMidLE**" - beta angle of the impeller blades, defined for the middle curve at the leading edge.
- "**Impeller_BetaMidLE_derivative**" - coefficient of derivative of the middle beta angle curve at the leading edge.
- "**Impeller_BetaMidTE**" - beta angle of the impeller blades, defined for the middle curve at the trailing edge.
- "**Impeller_BetaMidTE_derivative**" - coefficient of derivative of the middle beta angle curve at the trailing edge.
- "**Impeller_holdBetaMidLE**" - length of the straight segment at leading edge, where the angle value is held constant. Its value is considered as relative to the length of the whole curve.
- "**Impeller_holdBetaMidTE**" - length of the straight segment at trailing edge, where the angle value is held constant.
- "**Impeller_diffBetaMid25**" - increase of beta angle at 25 % (of curve length) relative to the value at leading edge.
- "**Impeller_diffBetaMid50**" - a value of beta angle “added” to the curve at 50 % of the curve length.
- "**Impeller_BetaShroudLE**" - beta angle of the impeller blades, defined for the shroud curve at the leading edge.
- "**Impeller_BetaShroudLE_derivative**" - coefficient of derivative of the shroud beta angle curve at the leading edge.
- "**Impeller_BetaShroudTE**" - beta angle of the impeller blades, defined for the shroud curve at the trailing edge.
- "**Impeller_BetaShroudTE_derivative**" - beta angle of the impeller blades, defined for the shroud curve at the trailing edge.
- "**Impeller_holdBetaShroudLE**" - length of the straight segment at leading edge, where the angle value is held constant. Its value is considered as relative to the length of the whole curve.
- "**Impeller_holdBetaShroudTE**" - length of the straight segment at trailing edge, where the angle value is held constant.
- "**Impeller_diffBetaShroud25**" - increase of beta angle at 25 % (of curve length) relative to the value at leading edge.
- "**Impeller_diffBetaShroud50**" - a value of beta angle “added” to the curve at 50 % of the curve length.
- "**Impeller_SweepMid_relative**" – Relative value of the sweep-angle for the midline, used for a linear interpolation (0 for hub, 1 for shroud).

- "**Impeller_SweepShroud**" – The so-called *sweep angle* defined for the blade at the shroud. I.e., the *circumferential* angle between the blade leading edge position at the hub and at the shroud.

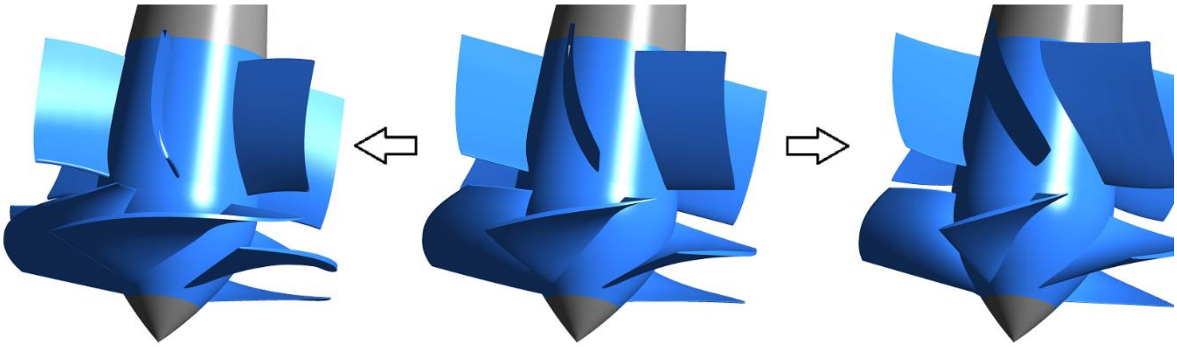

Figure 52: Impeller and diffuser blades for various settings of the parameters. The parametrized part of the geometry is displayed in blue.

- "**Diffuser_length**" – Meridional length of the diffuser, i.e., a distance between *Impeller – Diffuser* and *Diffuser – Outflow* interfaces.
- "**Diffuser_LE_relative_position_Hub**" - leading edge relative position at the hub (defined as a fraction of the hub curve).
- "**Diffuser_LE_relative_position_Shroud**" - leading edge relative position at the shroud (defined as a fraction of the shroud curve).
- "**Diffuser_LE_arc_offset**" - "curvature" of the leading edge, defined by the arc distance from the straight line in the middle of the curve between hub and shroud.
- "**Diffuser_TE_relative_position_Hub**" - trailing edge relative position at the hub (defined as a fraction of the hub curve).
- "**Diffuser_TE_relative_position_Shroud**" - trailing edge relative position at the shroud (defined as a fraction of the shroud curve).
- "**Diffuser_TE_arc_offset**" – "curvature" of the trailing edge, defined by the arc distance from the straight line in the middle of the curve between hub and shroud.
- "**Diffuser_HubLE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the leading edge at the hub.
- "**Diffuser_ShrLE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the leading edge at the shroud.
- "**Diffuser_HubTE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the trailing edge at the hub.
- "**Diffuser_ShrTE_EllipseRatio**" - aspect ratio of the ellipse driving the shape of the trailing edge at the shroud.
- "**Diffuser_BetaHubLE**" - beta angle of the diffuser blades, defined for the hub curve at the leading edge.
- "**Diffuser_BetaHubLE_derivative**" - coefficient of derivative of the hub beta angle curve at the leading edge.
- "**Diffuser_BetaHubTE**" - beta angle of the diffuser blades, defined for the hub curve at the trailing edge.
- "**Diffuser_BetaHubTE_derivative**" - coefficient of derivative of the hub beta angle curve at the trailing edge.
- "**Diffuser_holdBetaHubLE**" - length of the straight segment at leading edge, where the angle value is held constant.
- "**Diffuser_holdBetaHubTE**" - length of the straight segment at trailing edge, where the angle value is held constant.
- "**Diffuser_diffBetaHub25**" - increase of beta angle at 25 % (of curve length) relative to the value at leading edge.

- "**Diffuser_diffBetaHub50**" - a value of beta angle “added” to the curve at 50 % of the curve length.
- "**Diffuser_BetaShroudLE**" - beta angle of the diffuser blades, defined for the shroud curve at the leading edge.
- "**Diffuser_BetaShroudLE_derivative**" - coefficient of derivative of the shroud beta angle curve at the leading edge.
- "**Diffuser_BetaShroudTE**" - beta angle of the diffuser blades, defined for the shroud curve at the trailing edge.
- "**Diffuser_BetaShroudTE_derivative**" - beta angle of the impeller blades, defined for the shroud curve at the trailing edge.
- "**Diffuser_holdBetaShroudLE**" - length of the straight segment at leading edge, where the angle value is held constant.
- "**Diffuser_holdBetaShroudTE**" - length of the straight segment at trailing edge, where the angle value is held constant.
- "**Diffuser_diffBetaShroud25**" - increase of beta angle at 25 % (of curve length) relative to the value at leading edge.
- "**Diffuser_diffBetaShroud50**" - a value of beta angle “added” to the curve at 50 % of the curve length.
- "**Diffuser_SweepShroud**" – Sweep angle of the blade, defined at the shroud for the leading edge.

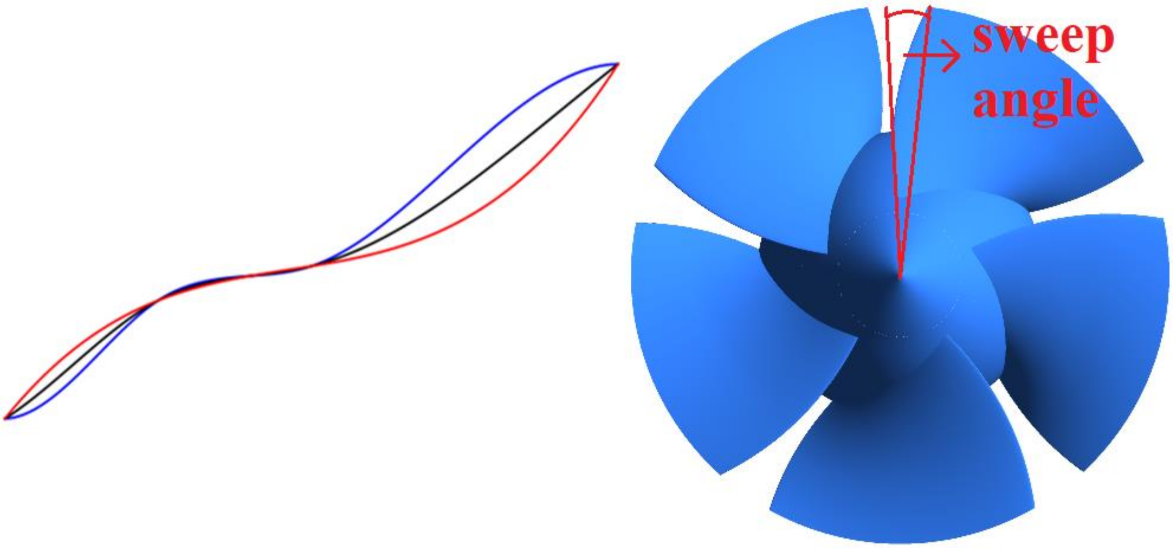

Figure 53: Visualisation of blade angle curves for different parameter settings (left) and sweep angle definition (right).

Stator hub and shroud meridional shape is defined as a straight line connecting two points. The outlet shroud diameter is fixed, to respect the outflow pipe diameter.

Beta angle distribution is generated from following parameters:

- Starting and ending angles. Optimal values of these angles are known from the hydraulic theory. They are decided from the velocity triangles.
- Distances of “holding” the starting and ending angles (“zero” length, i.e., no segment with constant angle, is also possible).
- Spline connecting the starting and ending segments. It is prescribed by values and derivatives at the edges, and values at 25% and 50% of the length. This helps obtaining the desired “saddle” shape of the curve.

The angles are generated at three streamlines (hub, middle and shroud) for the impeller blades and at two streamlines (hub and shroud) for the diffuser blades.

Other parameters were for the impeller blade “sweep” (middle and shroud) and meridional positions of leading and trailing edges of the blades.

An example of a comprehensive visualization of a generated geometry, created automatically by the framework, is in Figure 50. It displays slice of the CFD model, meridional sections, blade views and beta angles (for both rotor and stator), mesh information and normalized (in the objective space hypercube) parameter values – in comparison with a reference design. Such visualization serves for a quick inspection of the parametric model.

Table 11´Parameter ranges. Grey denotes parameters with value fixed during the optimization.

Name	**min**	**max**
Impeller_length	90	130
Impeller_Hub_R_OUT	75	90
Impeller_Hub_angle_IN	35	40
Impeller_Hub_hold_IN	0	25
Impeller_Hub_angle_OUT	-5	5
Impeller_Hub_hold_OUT	0	25
Impeller_Hub_arc_offset	-5	5
Impeller_LE_relative_position_Hub	0.05	0.15
Impeller_LE_relative_position_Shroud	0.25	0.35
Impeller_LE_arc_offset	-5	5
Impeller_TE_relative_position_Hub	0.85	0.95
Impeller_TE_relative_position_Shroud	0.7	0.8
Impeller_TE_arc_offset	-5	5
Impeller_HubLE_EllipseRatio	1	10
Impeller_ShrLE_EllipseRatio	1	10
Impeller_HubTE_EllipseRatio	1	3
Impeller_ShrTE_EllipseRatio	1	3
Impeller_BetaHubLE	35	45
Impeller_BetaHubLE_derivative	0	1
Impeller_BetaHubTE	55	65
Impeller_BetaHubTE_derivative	0	1
Impeller_holdBetaHubLE	0	
Impeller_holdBetaHubTE	0	
Impeller_diffBetaHub25	5	10
Impeller_diffBetaHub50	-4	0
Impeller_BetaMidLE	20	30
Impeller_BetaMidLE_derivative	0	1
Impeller_BetaMidTE	30	40
Impeller_BetaMidTE_derivative	0	1
Impeller_holdBetaMidLE	0	
Impeller_holdBetaMidTE	0	
Impeller_diffBetaMid25	-1	3
Impeller_diffBetaMid50	-4	0
Impeller_BetaShroudLE	10	16
Impeller_BetaShroudLE_derivative	0	1
Impeller_BetaShroudTE	20	26
Impeller_BetaShroudTE_derivative	0	1

Impeller_holdBetaShroudLE	0	
Impeller_holdBetaShroudTE	0	
Impeller_diffBetaShroud25	-1	3
Impeller_diffBetaShroud50	-4	0
Impeller_SweepMid_relative	0.2	0.8
Impeller_SweepShroud	0	16
Diffuser_length	100	200
Diffuser_LE_relative_position_Hub	0.02	0.1
Diffuser_LE_relative_position_Shroud	0.02	0.1
Diffuser_LE_arc_offset	-5	5
Diffuser_TE_relative_position_Hub	0.9	0.98
Diffuser_TE_relative_position_Shroud	0.9	0.98
Diffuser_TE_arc_offset	-5	5
Diffuser_HubLE_EllipseRatio	1	10
Diffuser_ShrLE_EllipseRatio	1	10
Diffuser_HubTE_EllipseRatio	1	
Diffuser_ShrTE_EllipseRatio	1	
Diffuser_BetaHubLE	127	137
Diffuser_BetaHubLE_derivative	0	1
Diffuser_BetaHubTE	85	95
Diffuser_BetaHubTE_derivative	0	1
Diffuser_holdBetaHubLE	0	
Diffuser_holdBetaHubTE	0	
Diffuser_diffBetaHub25	-15	-5
Diffuser_diffBetaHub50	-2	4
Diffuser_BetaShroudLE	122	132
Diffuser_BetaShroudLE_derivative	0	1
Diffuser_BetaShroudTE	85	95
Diffuser_BetaShroudTE_derivative	0	1
Diffuser_holdBetaShroudLE	0	
Diffuser_holdBetaShroudTE	0	
Diffuser_diffBetaShroud25	-15	-5
Diffuser_diffBetaShroud50	-2	4
Diffuser_SweepShroud	-5	5

In Table 11, all the 71 parameters used for creating the geometry are listed, including their minimal and maximal values. As was mentioned in the previous text, only some of these parameters are passed directly to the BladeGen script. Instead, most of them serve as an input for separate Python routines, responsible for creating the BladeGen input data. There are routines for impeller meridional curves, diffuser meridional curves and blade angle beta curve. Another routine ensures the necessary displacement of the fixed hydraulic parts (inflow and outflow), so that their position in the assembled solver input file matches the position of the created impeller and diffuser. The Python codes also give a lot of flexibility for managing the parameters. The geometry is created from all the available 71 parameters, but some of them are simply marked as "*inactive*". When the script finds an inactive parameter, it simply uses a pre-set default value, instead of considering it among the optimized parameters.

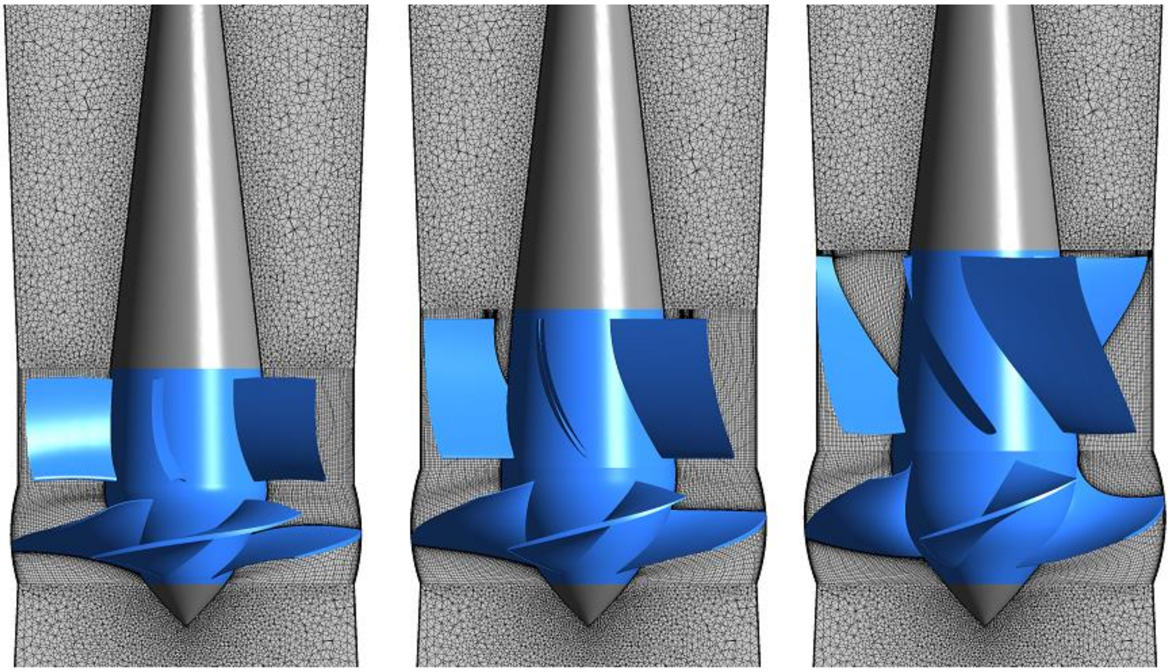

Figure 54: Examples of generated designs, including a slice through the computational mesh, for three different parameters settings. The parametrized parts of the pump are in blue.

8.3. Initial sampling and objectives

The “baseline” hydraulic design, manually created by an experienced hydraulic expert, was used for setting the parametric model. First, the selection of active parameters was set (after a discussion with the hydraulic expert) according to this design.

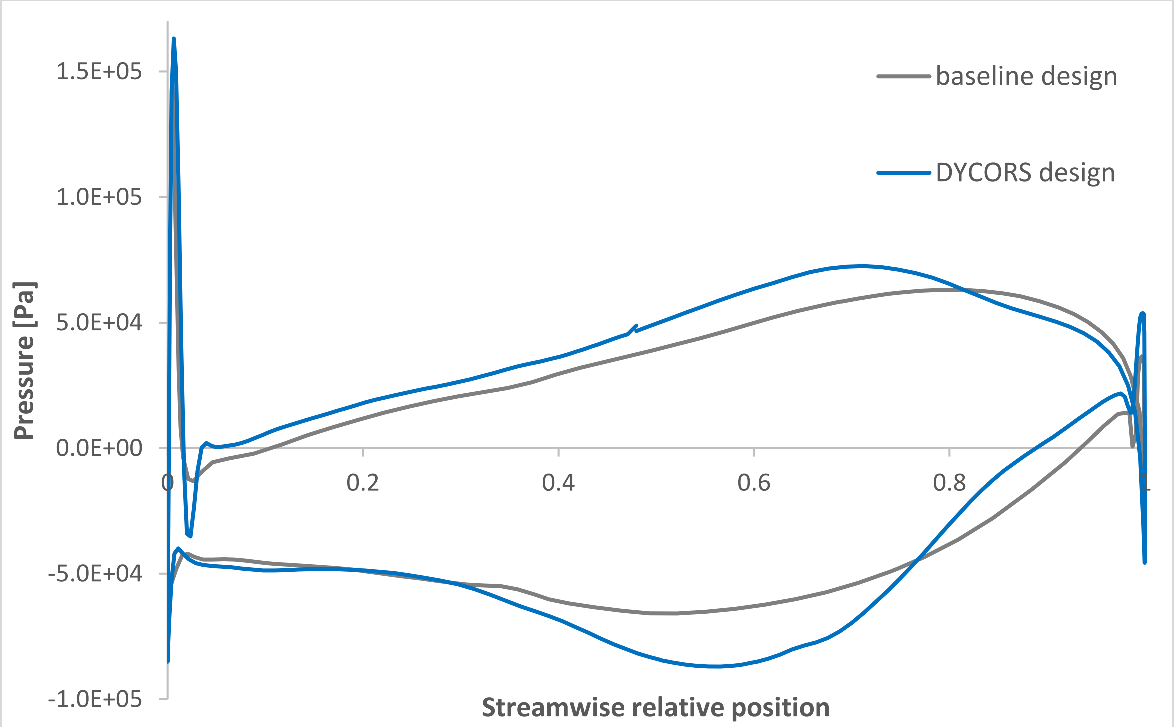

Figure 55: Optimization results. Comparison of BL at 110 % of Q_{OPT}.

Then, lower and upper bounds were decided as reasonable offsets to the values used for the baseline design. For the start, the initial sampling was generated. Its size was decided by the *2 * (N + 1)* formula, same as in the previous cases. For 59 active parameters this means 120 samples. Out of these 120 samples, 3 failed for various geometry reasons, and another 2 had a very poor mesh quality. In total there were 115 samples left for the expensive CFD evaluation. Every sample was evaluated in five pre-set flow rates – *60*, *70*, *80*, *100* and *120* % of Q_{OPT}. The intention was to perform a more thorough analysis than in the diffuser's case, so five flow rates instead of three were considered. The simulations were run on an HPC cluster. To capture the BL transient behaviour, it was necessary to analyse the data for every single timestep of the run. For this mesh sizes, the trn file, containing the intermediate results, has ca. 1 GB per timestep. A typical run requires approximately 2000 timesteps, thus almost 2 TB od data had to be processed for each design and for each flow rate. To avoid excessive data transfers, the results were processed on-the-fly, with only csv files with values being transferred.

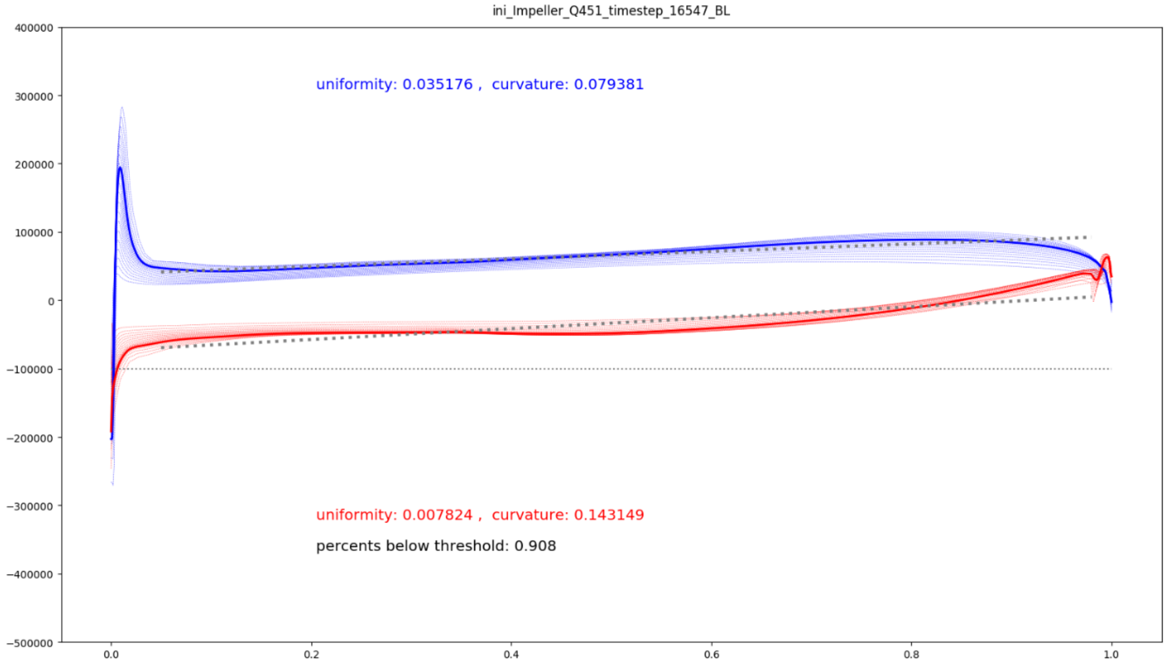

Figure 56: Blade loading plot example.

An example of the Bl data can be seen in Figure 56. The LE to TE direction is displayed from left to right, red colour is for the suction side and blue for pressure side. The pressure data were processed at 19 layers (0.05, 0.10, … of Hub-to-Shroud distance) and averaged to a single curve, with weights 0.05, 0.10, … This reflects the fact that the flow character at the shroud is more important for the overall performance, as the velocity increases with the increasing distance to the axis of rotation. Another technical problem was transferring the large number of files, related to the BL values. In CFD-Post, values for each of these 19 layers can only be extracted into a separate csv file. When considering 19 files for both rotor and stator, and for every design, flow rates and time steps, hundreds of thousands of files had to be transferred every two hours – putting a pressure on the network connection. To solve this problem, the csv files had to be packed into a single archive prior to data transfer. This is just an example of problems faced when solving a real-world, computationally expensive, simulation-driven optimization problem – something, that is often not considered in theory.

For practical reasons, the CFD runs were split into a sequence of shorter 2-hours runs, continued "until ending criterion" ("convergence" or maximum number of timesteps). At the end of each 2-hours run, the BL data (for both rotor and stator and every timestep) were extracted to csv, packed (to prevent unnecessary load on the file servers) and sent to the central storage. Once the CFD simulations for the initial sampling was finished, the results were analysed. There were multiple objectives to consider and evaluated, to decide the formulation of objectives later:

1. The peak efficiency, respective the efficiency at the design point. (The actual optimum can occur at a different flow rate.)
2. The overall efficiency at the working range.
3. The *NPSH*$_3$ (cavitational) properties in the working range.
4. Meeting the desired head at the design point.

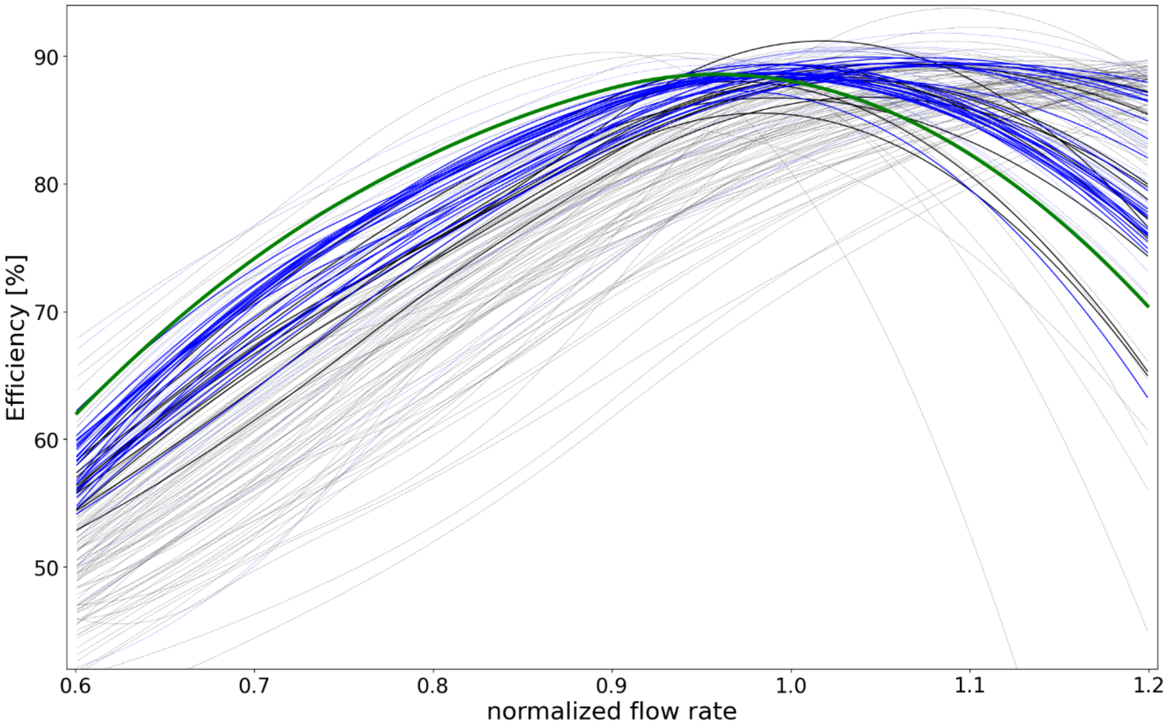

Figure 57: Efficiency curve approximated from five points (flow rates). The initial sampling is in black, DYCORS in blue and the baseline design in green. The lines are bolder for design that are within the tolerance for the design head.

Evaluating the efficiencies and head is trivial. The cavitation properties, however, are more challenging, as the correlation between *NPSH*$_3$ and BL properties is not completely known. Based on hydraulic experts' opinions and (2), three main measurable qualities of the BL shape were considered:

- The percentage of the curve where the pressure drops below the water vapour pressure (3169 Pa). In our case the pump was designed to operate at water level, i.e., at the atmospheric pressure. Thus, the threshold was set to approximately -1e5 Pa. The pressure drop occurs at the leading edge, and the designs with a good *NPSH*$_3$ typically have this peek very narrow. The problem with this criterion is that once the cavitation develops, the BL changes. Thus, this metric only gives a hint about the cavitational behaviour up to a point where the cavitation starts to develop.
- A "linearity" of the curve at the main passage, i.e., between the leading edge and trailing edge. The total pressure difference between TE and LE is basically given (by the head), but the distribution of the change between these two boundary points should optimally be as uniform as possible. To measure this quality by a single numerical value, the Pearson correlation coefficient was used. First, the inner part of the blade is identified in the BL data. Next, a linear fit to this part is created using *Least squares method*. Then the correlation between the selected inner part and the linear curve is measured by the Pearson correlation coefficient.
- The "transient" stability of the BL curves. This one is supposed to be related to the stability of other hydraulic characteristics (efficiency, head), which is desirable in general. It was evaluated as Pearson correlation coefficient between BL curves for each subsequent pair of timesteps. For *N* timesteps, the resulting output was a curve

containing $N - 1$ values. This was evaluated as part of the testing, but it is already correlated with the other available characteristics.

For all available data from the initial sampling, the efficiency and head values and BL properties were evaluated and examined. Understanding the data was important for the following part, to decide the importance of various characteristics and to guide the process of formulation of the objectives for optimization purposes:

8.4. Optimization – DYCORS

For the first optimization run and testing, the single-objective DYCORS (28) method was used. This method is an improvement over its predecessor, Stochastic RBF, and uses a more sophisticated approach for generating the *Candidate points*. Instead of variating all available parameters, it continually narrows the search as the optimization progresses. According to the numerical testing, it is more efficient for higher dimensions, especially if the number of parameters exceeds 30. Based on the results obtained from the initial sampling data, the considered objectives were:

1. Efficiency at the design point - the goal was to ensure meeting efficiency objective at the design point Q_{OPT}.
2. Efficiency curve – multiple flow rates were considered to ensure a good performance at the whole working range. The objective was based on a sum of their values.
3. BL curve – the percentage of the curve where the pressure drops below the water vapour pressure and the linearity of the BL curve at the suction side of the blade.
4. Difference from the head (at the design point) – this was to ensure that the optimized design meets the required value of head, instead of finding a geometry with great efficiency, but with an unacceptable value of head.

For the scalarization, the approach recommended in (24) was used. For each objective Ji, its value is normalized as

$$J_{i\_norm} = \frac{J_i - J_{i\_threshold}}{J_i - J_{i\_utopian}}$$

Where $J_{i\_threshold}$ is a selected threshold value, and $J_{i\_utopian}$ an ideal (but unreachable) value. The scalarized objective function is then assembled as

$$J_{total} = \max_{i=1,\ldots,N} J_{i\_norm} + \sum_{i=1,\ldots,N} \frac{J_{i\_norm}}{"large\ number"}$$

The “*large number”* can be selected arbitrarily, but such, that the sum is relatively small when compared to the *max* part of the formula. Thanks to this, the scalarized objective can capture changes to objectives that are lost in the *max*, and the desired properties of the scalarization are still preserved. This way it is possible to capture the Pareto front more efficiently. The choice of the threshold and utopian values influence the “weights” of the objectives. In our case, the results of the initial hydraulic design were used for the thresholds. The objective functions and the scalarized value are shown in Figure 58. This approach was used for testing purposes and for a lack of better options (with the DYCORS method). However, it can be assumed the scalarization of multiple heterogenous objectives is problematic for the response surface quality.

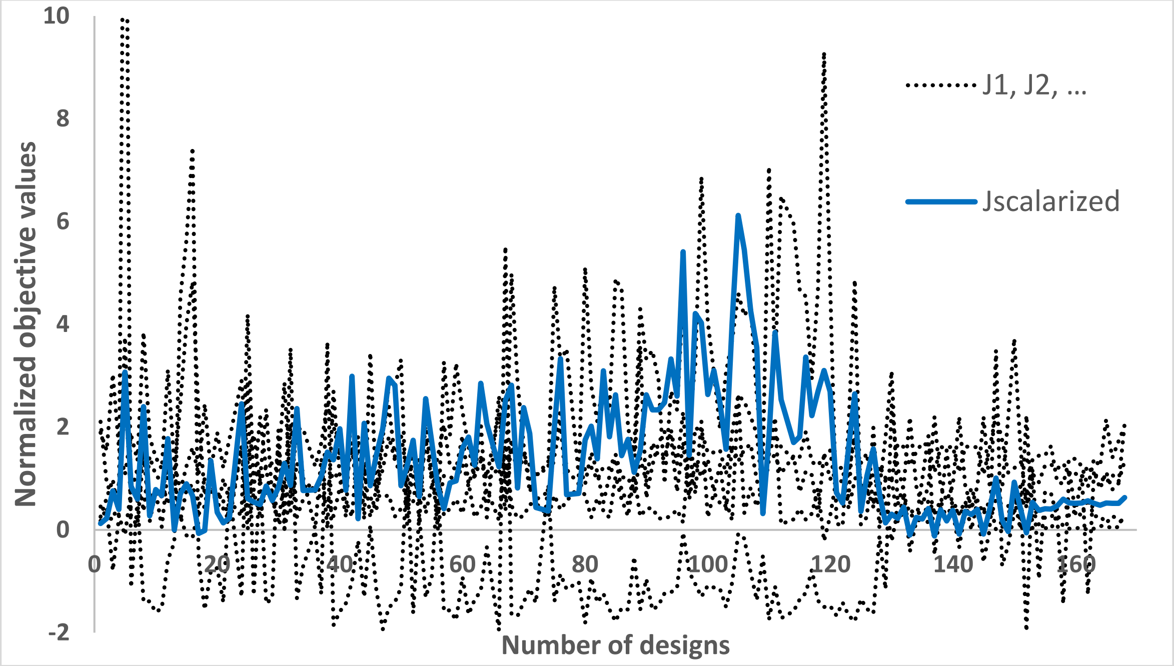

Figure 58: Objective functions and the scalarized objective function for DYCORS.

The optimization run was initiated from the data obtained from the initial sampling. To improve the exploration, the method was set differently from previous case. Instead of generating multiple samples at once and running them in parallel, there were four independent "*instances*" of DYCORS optimization, operating on the same data, but with different objective functions. The first objective was for the maximal efficiency (at Q_{OPT}), the second one was working with the whole efficiency curve, the third one used the scalarized value of the two BL metrics, and for the last one, the scalarization of the first three objectives was available. Every DYCORS "*instance*" was set to generate a single new sample in each iteration, i.e., depending on the failed design, there were up to four new samples to evaluate per iteration. To guide all the optimizations for meeting the design head, it was necessary to penalize the first three objectives by the difference from the desired value. Such solution was expected to improve the optimization results.

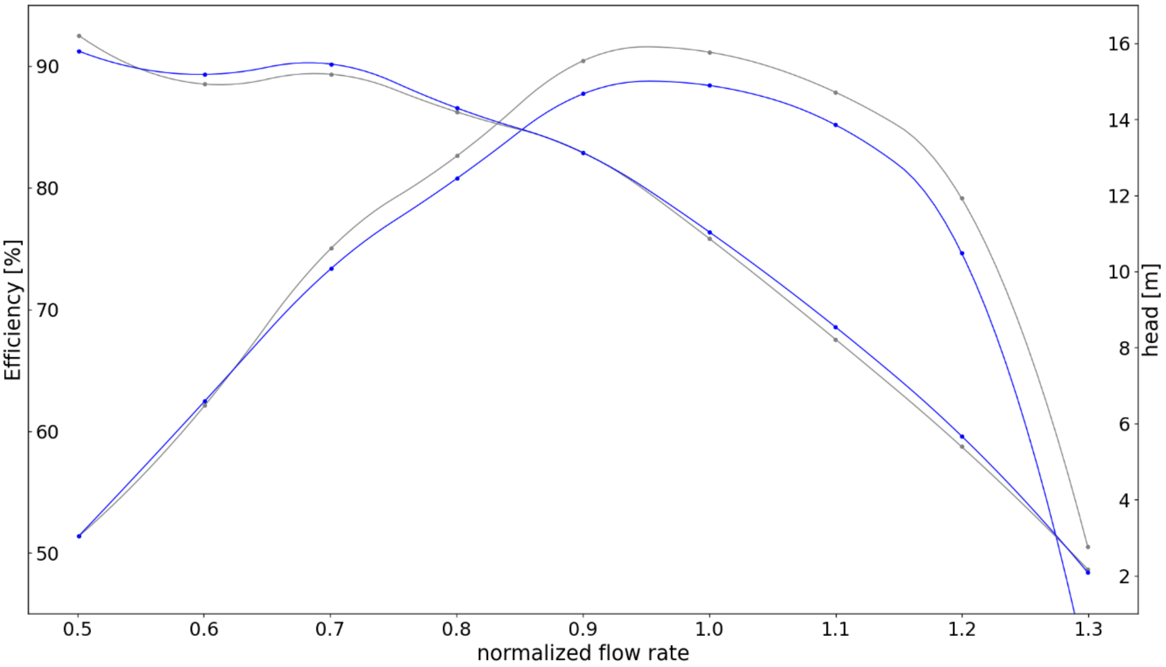

Figure 59: Optimization results. Comparison of hydraulic performance (efficiency and head) with the baseline design.

Once created, all the designs were computed on the HPC cluster. All the data were then copied to the local disc and processed with Python codes. This routine included efficiency and head curves processing from the monitor points data, BL data processing, objectives evaluation and various graphical outputs for better understanding of the data and optimization process. Examples of such graphical outputs can be seen in Figure 54, Figure 56, Figure 57 and Figure 59. The intention was to compare the results to the baseline design, and to judge how the objectives are being improved by the DYCORS search. As the simulation times were relatively long (ca. 2 to 3 days) and the processing of the data very complex and time-consuming (hours for every DYCORS iteration), the whole process was not fully automated.

In total, 12 iterations of this DYCORS optimization were executed. Hypervolume plot is shown in Figure 60. The hydraulic performance of the best design obtained from these 12 iterations can be seen in Figure 59. The BL shape for selected flow rate (*110 %* of Q_{OPT}) is in Figure 55. In both figures there is a comparison with the baseline design. As can be seen, the performance figures of the baseline design were still not matched.

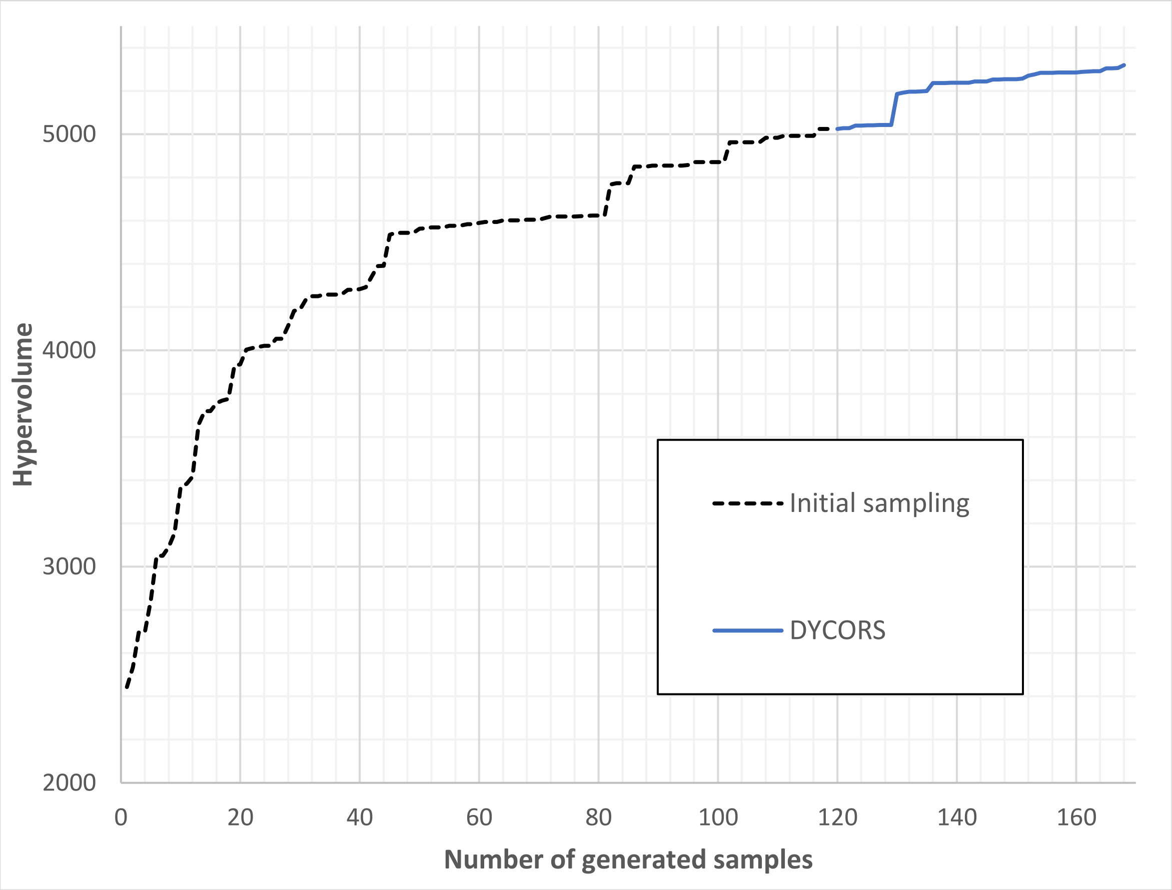

Figure 60: Hypervolume of generated samples in objective space.

After the 12 iteration, the hypervolume of the solutions was still growing. By analysing the results, though, it was decided that the optimization is still not sufficient to surpass the baseline design, and the run was terminated. The biggest concern was with the objectives related to the BL, and with the penalty function. None of the designs evaluated during the optimization run has managed to improve over the BL characteristics displayed by the baseline design. As for the performance characteristics, there was no improvement, either. Many of the samples were missing the head value, despite the penalization. The probable explanation is that the problem was too complex and too difficult for the DYCORS method to handle. The high number of parameters, together with a complicated scalarization approach, makes approximating the objective function landscape very difficult.

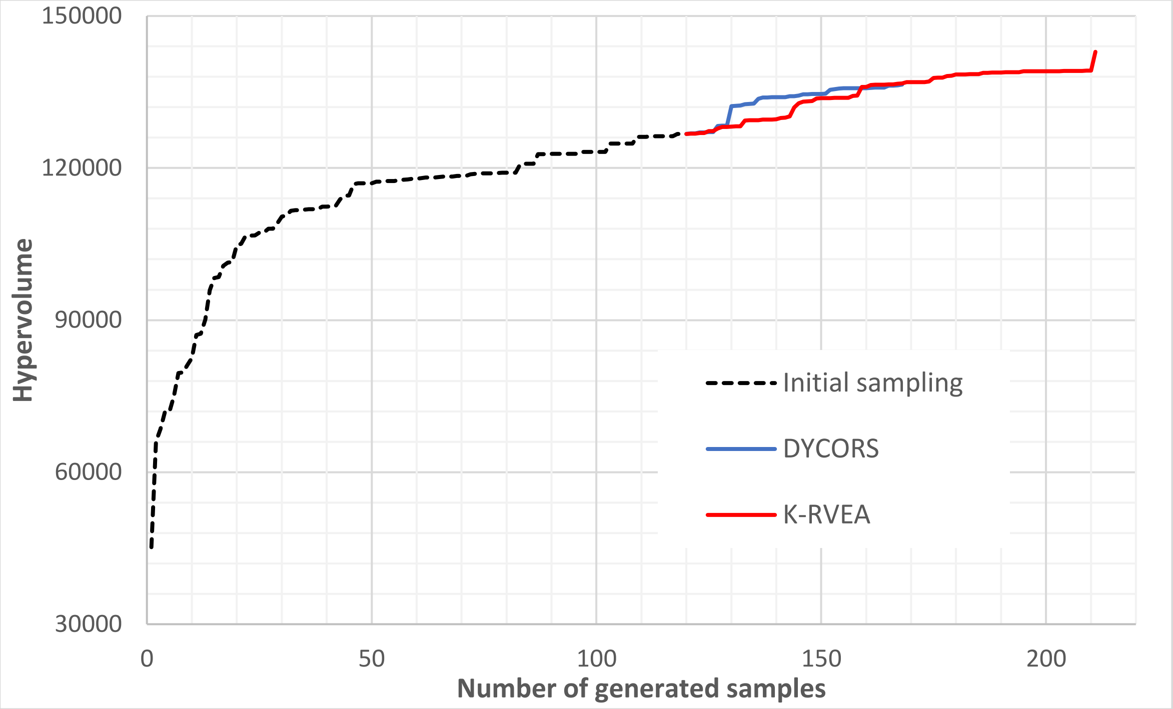

Figure 61: Hypervolume for both DYCORS and K-RVEA. For the comparison, the DYCORS samples were evaluated by the same objectives that were used for K-RVEA optimization.

8.5. Optimization – K-RVEA

For the second run, a multi-objective K-RVEA method, described in the previously studied case, was used. This time, six different objectives were considered:

1. Efficiency at the design point
2. Efficiency in the range of 60 to 100 % of the design point
3. Efficiency in the range of 100 to 120 % of the design point.
4. BL curve – percentage below the threshold
5. BL curve at the suction
6. Difference from the design head (at the design point)

Instead of the scalarization, necessary for DYCORS, each objective was considered independently now. Also, overall efficiencies for the sub-optimal, and “above-optimal” flow rates were considered separately. The intention was to have better control on optimizing the design for lower or higher flow rates. The difference from the design head was added to the objectives, too. To start the K-RVEA optimization, data from the same initial sampling as in the DYCORS optimization were used. As in the diffuser case, the method was set to generate three new samples in each iteration. Overall, ca. 60 samples were created – the non-dominated solutions can be seen in Figure 62, hypervolume of the objective space is shown in Figure 61.

K-RVEA displayed better performance than DYCORS. However, the improvement was still too slow, and the optimization run was finally ended. Despite the fact that the hypervolume was still increasing. One source of problems was the “difference from design head” objective. Setting the objective in such way actually means that the optimization was trying to find solutions with good performance, but with head value different than what was required. This in the end means wasting computational resources, as many of the designs did not meet the “design head” condition. Another complication was caused by the BL objectives. As measuring the BL properties is only a “replacement” for the true *NPSH*$_3$ computation, these two qualities are not completely correlated. These problems are further discussed in the following section.

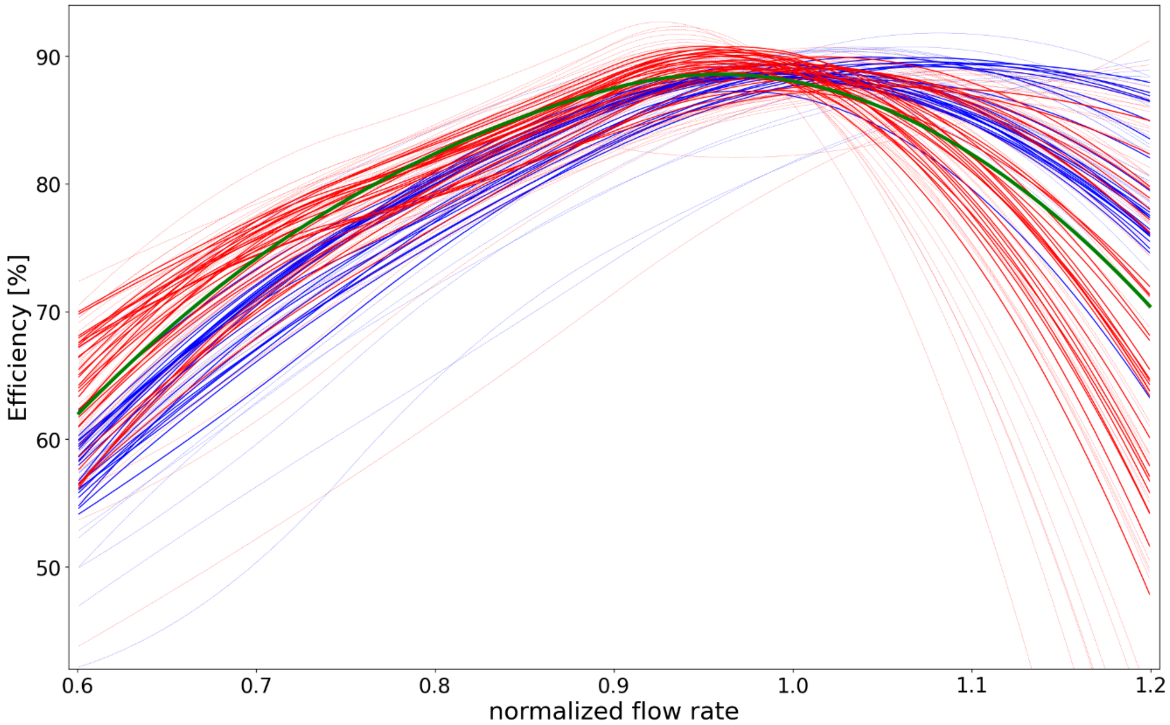

Figure 62: Efficiency curve approximated from five points (flow rates). The K-RVEA samples are in red, DYCORS in blue and the baseline design in green. The lines are bolder for design that are within the tolerance for the design head.

8.6. Conclusion and future plans

To perform the shape optimization of both impeller and diffuser, a new framework, based on ANSYS BladeGen and TurboGrid and Python codes was created. Then the framework was successfully tested and used to solve the optimization problem of an axial pump. When compared to the previously used ANSYS Workbench script, this new framework enabled a better control over the geometry creation and simulations. Its main advantage was avoiding the limitations of available tools offered by ANSYS. Thanks to this, a more complex approach to geometry creation and post-processing was possible. Multiple objectives were evaluated, including those based on analysing the Blade Loading over impeller blades. While not overly complicated in principle, driving multiple sequential simulations and processing of large amount of data on an HPC cluster is a complex task. Thus, the technical solutions related to these objectives were fairly challenging.

The ultimate goal of such approach was to have a general framework, ready to be a part of the hydraulic design process. The intention was to formulate the hydraulic design in a more abstract way, and to be able to optimize a given design (delivered by a hydraulic expert) by a combination of an efficient multiobjective optimization and automated parametric model. The framework is ready for this task, as by driving the BladeGen and TurboGrid via scripts, the full range of hydraulic capabilities (included in these software tools) is available for the automation. As BladeGen is commonly used for the hydraulic design, continuation of the process is very straightforward. The values of the hydraulic parameters (blade angles, thickness, dimensions, …) delivered by the hydraulic experts can be easily used for defining the parametric space – simply by creating the lower and upper bounds by offsetting the baseline values. Unlike the *shape-deformation* approach, described in *Chapter 3*, such parameters-based hydraulic shapes creation can include much wider range of designs and still allows for a good mesh quality.

When applied to the axial pump of *specific speed* $n_s = 680$, the results were not better than the baseline design. The high number of parameters, combined with multiple objectives, makes the optimization very complex and difficult. A proper formulation of the objectives has also proven to be very challenging. The problems are related to the way the objectives are handled in the multiobjective optimization. Basically, by searching for the Pareto front, the optimization tries to find all possible *trade-offs* between all the objectives. But for example, searching for trade-offs between efficiency and BL shape is problematic. The efficiency is a numerical value, but the BL shape is just an estimation of the real *$NPSH_3$* characteristics. I.e., the efficiency should be preferred by the optimization, which is difficult to obtain.

The suggested solution for the future development is to modify the optimization in such way, that it is capable of processing different types of information. The classical codes work on a strictly numerical basis. I.e., all the inputs it can use are numbers. But the optimization methods are nowadays generally based on the principle of maximizing the information that is obtained by (computationally expensive) evaluation of the objectives. With currently used codes, it is of course possible to include the additional information in a form of a penalty function. Meeting the head value, visually smooth blade shapes, desired BL curves – pursuing all these goals can be in principle supported by a proper formulation of a penalty function, added to the efficiency objectives. However, such approach interferes with the surrogates. A penalty function distorts the shape of the response surface, and, as a consequence, influences the selection of new samples for evaluations. Optimization method like K-RVEA, on the other hand, allows incorporating such objectives in a form of *preferences* or *classification*. The response surfaces of efficiency objectives than keep the original shape, and the additional information become part of the process of selecting new samples for objective function(s) evaluation.

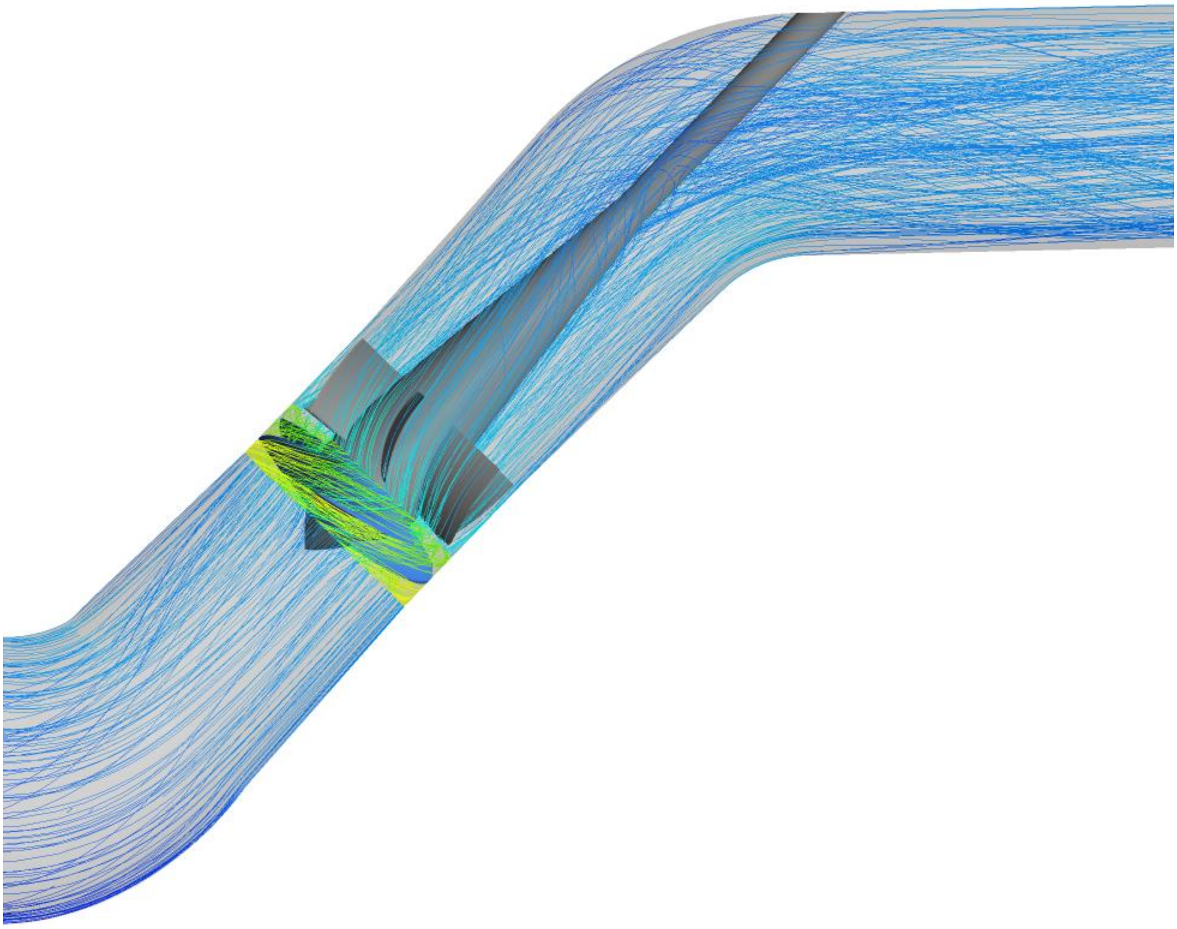

Figure 63: CFD of the baseline design - streamlines at Q_{OPT}.

9. Conclusion

In this thesis, experience with practical implementation of mathematical optimization for pump hydraulic design is summarized. Multiple real-world optimization cases were presented – ordered from lower to higher difficulty / complexity. The first presented case is not an actual optimization, but only uses iterative sampling in the parametric space. Next, the automated parametric model was created and connected with a suitable surrogate-based, single objective method. In the following cases, multiobjective optimization of pump stator was presented. Finally, in the last case, the optimization of the rotor and stator parts with respect to multiple objectives was presented.

The main goal of the work was to approach the hydraulic design more as an abstract optimization problem, that can be solved by a combination of sophisticated optimization methods and raw computational power. In principle, such solution could benefit from the increasing available computational power, caused by the technical development. However, in its most general form (a parametric model allowing for any possible pump hydraulic shapes), resulting optimization problem is still too difficult to be solved by the means of mathematical optimization. In practice, the best option is to build the optimization on the basis of the "classical" hydraulic design. The design performed by a hydraulic expert can serve as a baseline design, and the optimization is performed over a limited parametric space, "in a vicinity" of the baseline design. Practically, this can be realized by defining a hypercube (parameter ranges are set by lower and upper bounds) such, that is contains the baseline design. Narrowing the searched parametric space then significantly alleviates the complexity of the optimization.

The practical results have shown that for the smaller partial problems (of a shape optimization of some selected hydraulic parts, such as a diffuser), the optimization displayed great performance, and the performance of the baseline model was improved. For the more complex parametric model (impeller and diffuser), the results were not convincing. The high number of parameters, and difficult formulation of objectives, pose a big challenge for the selected optimization methods. Nonetheless, many practical aspects of the computationally expensive optimization have been tested in the presented cases, and the codes and techniques will be used for future development. Difficulties faced when optimizing pump hydraulic design can be like:

- Pump geometry creation.
- Mesh creation
- Automation of the process
- Computational demands of the numerical simulations
- Objectives formulation and evaluation
- High complexity of the optimization

Different ways of creating the geometry have been tested. From relatively simple parametric models, limited by the capabilities of DesignModeler, to complex Python-driven BladeGen models. The Python allowed for much more versatile shaping of the geometry, and also made a better control over the geometry and mesh quality possible. Automated mesh creation is a difficult task. Fortunately, ANSYS TurboGrid uses an effective combination of premade mesh topologies and optimization of control points, and works with great results on a vast majority of designs. Thus, impeller or diffuser meshes could be easily generated ad hexahedral and with high quality.

For the automation, batch mode of ANSYS program, driven by scripts, was used. Usually, a sample script was created, the parameters replaced by keywords, and used as a template for the Python routines. Overall, apart from complications caused by ANSYS limitations (lack of programming guide for Workbench etc.), the automation of geometry and mesh creation and CFX model assembly was relatively easy.

CFD simulations are computationally very expensive. To obtain at least a basic level of accuracy, computational meshes with millions of nodes are necessary for pump. Combined with high number of timesteps, on a single computer the transient simulation takes many hours or even days. For cavitational simulations it is even worse, as the sequence of simulations, necessary for obtaining a *head-drop* curve, can take even weeks. This severely limits the maximum number of objective evaluations. It is also necessary to use HPC resources and to use parallelization as much as possible, on both simulation and optimization level. Performing the automated optimization on an HPC cluster is considerably more complicated, as the individual tasks need to be first sent to a queue, and are not guaranteed to start at the same time. For optimal utilization of the resources, the long CFD simulations were split into a sequence of 2-hours runs. Another problem is frequent moving of the data between the temporary and central shared storages.

Formulation of the objectives proved to be possibly the biggest challenge. In case of diffuser optimization, considering only efficiency is sufficient. Three points (flow rates) at the efficiency curve is believed to be a good compromise between the computational costs and control over the hydraulic characteristics. For the impeller, the situation is much more complicated. For example, the cavitational properties represented by the $NPSH_3$ curve cannot be used for the optimization, as the evaluation of the curve takes too much time. A newly designed pump has also to meet a prescribed head value at the design point Q_{OPT}. Ensuring this during the optimization is not easy, as penalizing the objective functions (when the head value is not met) interferes with the surrogate models. Evaluation of the objectives was based on time data obtained from the simulations. In most case, the values averaged over last two rotations of the impeller were used. For the Blade Loading, a more extensive approach, based on processing a large amount of data, had to be used.

General problem faced in hydraulic shapes optimization is that the optimization only works with partial information, and with added “noise” (caused by limited accuracy of the numerical simulations). Situation, where the objective value is not given, can also happen due to geometry or mesh failures. This is caused by the fact that within the computational budget, only limited complexity of the numerical models can be allowed. This means that some of the physical phenomena, relevant for the pump performance evaluation, cannot be given by an objective function based on the numerical simulations. With technical development of CPU performance, it can be expected that in the future more advanced simulations will be used for optimization – such as models containing acoustics or Fluid-Structure interaction. Still, the most logical option for future development is an optimization routine such, that can effectively utilize all available information. Ideally in such way, that fast available information is used fast. For example, some combinations of the parameters lead to geometries with undesirable shape. Human expert can tell this by a visual control. But optimization, based purely on the computationally expensive evaluations, wastes hours of time to confirm information, that could have been obtained from (much faster) analysis of the geometry. In the same way it is possible to classify the Blade Loading results, efficiency curve shape, or meeting the head. I.e., it is possible to combine “hard” objectives (efficiency) with “soft” ones, based on classifications or user preferences. Framework developed in this work is a starting point for such future development.

Overall, when used properly, the shape optimization based on numerical simulations can be a valuable addition to the tools available to hydraulic designers. Apart from the optimization itself, there are other benefits. Understanding multiple objectives and possible trade-offs can be invaluable. It is still not possible to replace a human expertise with raw computational. But, with growing power available for the numerical simulations, the optimization can give the designer more and more insight into the design process, and allows for faster and cheaper development of new products.

10. References

1. R.K., Turton. Principles and practice of scaling laws. *Principles of Turbomachinery.* Dordrecht : Springer, 1984.
2. Gülich, Johann Friedrich. *Centrifugal Pumps.* s.l. : Springer-Verlag Berlin Heidelberg, 2014. 978-3-642-40114-5.
3. Stepanoff, A. J. *Centrifugal and Axial Flow Pumps: Theory, Design, and Application 2nd Edition.* s.l. : Krieger Publishing Company, 1993. 978-0894647239.
4. *https://en.wikipedia.org/wiki/Hydraulic_engineering.* [Online]
5. Sederberg, Thomas W. and Parry, Scott R. Free-Form Deformation of Solid Geometric Models. *ACM SIGGRAPH Computer Graphics.* 1986, Vol. 20, 4.
6. A. de Boer, M.S. van der Schoot, H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures.* 2007, Vol. 85, 11–14, pp. 784-795.
7. Hicks R. M., Henne, P. A. Wing Design by Numerical Optimization. *Journal of Aircraft.* 1978, Vol. 15, 7, pp. 407–412.
8. Masters, Dominic & Taylor, Nigel & Rendall, Thomas & Allen, C. & Poole, Daniel. Review of Aerofoil Parameterisation Methods for Aerodynamic Shape Optimisation. *Proceedings AIAA Science and Technology Forum.* 2015.
9. *Exploring the Fitness Landscape of a Realistic Turbofan Rotor Blade Optimization.* Kmec, Jakub & Schmitt, Sebastian. 2019.
10. Storn, R., Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization.* 1997, Vol. 11, pp. 341–359.
11. Kennedy, J. and Eberhart, R. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks.* 1995, pp. 1942–1948.
12. Preuss, Mike. *Multimodal Optimization by Means of Evolutionary Algorithms.* s.l. : Springer International Publishing AG, 2015. 9783319074061 .
13. Chugh, T., Sindhya, K., Hakanen, J. et al. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing.* 2019.
14. Ye, Pengcheng. A Review on Surrogate-Based Global Optimization Methods for Computationally Expensive Functions. *Software Engineering.* 2019, Vol. 7, 4, pp. 68-84.
15. Chugh, Tinkle & Sindhya, Karthik & Miettinen, Kaisa & Jin, Yaochu & Kratky, Tomas & Makkonen, Pekka. Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. 2017.
16. Witteck, D, Micallef, D, & Mailach, R. Comparison of Transient Blade Row Methods for the CFD Analysis of a High-Pressure Turbine. *Proceedings of the ASME Turbo Expo 2014: Turbine Technical Conference and Exposition.* June 16–20, 2014, Vol. 20: Turbomachinery.
17. *Study of Steady State and Transient Blade Row CFD Methods in a Moderately Loaded NASA Transonic High-Speed Axial Compressor Stage.* Qizar, Mohammed & Mansour, Mahmoud & Goswami, Shraman. 2013. ASME Turbo Expo 2013: Turbine Technical Conference and Exposition.
18. Giannakoglou K.C., Papadimitriou D.I. Adjoint Methods for Shape Optimization. [book auth.] Janiga G. (eds) Thévenin D. *Optimization and Computational Fluid Dynamics.* s.l. : Springer, Berlin, Heidelberg, 2008.
19. Liu, H., Ong, Y. & Cai, J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization.* 2018, Vol. 57, pp. 393–416.
20. Jin, Yaochu. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation.* 2011, Vol. 1, 2, pp. 61-70.

21. Regis, Rommel & Shoemaker, Christine. A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. *INFORMS Journal on Computing.* 2007, Vol. 19, pp. 497-509.
22. —. Parallel Stochastic Global Optimization Using Radial Basis Functions. *INFORMS Journal on Computing.* 2009, Vol. 21, pp. 411-426.
23. https://ccse.lbl.gov/people/julianem/index.html. [Online]
24. Miettinen, K. Introduction to Multiobjective Optimization : Noninteractive Approaches. *Multiobjective Optimization : Interactive And Evolutionary Approaches.* 2008, pp. 1–26.
25. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., & Sindhya, K. A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-objective Optimization. *IEEE Transactions on Evolutionary Computation.* 2018, Vol. 22, 1, pp. 129-142.
26. R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation.* 2016, Vol. 20, 5, pp. 773-791.
27. *Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system.* T. Chugh, K. Sindhya, K. Miettinen, Yaochu Jin, T. Kratky and P. Makkonen. San Sebastian : s.n., 2017. IEEE Congress on Evolutionary Computation (CEC). pp. 1541-1548.
28. Shoemaker, Rommel G. Regis & Christine A. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization.* 2013, Vol. 45, 5, pp. 529-555.
29. S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera,. *S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera,.*
30. S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera. CFD for Centrifugal Pumps: A Review of the State-of-the-Art. *Procedia Engineering.* 2013, Vol. 51, pp. 715-720.
31. R. Spence, J. Amaral-Teixeira. A CFD parametric study of geometrical variations on the pressure pulsations and performance characteristics of a centrifugal pump. *Computers & Fluids.* 2009, Vol. 38, 6, pp. 1243-1257.
32. *Shape Optimization of Turbomachinery Blade Using Multiple Surrogate Models.* Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei. 2006. ASME 2006 2nd Joint U.S.-European Fluids Engineering Summer Meeting Collocated With the 14th International Conference on Nuclear Engineering.
33. Jingwei Jiang, Haopeng Cai, Cheng Ma, Zhengfang Qian, Ke Chen & Peng Wu. A ship propeller design methodology of multi-objective optimization considering fluid–structure interaction. *Engineering Applications of Computational Fluid Mechanics.* 2018, Vol. 12, 1.
34. Derakhshan, Shahram & Mohammadi, Bijan & Nourbakhsh, Ahmad. The comparison of incomplete sensitivities and Genetic algorithms applications in 3D radial turbomachinery blade optimization. *Computers & Fluids.* 2010, Vol. 39, pp. 2022-2029.
35. ones, D.R. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization.* 2001, Vol. 21, pp. 345–383.
36. Shahram Derakhshan, Maryam Pourmahdavi, Ehsan Abdolahnejad, Amin Reihani, Ashkan Ojaghi. Numerical shape optimization of a centrifugal pump impeller using artificial bee colony algorithm. *Computers & Fluids.* 2013, Vol. 81, pp. 145-151.
37. Xiao Tang, Jiaqi Luo, Feng Liu. Aerodynamic shape optimization of a transonic fan by an adjoint-response surface method. *Aerospace Science and Technology.* 2017, Vol. 68, pp. 26-36.

38. C.A. Gilkeson, V.V. Toropov, H.M. Thompson, M.C.T. Wilson, N.A. Foxley, P.H. Gaskell. Dealing with numerical noise in CFD-based design optimization. *Computers & Fluids.* 2014, Vol. 94, pp. 84-97.
39. V.O. Lomakin, P.S. Chaburko, M.S. Kuleshova. Multi-criteria Optimization of the Flow of a Centrifugal Pump on Energy and Vibroacoustic Characteristics. *Procedia Engineering.* Vol. 176.
40. Huang, R., Luo, X., Ji, B. et al. Multi-objective optimization of a mixed-flow pump impeller using modified NSGA-II algorithm. *Sci. China Technol. Sci.* 2015, Vol. 58, pp. 2122–2130.
41. Yongsheng Lian, Akira Oyama, Meng-Sing Liou. Progress in design optimization using evolutionary algorithms for aerodynamic problems. *Progress in Aerospace Sciences.* 2010, Vol. 46, 5-6, pp. Pages 199-223.
42. K.-H. Wu, B.-J. Lin & C.-I. Hung. Novel Design of Centrifugal Pump Impellers Using Generated Machining Method and CFD. *ngineering Applications of Computational Fluid Mechanics.* 2008, Vol. 2, 2, pp. 195-207.
43. Siddique, M Hamid & Afzal, Arshad & Samad, Abdus. Design Optimization of the Centrifugal Pumps via Low Fidelity Models. *Mathematical Problems in Engineering.* 2018, pp. 1-14.
44. Ivo Marinić-Kragić, Damir Vučina & Zoran Milas. 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise-related excellence criteria and constraints. *Engineering Applications of Computational Fluid Mechanics.* 2016, Vol. 10, 1, pp. 209-227.
45. Xiao Tang, Jiaqi Luo, Feng Liu. Aerodynamic shape optimization of a transonic fan by an adjoint-response surface method. *Aerospace Science and Technology.* 2017, Vol. 68, pp. Pages 26-36.
46. Kang, I. S. Jung and W. H. Jung and S. H. Baek and S. Shape Optimization of Impeller Blades for a Bidirectional Axial Flow Pump using Polynomial Surrogate Model. *Transactions of the Korean Society of Mechanical Engineers B.* 2012, Vol. 6, 6.
47. H. Safikhani, A. Khalkhali & M. Farajpoor. Pareto Based Multi-Objective Optimization of Centrifugal Pumps Using CFD, Neural Networks and Genetic Algorithms. *Engineering Applications of Computational Fluid Mechanics.* 2011, Vol. 5, 1, pp. 37-48.
48. V.O. Lomakin, P.S. Chaburko, M.S. Kuleshova. Multi-criteria Optimization of the Flow of a Centrifugal Pump on Energy and Vibroacoustic Characteristics. *Procedia Engineering.* 2017, Vol. 176, pp. 476-482.
49. *Optimal design of multi-conditions for axial flow pump.* Shi, L & Tang, F & Liu, C & Xie, R & Zhang, W. 2016. IOP Conference Series: Earth and Environmental Science. Vol. 49.
50. Huang, R., Luo, X., Ji, B. et al. Multi-objective optimization of a mixed-flow pump impeller using modified NSGA-II algorithm. *Sci. China Technol. Sci.* 2015, Vol. 58, pp. 2122–2130.
51. Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei. Multiple Surrogate Modeling for Axial Compressor Blade Shape Optimization. *Journal of Propulsion and Power.* 2008, Vol. 24, pp. 302-310.
52. Han, X., et al. Impeller Optimized Design of the Centrifugal Pump: A Numerical and Experimental Investigation. *Energies.* 2018, Vol. 11, p. 1444.
53. Xing’an Zhao, Biao Huang, Tairan Chen, Guoyu Wang, Deming Gao, Jing Zhao. Numerical simulations and surrogate-based optimization of cavitation performance for an aviation fuel pump. *Journal of Mechanical Science and Technology.* 2017, Vol. 31, 2, pp. 705-716.

54. Rajkumar Roy, Srichand Hinduja, Roberto Teti. Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals.* 2008, Vol. 57, 2, pp. 697-715.
55. *Shape Optimization of Turbomachinery Blade Using Multiple Surrogate Models.* Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei. 2006. ASME 2006 2nd Joint U.S.-European Fluids Engineering Summer Meeting Collocated With the 14th International Conference on Nuclear Engineering.
56. Siddique, M Hamid & Afzal, Arshad & Samad, Abdus. Design Optimization of the Centrifugal Pumps via Low Fidelity Models. *Mathematical Problems in Engineering.* 2018, pp. 1-14.
57. Baoshan Zhu, Xuhe Wang, Lei Tan, Dongyue Zhou, Yue Zhao, Shuliang Cao. Optimization design of a reversible pump–turbine runner with high efficiency and stability. *Renewable Energy.* 2015, Vol. 81, pp. 366-376.
58. Milas, Zoran & Vučina, Damir & Marinic-Kragic, Ivo. Multi-Regime Shape Optimization of Fan Vanes for Energy Conversion Efficiency using CFD, 3D Optical Scanning and Parameterization. *Engineering Applications of Computational Fluid Mechanics.* 2014, Vol. 8, pp. 407-421.
59. X.D. Wang, C. Hirsch, Sh. Kang, C. Lacor. Multi-objective optimization of turbomachinery using improved NSGA-II and approximation model. *Computer Methods in Applied Mechanics and Engineering.* 2011, Vol. 200, 9-12, pp. 883-895.
60. Škerlavaj, Aljaž & Morgut, M & Jošt, Dragica & Nobile, Enrico. Optimization of a single-stage double-suction centrifugal pump. *Journal of Physics: Conference Series.* 2017, Vol. 796, 1.
61. Iuliano, Emiliano. Global optimization of benchmark aerodynamic cases using physics-based surrogate models. *Aerospace Science and Technology.* 2017, Vol. 67, pp. 273-286.
62. KIM, Jin-Hyuk & Choi, Jae-Ho & Husain, Afzal & Kim, Kwang-Yong. Multi-Objective Optimization of a Centrifugal Compressor Impeller through Evolutionary Algorithms. *Proceedings of the Institution of Mechanical Engineers Part A Journal of Power and Energy.* 2010, Vol. 224, pp. 711-721.
63. Erik Tengs, Pål-Tore Storli & Martin Holst. Optimization procedure for variable speed turbine design. *Engineering Applications of Computational Fluid Mechanics.* 2018, Vol. 12, 1, pp. 652-661.
64. Wang, Yuqin & Huo, Xinwang. Multiobjective Optimization Design and Performance Prediction of Centrifugal Pump Based on Orthogonal Test. *Advances in Materials Science and Engineering.* 2018, Vol. 15, pp. 1 - 10.
65. Škerlavaj, Aljaž & Morgut, M & Jošt, Dragica & Nobile, Enrico. Decoupled CFD-based optimization of efficiency and cavitation performance of a double-suction pump. *Journal of Physics: Conference Series.* 2017, Vol. 813, 1.
66. Shaaban, S. Design optimization of a centrifugal compressor vaneless diffuser. *International Journal of Refrigeration.* 2015, Vol. 60, pp. 142-154.
67. Feliot P., Bect J., Vazquez E. User Preferences in Bayesian Multi-objective Optimization: The Expected Weighted Hypervolume Improvement Criterion. *Lecture Notes in Computer Science, vol 11331.* 2019.
68. S.N. Skinner, H. Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing.* 2018, Vol. 62, pp. 933-962.
69. Safikhani, Hamed & Khalkhali, Abolfazl & Farajpoor, M. Pareto Based Multi-Objective Optimization of Centrifugal Pumps Using CFD, Neural Networks and Genetic Algorithms. *Engineering Applications of Computational Fluid Mechanics.* 2011, Vol. 5, pp. 37-48.

70. Jinya Zhang, Hongwu Zhu, Chun Yang, Yan Li, Huan Wei. Multi-objective shape optimization of helico-axial multiphase pump impeller based on NSGA-II and ANN. *Energy Conversion and Management.* 2011, Vol. 52, 1, pp. 538-546.
71. *A multi-response adaptive sampling approach for global metamodeling.* Liu, Haitao & Xu, Shengli & Wang, Xiaofang & Yang, Shuhua & Meng, Jigang. 2016. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.
72. Zhihui Li, Xinqian Zheng. Review of design optimization methods for turbomachinery aerodynamics. *Progress in Aerospace Sciences.* 2017, Vol. 93, pp. 1-23.
73. Yin Yu, Zhoujie Lyu, Zelu Xu, Joaquim R.R.A. Martins. On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Aerospace Science and Technology.* 2018, Vol. 75, pp. 183-199.
74. J. Tian, Y. Tan, J. Zeng, C. Sun and Y. Jin. Multiobjective Infill Criterion Driven Gaussian Process-Assisted Particle Swarm Optimization of High-Dimensional Expensive Problems. *IEEE Transactions on Evolutionary Computation.* 2019, Vol. 23, 3, pp. 459-472.
75. Zhang, Qi-hua & Cao, Li & Yan, Zhaoxu & Zhang, Weidong. Hydraulics and Blading of Centrifugal Pump Impellers: A Systematic Review and Application. *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering.* 2018.
76. Claudia Schillings, Stephan Schmidt, Volker Schulz. Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids.* 2011, Vol. 46, 1, pp. 78-87.
77. L. Bonfiglio, P. Perdikaris, S. Brizzolara, G.E. Karniadakis. Multi-fidelity optimization of super-cavitating hydrofoils. *Computer Methods in Applied Mechanics and Engineering.* 2018, Vol. 332, pp. 63-85.
78. Wei-xue Cao, Xue-yi You. The inverse optimization of exhaust hood by using intelligent algorithms and CFD simulation. *Powder Technology.* 2017, Vol. 315, pp. 282-289.
79. Antonio Posa, Antonio Lippolis. A LES investigation of off-design performance of a centrifugal pump with variable-geometry diffuser. *International Journal of Heat and Fluid Flow.* 2018, Vol. 70, pp. 299-314.
80. Md. Abdul Raheem Junaidi, N.B.V Laksmi Kumari, Mohd Abdul Samad, G.M. Sayeed Ahmed. CFD Simulation to Enhance the Efficiency of Centrifugal Pump by Application of Inner Guide Vanes. *Materials Today: Proceedings.* 2015, Vol. 2, 4-5, pp. 2073-2082.
81. Paquete, Andreia P. Guerreiro and Carlos M. Fonseca and Luís. The Hypervolume Indicator: Problems and Algorithms. *arXiv:2005.00515.* 2020.

11. List of Images

Palacký University Olomouc
Faculty of Science
Department of Experimental Physics

Dissertation Outline

Shape optimization of hydraulic surfaces of the impeller and stator parts of hydrodynamic pumps

1. Introduction

Mathematical optimization is an important part of mathematical theory. It can be applied to many theoretical and real-world problems. With the rapid development of numerical modelling in last decades, engineering optimization has become a hot topic. By connecting the tools of mathematical optimization and numerical simulations, it is possible to guide many design and technical problems, such as shape / topology optimization, or inverse problems. In modern society, virtually every product has been optimized with respect to some objectives.

In this work, the goal is to present how the optimization can be used in pump industry. To succeed in the competition, every pump design needs to be carefully optimized. This is done on multiple levels – mechanical and material solutions, production technologies and costs and hydraulic shapes. The hydraulic design typically utilizes exact analytical formulas, combined with data and experience-based corrections. For this purpose, numerical modelling of fluid is commonly used. It is capable of describing the pump performance and operation with a high level of accuracy, and gives the designer a tool to fine-tune the pump and balance the trade-offs between various conflicting objectives.

Here we focus on shape optimization as a part of the hydraulic design. The pump design uses input in the form of numerical geometry parameters (dimension, blade angles etc.) and numerical output such as performance characteristics (efficiency at a given flow-rate etc.). Thus, in principle it is possible to view the hydraulic design as an optimization task. When performed by a hydraulic expert, it is approached as a manual, semi-intuitive optimization. However, numerical simulations can be automated, and connected with an optimization code. This way, we can utilize the full potential of modern optimization methods, developed specifically to help aiding the design process. Still, there remain many challenges related to the practical incorporation of optimization into the hydraulic design process. This thesis deals mostly with the practical side of the pump design and optimization. On multiple selected examples, the goals and practical challenges are shown and explained.

2. Centrifugal Pumps - basic concepts

Centrifugal pumps are machines that convert mechanical energy to a kinetic energy of the moving fluid. It typically comprises of a rotating impeller, driven by an engine through a shaft, a stator part and a casing. The impeller adds the kinetic energy to the fluid, and the stator helps to convert the kinetic energy into static pressure. They are utilized in many areas, such as water or petroleum pumping. Any pump can be described by its performance and operational characteristics.

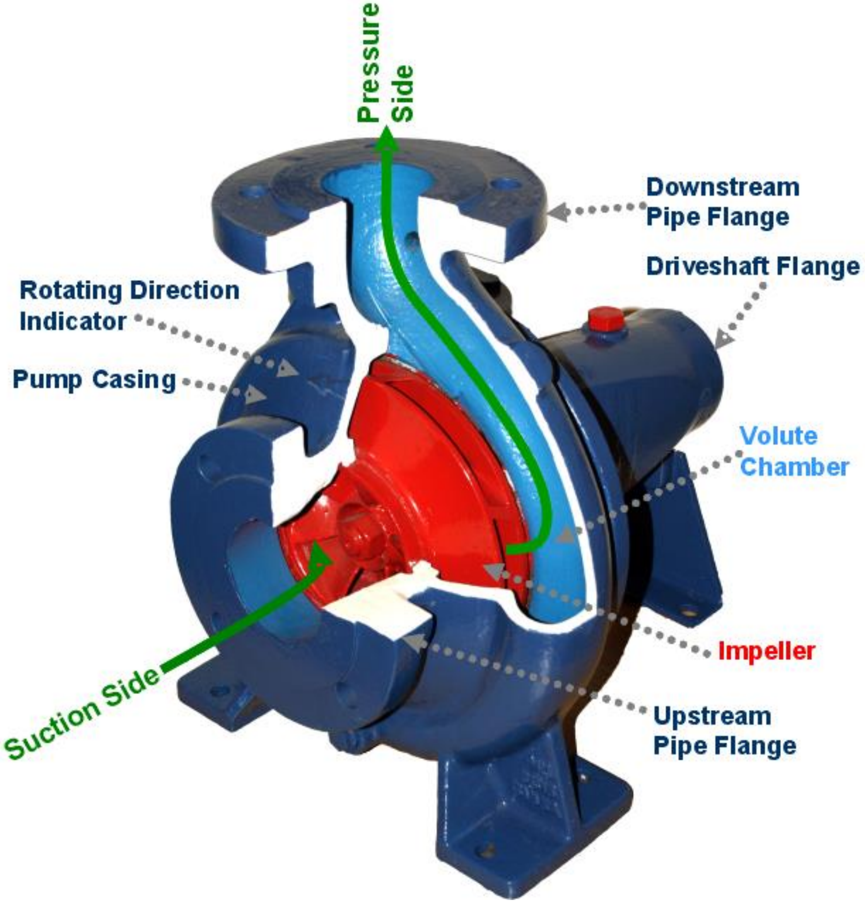

Figure 1: Centrifugal pump schematics (source: Wikipedia)

The most important characteristics related to the hydraulic performance of the pump are:

1. flow rate *Q* - volume of fluid transported per time unit. The flow rate where the pump reaches peak efficiency is called optimum and denoted Q_{OPT}.
2. Head *H* – difference of the total pressure between the outlet and the inlet
3. efficiency η - the ratio between energy transferred to the fluid kinetic energy and mechanical energy spent on the pump operation
4. $NPSH_3$, the so-called suction ability of the pump – it denotes the Net Positive Suction Head (i.e. the inflow pressure), below which the pump performance (measured by head) is degraded by 3 percent.

2.1. (Centrifugal) pumps classification and specific speed

The size of centrifugal pumps can vary from a few watts up to hundreds of MWs. However, all these pumps can be classified by the so-called specific speed.

$$n_s = n \cdot \frac{\sqrt{Q_{OPT}}}{(H_{OPT})^{0.75}}$$

Where *n* is the pump speed (in *rpm)* and H_{OPT} is the head at Q_{OPT}. If we follow the so-called *Affinity laws:* $Q_\lambda = \lambda^3 \cdot Q$, $H_\lambda = \lambda^2 \cdot Q$, we get:

$$n_{s,\lambda} = n \cdot \frac{\sqrt{\lambda^3 \cdot Q_{OPT}}}{(\lambda^2 \cdot H_{OPT})^{0.75}} = n_s$$

I.e. the specific speed remains the same when scaling the pump. Pumps range from "*low Q, high H"* types (low specific speed – radial pumps) to "*high Q, low H"* specifications (high specific speed – axial pumps).

Two pumps are called hydraulically similar, if both the hydraulic shapes and the fluid dynamics are similar. I.e. there exists a real coefficient λ > 0 such that *the hydraulic shape of the second pump is λ-scaled geometry of the first pump (including the surface roughness).*

Of course, the conditions of hydraulic similarity are not met rigorously in real life. Obviously, gravity or speed of sound remain the same. But, for practical applications this can be safely ignored. This is useful for both so-called *model testing* (i.e. testing a *scaled-down* version of a large production pump) and also for the methods of hydraulic design. The hydraulic similarity plays an important role for the pump design, as any pump can be classified by the *specific speed*. Thus, a pump can be designed by its type, no matter the scale.

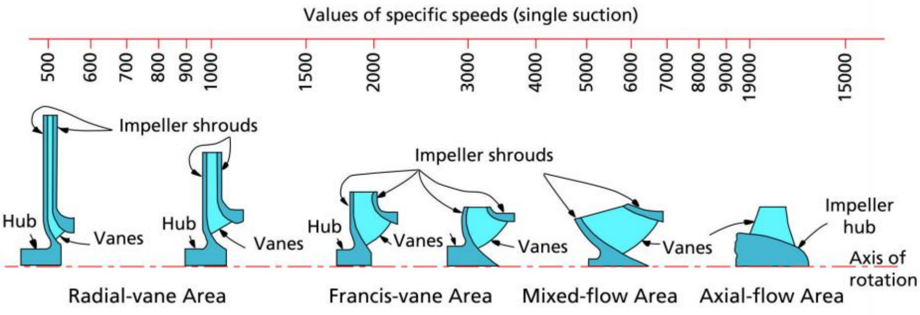

Figure 2: Pump types by specific speed. Source: www.introtopumps.com/pump-terms/ns-specific-speed/

2.2. Hydraulic design

By hydraulic design we mean designing the hydraulic (wetted) shapes of the pump. No direct analytical method for *optimal* design is known. Instead, multiple “hybrid” (semi-analytical) design theories, each suitable or recommended for a certain interval of specific speeds, have been proven for the impeller design. “Hybrid” means that these methods combine exact analytical formulas with “engineering” approximations, based on experience and simplifications. The theory originates from Euler’s pump equation and velocity triangles (between the circumferential and relative velocity components). The good practice is to approach the impeller design process as a sequence of steps:

1. Deciding the *Main dimensions*, i.e. the inflow and outflow dimensions. This is mostly dictated by the required performance parameters (flow rate and head).

2. Meridional shape of the impeller, i.e. the transition from inflow to outflow part. The goal is to distribute the fluid momentum change in the “optimal” way.
3. Blades – number of blades, relative position in the passage and leading and trailing angles. The blades are shaped at multiple *camber lines*, in case of an impeller there are usually at least three – hub, midline and shroud.

If the stator part contains blades (i.e. for example in a case of axial diffuser), the design methods are similar to the impeller. For different parts (volute, suction), different design methods exist. Generally, the designer tries to minimize the energy dissipation (= efficiency loss), caused by the whirls. I.e. from the geometrical perspective, the energy conversion needs to be as smooth as possible.

All the design theories are very well elaborated and described in literature (1), with many experimentally verified examples. Thanks to these verifications, experiment-based corrections are also available. These corrections are useful, as the hydraulic theory does not consider the three-dimensionality of the flow (and other physical phenomena, such as cavitation). As a result, for one particular flow-rate, it is typically possible to design a highly-efficient pump – very quickly and using solely analytical tools and experience. However, vast majority of pumps is not operated at a single point. Instead, a whole working range has to be considered. As in the practical applications the pumps speed (rpm) remains fixed (and the same obviously holds for the shape), the velocity triangles change with the flow rate. As a consequence, any pump inevitably displays sub-optimal performance when it is not working at its design point. This all means that pump design is a very complex process, that involves balancing multiple conflicting objectives and utilize many “layers” of information – from fast analytical methods to demanding and costly numerical simulations and experimental measurements. Here are some examples what needs to be considered by the designer:

- Peak efficiency (at the design point)
- Efficiency in the whole working range
- Head at the design point
- NPSH$_3$

- Technical limits (minimal thickness of the blades, manufacturability, dimension limits specified by customer, …)

And the tools available for estimating / evaluating the design performance:

- Analytical methods based on "one-dimensional" flow properties
- Engineering approximations based on previously measured designs
- Numerical simulations
- Experimental measurements, typically performed on model (scaled down) pump.

The analytical methods, together with the corrections, are very fast and efficient. But since fluid dynamics is highly non-linear phenomenon, they have limited accuracy for non-optimal flow rates, or in cases where the assumptions of the methods cannot be quite met for some reasons. (For example, limits on pump dimensions that do not allow for the "best practice" main dimensions, etc.)

The numerical simulations and experimental measurements are both used as a tool of assessing the performance. These methods can give valuable information to guide the designer, but they are also much slower and more expensive than the analytical methods. The workflow is typically like:

1. Hydraulic design – analytical methods.
2. Evaluate hydraulic performance by numerical simulations
3. If changes needed then back to 2.
4. Evaluate the $NPSH_3$ by more complex, multiphase simulations
5. If changes needed then back to 2.
6. Construction
7. Mechanical simulations.
8. If changes needed then back to 2 or 6 (depending on what needs to be improved).
9. (Model) pump production
10. Experimental measurements

A visualisation of the workflow can be seen in Figure 3. The later the stage of development, the more expensive any changes become.

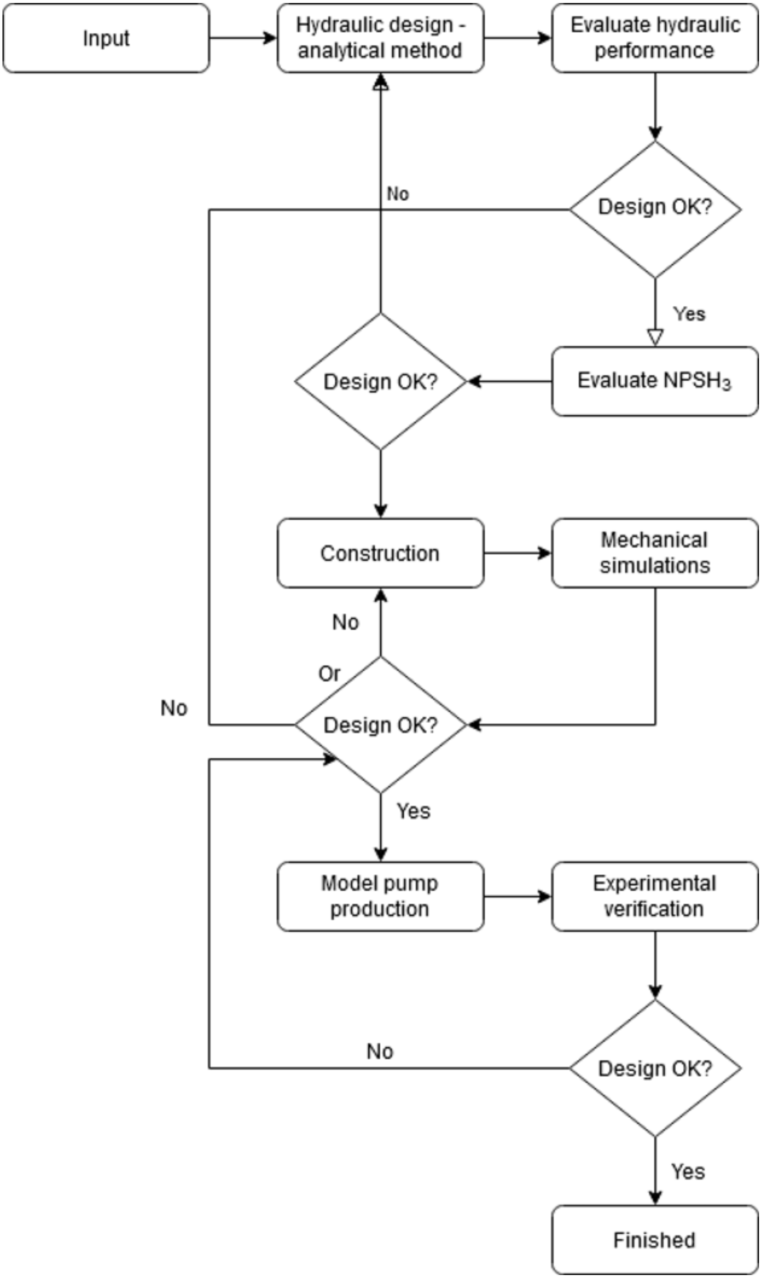

Figure 3: Scheme of hydraulic design.

2.3. Hydraulic design and optimization

As was described in the previous text, the "good practice" approach to hydraulic design is, in principle, a multi-objective (and "multi-level") optimization. The pump geometry is always described by (a finite number) N parameters, and we also have a finite number M of objectives. Thus, the problem of hydraulic design can be viewed as:

$$minimize\, J: \Omega \subset R^N \rightarrow R^M$$

where Ω is a (bounded) set of N parameters describing the pump geometry, and J

is an evaluation of pump performance. In the "classical", i.e. expert approach to the hydraulic design, this optimization is not performed in a strictly mathematical sense. Instead, it is combined with human "impressions", where many of the objectives are not evaluated solely by numbers. It is of course possible to perform a "true" optimization, but it brings considerable challenges:

1. It is necessary to fully automate the evaluation of the objectives. I.e. geometry creation, mesh generation, computational model assembly, running simulations and the post-processing. While this is no problem in theory, in real applications it is not an easy task.
2. In many cases, exact formulation of the objectives can be challenging.
3. Numerical simulations are computationally very demanding. Having a high-accuracy, multi-physics model is not a viable option for an optimization run. Instead, compromises and simplification are necessary. This, on the other hand, further (together with 2) limits the set of objectives that can be actually used for the optimization.

In theory, it is possible to make a general parametric model, that can cover all possible pump shapes. And to optimize it just for any physically possible pump performance. This is ensured by the fact that both the parameters and objectives are bounded, and their numbers are finite. However, such optimization would be extremely demanding due to large number of both parameters and objectives, combined with multimodality.

When practical limitations are considered, the optimization becomes not a replacement, but instead an addition for the design process performed by human experts. I.e. it is usually incorporated into the steps 2 and 3 (as mentioned in the workflow description in the previous chapter) of hydraulic design process. This means that the initial design is still performed by a hydraulic expert, and the consecutive optimization uses this initial design as a starting point. This can save a lot of (computational) resources. And, since the fidelity of the numerical modelling is limited in practice, it limits the chance that the optimization arrives at an impractical design due to not having complete information about some aspects of the pump performance.

In the next chapter, the most common optimization approaches currently used in pump hydraulic design are described.

3. Current approach to hydraulic shapes optimization

There are multiple ways of approaching the hydraulic design as an optimization problem. Or, more generally, how the approach the so-called *Computationally-expensive optimization* problems. First classification would be by the type of the optimization routine:

1. **Gradient-based methods**. These methods can converge (to a local minimum) very fast, even for large number of parameters. However, these methods suffer from multiple drawbacks in practical application. First, in many commercial simulation packages, the derivatives (of the objective function) are not available (to the user). Second, these are *local* methods, and the objective functions in hydraulic design can be multi-modal. Another problem is, that the objective function evaluation can fail for various reasons, and such methods were not designed with such situation in mind.
2. **Global heuristic methods**. such as genetic algorithms, DE (2), PSO (3), etc (4). These are global methods, typically very robust, capable of dealing with very difficult objective functions. They also require relatively big number of the (computationally

expensive) objective function evaluations. This makes their application for hydraulic design problematic, as the computational costs can be too prohibitive.

3. **Surrogate-assisted optimization.** In this case, the objective function is replaced by an approximation, usually called a *surrogate* or *response surface*. The surrogate can help in guiding the search and speeding the optimization process. Of course, building and updating the surrogate cost computational resources, too. But in a case of computationally expensive objective functions the overhead is negligible. A survey of various methods performance is in (5).

4. Case 1 - Pump stator

Objectives: Pump efficiency and circumferential velocity at the outflow, at the optimal flow rate.
Solution: ANSYS Workbench parametric model driven through text script files, Transient Blade Row model. Best design selected from Latin hypercube sampling of the parametric space in second iteration.
Results: Efficiency increased from 3% up to 8% in the working range.

4.1. Introduction

The studied pump was a diagonal one, designed for specific speed n_s = 240, with adjustable blades of the impeller and a diagonal diffuser. The “adjustable blades” means the attack angle of the blades can be set within a range of values, and allows for operating the pump in wider range of flow rates. The hydraulic parameters were only mediocre, and as the CFD analysis revealed, this was mainly due to the diffuser. A well-designed diffuser can transfer most of the circumferential velocity (inevitably found at the outlet of the rotating impeller) to the desired forward movement. In our case, the residual circumferential part of the velocity was still significant, even at the optimal flow rate. As a result, the total (combined) velocity of the flow in the outlet parts was increased. Thus, the efficiency was lowered significantly due to the hydraulic losses in the elbow. Because of this, the stator was deemed as the ideal starting point for pump optimization.

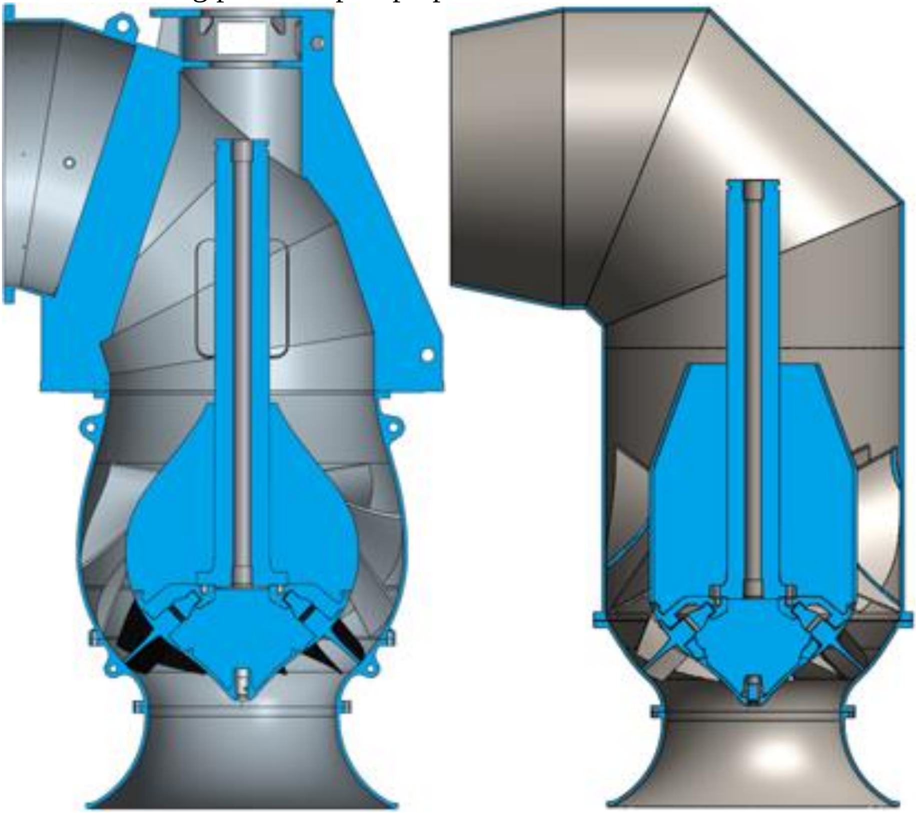

Figure 4: Original pump (left) vs the new design (right)

4.2. Parametric model and automation

ANSYS Workbench was used for creation of the computational models. Parametric diffuser model in DesignModeler was connected to automated structured (hexahedral) mesh generation in TurboGrid. Next, the diffuser mesh was updated in a premade CFX model. The other parts (Inflow, Impeller and Outflow) remained fixed. These steps (updating the diffuser geometry, mesh and CFX model) were recorded as a Workbench script. Next, the numerical values of the considered parameters were replaced by keywords (*par_1*, *par_2*, …) and the script file was used as a template. With Excel macros, a set of Workbench replay files could be easily created, based on a table containing a list of names and appropriate parameters. In total, 13 geometric parameters were considered.

The design was based on the "standard" methods for hydraulic design, and the parameter ranges were determined by an offset from these initial values. Since the intention was to only consider the design point Q_{OPT}, it was possible to use the so-called TBR (Transient Blade Row) method (6), (7). In this method, only one passage of the impeller and diffuser was considered.

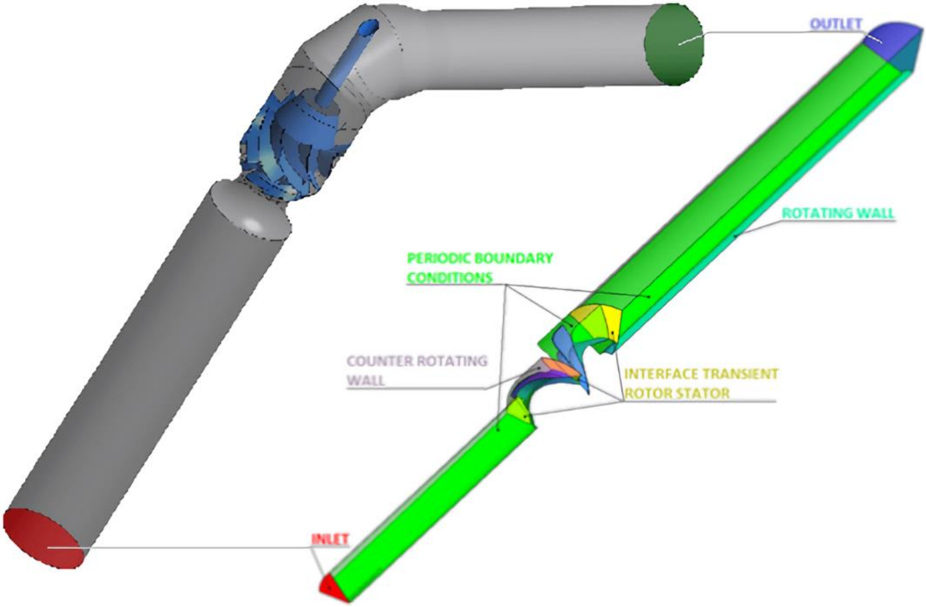
OUTLET
ROTATING WALL
PERIODIC BOUNDARY CONDITIONS
COUNTER ROTATING WALL
INTERFACE TRANSIENT ROTOR STATOR
INLET

Figure 5: Full (left) vs TBR (right) CFD model

Timestep was chosen with respect to count of impeller and diffuser blades. One timestep represents 2.5° and has this value:

$$(1)\ \Delta t = \frac{1}{i \cdot n \cdot p} = \frac{1 \cdot 60}{18 \cdot 294 \cdot 8} = 0.00141723(s)$$

with p being the impeller blades count, i being the number of timesteps per passage and n (rpm) the rotor rotation speed (in *rpm*).

As a turbulence model, SST (Shear Stress Transport) model has been chosen. In general, SST is the recommended choice for modelling fluids in geometries containing rotating parts.

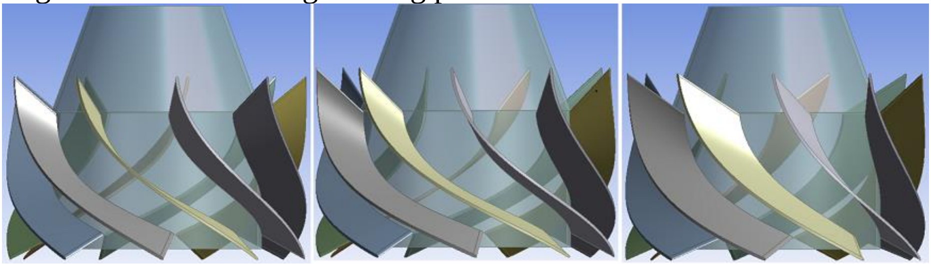

Figure 6: Examples of the stator geometry for various parameters settings

4.3. Optimization and results

Using ANSYS DesignXplorer tool, an initial sampling for the hypercube was created. In the first step, only 11 parameters were considered. Parameters 8 and 12 remained fixed at value of 0.35. The sampling had 151 items, of these 147 were created and computed successfully.

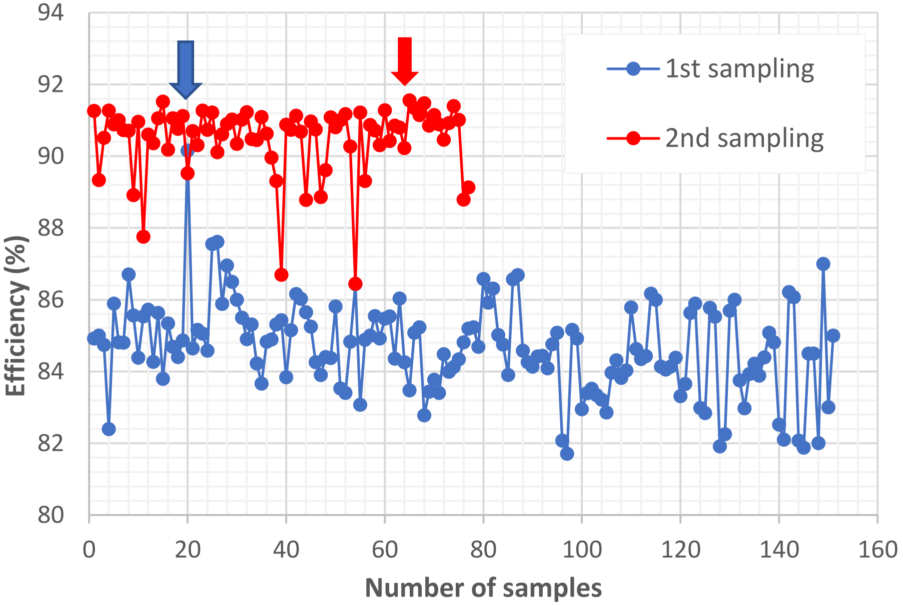

Figure 7: CFD results (efficiency) for the 1st and 2nd sampling. The arrows show the selected (s the best) designs for each sampling.

There were 105 samples in total, and only 77 were successfully generated and computed. Once again, all the CFD simulations were run. The efficiency results can be seen in Figure 7. From the new samples, the best design was selected. It displayed both excellent efficiency and minimal residual circumferential velocity at Q_{OPT}.

After minor modifications (mainly by rounding – i.e. setting the dimensions to sensible values), it was used as the final design. For this design, the full CFD model (all impeller and diffuser passages, and with the outflow elbow) was created, and complete performance curves were evaluated for three different settings (of attack angle) of the impeller blades – 0, 6 and 9 degrees. In total, 21 performance points (3 adjustments, 7 flow rates) were computed. The results can be seen in Figure 8.

Table 1: Comparison of parameter bounds – 1st vs 2nd sampling.

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
1st sampling	1	45	5	45	-20	-75	-60	0.35	-3	-70	-60	0.35	-3
	15	70	100	90	-5	-45	-30	0.35	12	-40	-30	0.35	12
2nd sampling	1	55	5	30	-15	-70	-65	0.25	-10	-70	-65	0.25	-10
	10	67	15	55	-5	-55	-50	0.5	3	-55	-50	0.5	3

As can be seen, the optimized design clearly dominates the old one. The difference ranges from 3% up to 8%. For a high-performance pump, this is a very significant improvement. The real product can vary in size a bit, as it can be scaled down or up in accordance with the laws of hydraulic similarity. The maximum power can exceed 5 MW in its largest form, and for such power consumption, every single per cent of efficiency matters when the pump is operated for a long period of time.

The head has improved by similar margin. This is only natural, as the impeller remained the same during the optimization. I.e. the pressure generated by the impeller is the same, only the energy loss in the diffuser was lowered. In Figure 9, a comparison of backflow areas between the old and optimized design can be seen. There is an evident and significant difference, and the lower dissipated energy translated into higher overall efficiency of the pump.

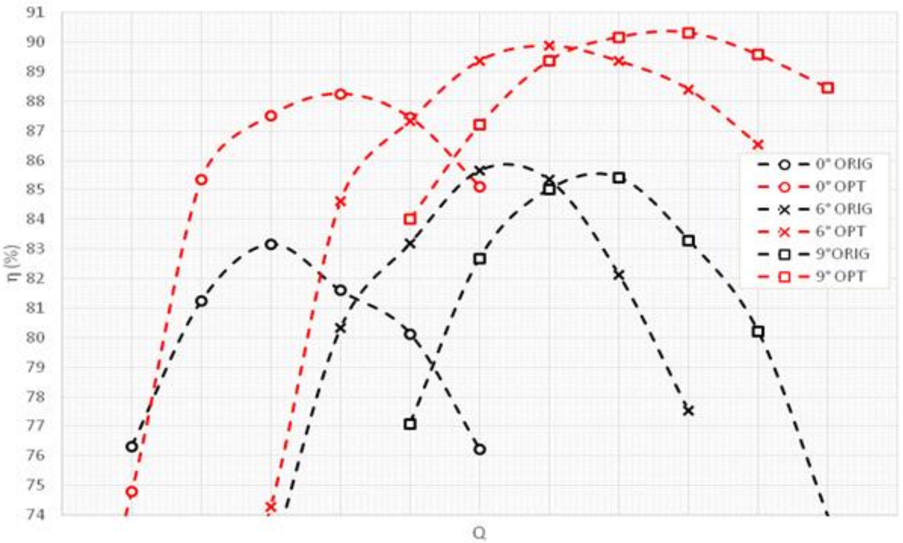

Figure 8: Efficiency for different blade adjustments. The original (black) vs optimized (red) design.

4.4. Summary and conclusion

The automated parametric model was successfully assembled and used as a part of CFD-driven hydraulic design of a pump diffuser. The optimization only worked with a simplified TBR model at Q_{OPT}, but the outcome was very good. The newly designed and optimized diffuser display performance superior to the original one – by 3 % up to 8 % in majority of the working range. It should be noted, however, that randomness can play an important role in the process, as the parametric space and initial sampling influence the results. For the first scenario, only one sample helped guiding the search and further narrowing the parametric space to the promising area.

It is also necessary to mention that considering a single flow rate for the optimization can be risky, as high efficiency at one point (flow rate) does not guarantee a good performance in the whole working range. This means that it is possible obtain an one-point-optimized design with a great maximal efficiency, and still have a poor overall performance in the working range.

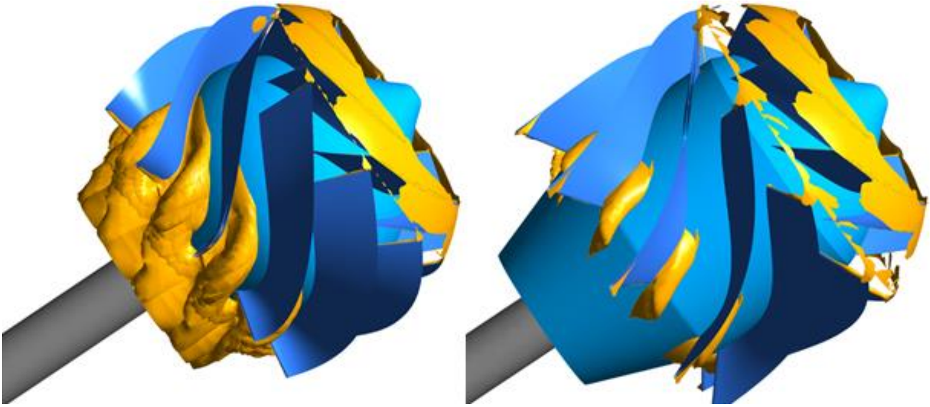

Figure 9: Backflow areas (in yellow) for the original (left) and optimized (right) design.

5. Case 2 - Pump suction

Objectives: Pressure loss and uniformity of velocity profile at the output of the suction part (at Q_{OPT}).
Solution: ANSYS Workbench parametric model driven through text script files, steady-state and transient simulations, Stochastic RBF optimization method
Results: Efficiency increased by 0.5%, $NPSH_3$ by 20%.

5.1. Introduction

In this case, a radial pump of specific speed $n_s = 135$ was considered. In order to improve the suction ability ($NPSH_3$ characteristic) of the pump for a specific application, an inducer[1] was developed. CFD analyses of the pump performance revealed, however, a problem at the inflow part of the pump. For optimal performance, the flow that comes into the rotor (i.e. the inducer in this case) should be perpendicular to the entrance and uniform. In this case, however, the CFD analyses revealed significant variations in the inflow profile.

[1] An inducer is added in front of the impeller and shares the same shaft. Its blades are designed to generate static pressure and help feeding the impeller to help to delay the cavitation occurring there. In exchange, the efficiency of the pump is lowered by a few per cents.

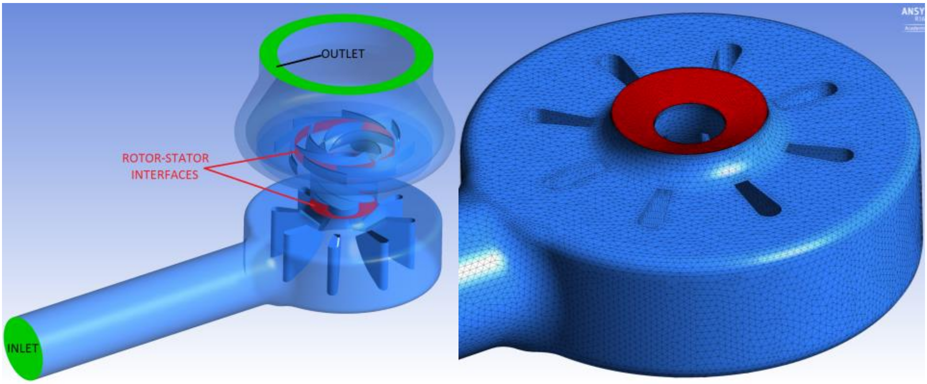

Figure 10: The pump CFD model and the detail of the suction

Based on the data, it was decided that optimizing the suction shape is a good opportunity for improving the pump performance. The expectation was that such optimization can be based on steady-state simulations of the suction only, significantly limiting the computational expense and efforts invested into the automation of the numerical modelling. The plan was to further develop the automation and connect it with a suitable optimization method.

5.2. Parametric model and automation

The parametric creation of the suction was done using ANSYS Workbench and DesignModeler. The approach to the hydraulic design and choice of parameters were mostly based on recommendations in (1). In total, there were 18 geometry parameters. Visualisation of selected parameters can be seen in Figure 11. The computational mesh was created as unstructured, i.e. tetra + prism for the boundary layers. The numerical modelling was fully automated, using ANSYS Workbench, Linux shell and Python scripts and codes. First, the update process in Workbench (DesignModeler, Meshing and CFX/Pre updates) was recorded as a script. By replacing the file name and parameter values by keywords, a source template for the automation was then created.

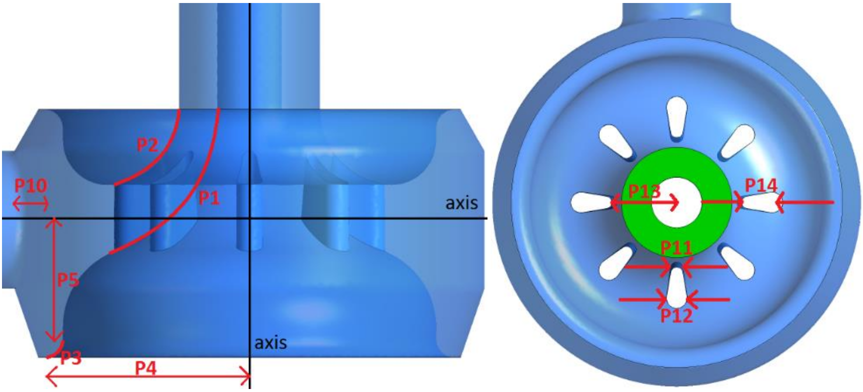

Figure 11: Visualization of selected parameters

The workflow of the automation is as follows:

1. The Python routine accepts csv file with parameter values as an input.
2. The Workbench template is loaded, keywords are replaced by the actual values, and a new script is saved – named after the input csv file.
3. The Workbench script is executed in command line. As a result, a CFX input file is created.
4. For pre-set flow rates (and possibly other solver settings defined in a premade ccl files), the CFX input file is solved in batch mode.
5. For each result file, monitored variables (efficiency, head, …) are extracted and stored in a csv file (with appropriate name).
6. The csv files are processed by a Python script, and efficiencies (for each result file of a particular design) values are saved into a csv file. This csv file then acts as an input for an optimization routine.

The step 4 (the solver) is typically performed on an HPC cluster via a PBS scheduler. This brings many technical challenges to the automation procedure. Apart from difficulties with data transfers between the “global” shared storage and local computation nodes, the simulation can fail for various reasons (licensing problems etc.). To maximize the utilization of the available resources and to make the simulation “flow” more manageable, the CFX model creation and assembly is separated from the solver run. The scripts work like this:

1. For all *wbjn* files (the Workbench scripts) present on the shared storage, the corresponding jobs are created and send to the queue.
2. Once the "model creation and assembly" job starts, it copies the necessary data to the local storage and tries to execute Workbench in the batch mode. As the Workbench run can "freeze" frequently (for various technical reasons), special measures needed to be taken. As the time spent for the generation of the solver file is known quite well (typically between five and ten minutes), the Workbench process is limited to 30 minutes (and killed after that, if necessary). At the end of the script run, a log file is created. Now three situations can occur:
   a. Neither the solver input file nor the log file is created. This means Workbench "froze" and got terminated by the timer. In this case, the job is sent to the queue again.
   b. Only the log file is created. This means the Workbench script was executed successfully, but the geometry or mesh generation failed. In this case, the design is written to a log file on the shared storage as "failed".
   c. The solver input file is created. In this case for each pre-set solver setting (uploaded at the shared storage), the solver jobs are set to the queue (in parallel).
3. The "solver" job copies the necessary data to the local storage and runs the CFX solver with the appropriate model and settings in batch mode. As no problems with "freezing" were observed during the CFX runs, no measures similar to running Workbench needed to be deployed. One the simulation end (typically in ca. 10 minutes), the results are copied back to the shared storage and an "objectives evaluation" script is run.
4. The evaluation script is created in Python, and process the data extracted from the result file. The outcome is an objective function value stored in a csv file. If all the results for the design (i.e. all result files corresponding to the defined list of solver settings) are available, the scalarized objective is generated and

written to a file. The design is also flagged as “finished” in another log file on the shared storage.

With this approach, it is possible to maintain sufficient control over multiple simulations running in parallel, without a central “driving” script. Such script would be technically very difficult to create, as both the user’s control over the scheduler and knowledge about running and finished jobs are very limited. Instead, any optimization code can just check for newly added “finished” jobs in regular intervals.

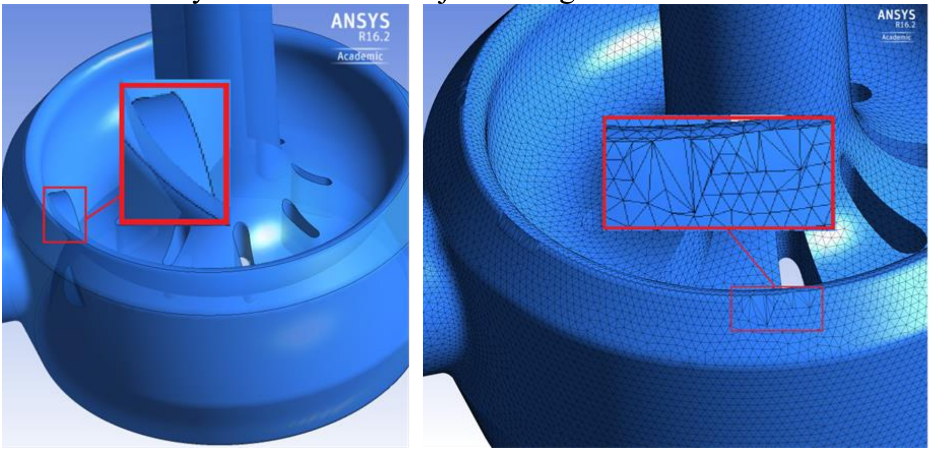

Figure 12: Geometry error (left) and mesh error (right).

5.3. Testing the parametric model

For the start, a relatively large sampling of size 100 was created and used for testing of the parametric geometry. Of these 100 samples, 43 failed due to geometry or mesh errors. This is a common problem faced when creating the parametric model, as ensuring a consistent geometry generation for all possible combinations of the parameters is often a challenging task.

After improving the geometry creation, 88 samples (out of the 100) were successfully generated. This rate was decided to be acceptable.

Next, a smaller sampling, intended for CFD evaluation and testing the parametric space, was created. For the 18 parameters, 38 samples (the default size for 18 parameters in the Stochastic RBF method codes) were created and used as an input for the batch processing.

Table 2: Parameter ranges

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
200	130	30	380	220	5	5	2	2	30	5	10	190	250	5	10	90	250
450	200	50	510	300	50	50	25	25	100	25	45	230	280	35	50	220	280

With this setting, 31 models were successfully generated and 7 failed because of geometry errors. These 31 samples were evaluated by CFD at Q_{OPT}. For the optimization, only the suction, together with prolonged inflow and outflow (for better numerical stability), were considered. As there are no moving parts in the suction, the simulation was set as “steady-state”, significantly faster than the transient option. SST k-ϖ model was used as the turbulence model. The original intention was to use both total pressure loss (denoted as *H*) and outlet velocity uniformity (denoted as *v*) as optimization objectives.

$$H = \frac{Total\ Pressure_{Inlet} - Total\ Pressure_{Outlet}}{\rho_{Water} \cdot g}$$

$$v = \int_{Outlet} \left| v - \ v_{avg} \right|$$

Unfortunately, upon more thorough testing the results proved to be significantly mesh and model dependent. Thus, only the pressure loss at the design flow rate was selected as the cost function.

5.4. The optimization method

When selecting an optimization method for a CFD-driven optimization, one has to consider multiple criteria:

1. The numerical simulations are computationally very expensive, this severely limits the maximum number of the objective function evaluations. This means that a method that can improve the objective(s) fast is preferable to a method that can do better, but only after very high number of (objective function) evaluations.
2. It needs to be a derivative-free method. In special cases and with in-house or open source (such as OpenFOAM) codes, it is possible to use the so-called *adjoint method* (8). But generally, especially for commercial codes such as ANSYS CFX, no

derivatives are available. Numerical differentiation is not a viable option here, due to accuracy limits of numerical methods and high number of parameters.

3. The objective functions are supposed to be multimodal, i.e. global optimization methods are preferable.
4. Failed simulations / crashes can occur for various reasons. I.e. for some input values, the objective function returns no output. The non-existent values can be “faked”, but this can distort the optimum search. Thus, having a method that can handle such situations easily is preferable.

Point 1 is a problem for classical population-based methods, points 2, 3 and 4 practically rule gradient based methods (steepest descent, Newton's method, etc.) out. Currently, the so-called Surrogate-Assisted Optimization (SAO) is the preferred option for computationally expensive engineering optimization. In SAO, the approximation of the objective function (surrogate) is used instead of the objective function itself. In every iteration, the optimization is performed on this (computationally cheap) surrogate. Once the new point for evaluation is decided, it goes through the computationally expensive simulations and the surrogate is updated. SAO has been studied extensively in literature, such as (5), (9) or (10).

For the suction optimization, the so-called Stochastic Radial Basis Function (Stochastic RBF) method, described in (11), (12), was selected for this task. There were two main reasons for the choice:

1. According to the testing done by the authors, it is very competitive when maximum number of evaluations is a concern.
2. Matlab codes were freely available [https://ccse.lbl.gov/people/julianem/index.html], and easy to understand and modify.

It is a single-objective, SAO method, working as follows:

1. Generate initial sampling and evaluate the samples.
2. Use already computed samples and fit the response surface.
3. Generate large number (ca. ten thousand) random testing points to cover the response surface uniformly.

4. From these testing points, select *N* new candidate points for evaluation. The selection is a compromise between exploiting local minima of the response surface and exploring areas further away from the already sampled points.
5. Evaluate the candidate points.
6. Repeat (2) until ending criteria are met.

The number *N* can be selected arbitrarily; the authors recommend either 4 or 8 for optimal performance. To deal with the technical difficulties in real-world application for the shape optimization (namely failed samples and heterogenous HPC cluster environment with a scheduler), the original method was modified to work with something that could be called “pipeline”.

1. Generate initial sampling and evaluate the samples.
2. Use already computed samples and fit the response surface.
3. Generate large number (ca. ten thousand) random testing points to cover the response surface uniformly.
4. Select *N* new candidate points for evaluation and add them to the queue.
5. Wait until at least one computation finishes. Then wait for a few more minutes and read all available results. (Due to the nature of the scheduler system, the results often come within very short interval.)
6. Select k new points, where *k = number of samples computed + number of samples failed*.
7. Add the newly selected samples to queue.
8. Repeat (5’) until ending criteria are met.

This way, the failed samples are simply ignored, and the “slower” ones are just used once they finish. Because of this, there are no bottlenecks caused by waiting for “stuck” computations. As the method always work with “all data available”, it is possible to insert new samples manually. It is thus for example possible to manually “tune” a selected design (an experienced hydraulic expert can often do this) and add it to the already evaluated data.

5.5. Optimization and results

The optimization continued from the results of the initial sampling. The number of newly generated samples was set to *N = 4*. The optimization was ended manually once no significant improvement in the objective have been observed for multiple iterations. The results can be seen in Figure 13. In this case, the optimization has shown hardly an improvement over the random sampling. As the results were not convincing, though, another, more complex parametric model was created.

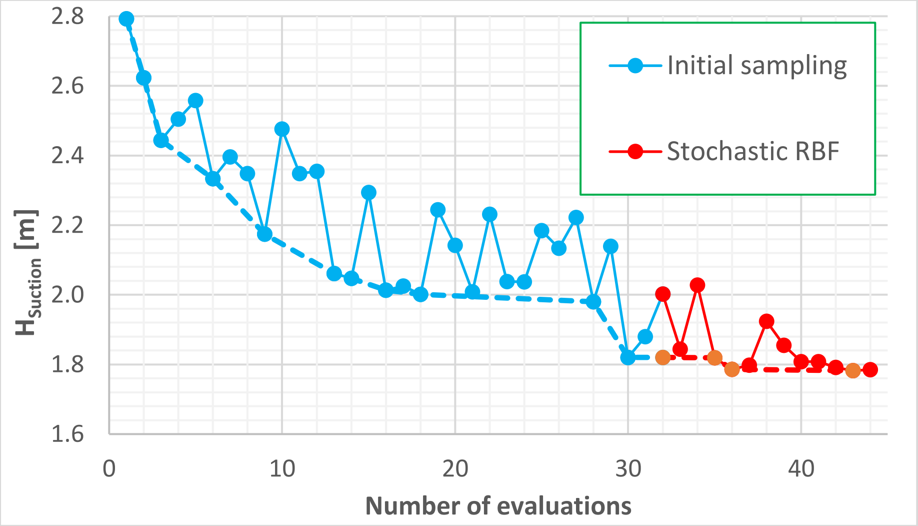

Figure 13: Objective function progress. The bold line connects the best values reached.

5.6. Parametric model – suction and inducer

After analysing the outcome of the optimization, the inducer was added to the parametric model. As the next step, a different way of evaluating the velocity uniformity was chosen. The inducer was added to the model, and the simulation was set as transient.

The inducer domain rotated around the *z* axis, with *rotor-stator* interface to the static suction domain. These changes were supposed to help in capturing the velocity objective *v* with higher accuracy. Inflow and outflow parts were again modified to improve the numerical stability.

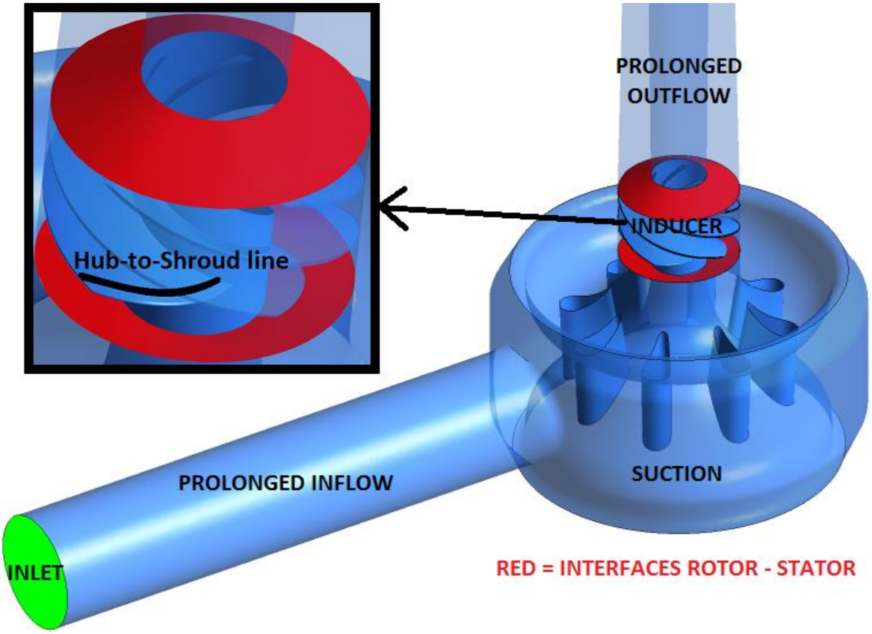

Figure 14: Parametric CFD model with the inducer and a detail of the Hub-to-Shroud line.

Using ANSYS CFD-Post built-in functions, the Hub-to-Shroud line was defined and meridional velocity v_m was exported in multiple points. The objective function was then defined as:

$$J = \sum_{i=1}^{N} \frac{\left(v_m^{(i)} - \bar{v}_m\right)^2}{\bar{v}_m}$$

where $\bar{v}_m = \frac{1}{N} \cdot \sum_{i=1}^{N} v_m^{(i)}$ is the average meridional velocity on the selected Hub-to-Shroud line. The number of points *N* was selected as *32*. The boundary conditions and turbulence model remained the same as in the previous case. The transient simulation gives more accurate results, as the rotor / stator interaction and inertia forces are considered. However, it also brings a steep increase in computational time and adds complexity to the automation process and post-processing. In a steady state simulation, only one “converged” value is used for the objective function. In the transient case, average values (typically over two last rotations of the impeller) need to be used. Unlike head, the *v* values were not averaged over time interval. Instead, only the last value was considered. This was a limitation forced by technical reasons, as processing the data with CFD-Post every timestep would be computationally too taxing.

With objective function defined this way, the intention was to ensure as uniform fluid entrance to the inducer as possible. Supposedly, this should lead to better pressure distribution along the blades and slower creation of the cavitational areas.

5.7. Optimization and results – suction plus inducer

The optimization was started from the initial sampling. The number of samples remained the same, i.e. *38*. The geometries for all the initial sampling were generated and computed on an HPC cluster. Then, started from these results, the Stochastic RBF optimization was performed. The procedure was set to generate 4 new samples in each iteration (same as in previous case), and run in an HPC environment.

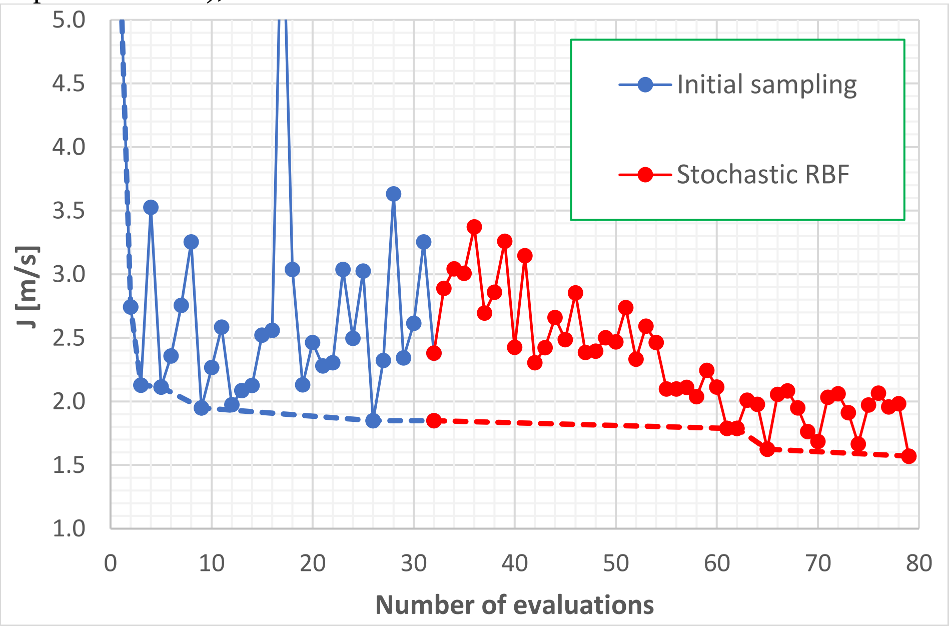

Figure 15: Optimization progress. The dashed line connects the best values.

For the best design, the complete pump CFD model was assembled and full set of simulations was performed. The geometries and results comparison between the original and optimized designs can be seen in the following section.

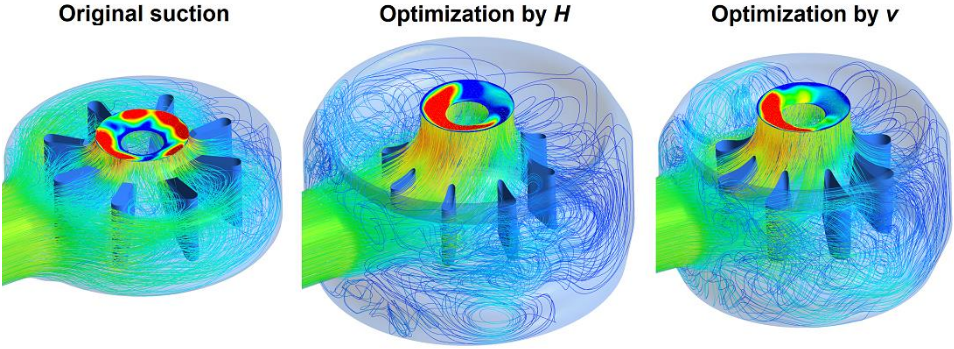

Figure 16: Suction designs and results of CFD analyses. The colours show differences in velocity profile at the suction outlet.

Quite unexpectedly, this objective function yielded inferior results to the pressure drop version. Further analysis of the results has revealed that the suction shape has very little to no effect on the velocity profile in the middle and trailing parts of the inducer blades. Due to this, the shape optimization with respect to the meridional velocity uniformity has no significant effect once the cavitational areas start to develop. Lowering pressure loss in the suction, on the other hand, gives more NPSH reserve.

5.8. Summary and conclusion

The automated parametric model was successfully created, tested and connected with a selected optimization method. Two different approaches were tested. First, a simple model with the suction only and set as computationally (relatively) inexpensive *steady-state* simulations. Second, a more complex model with the rotating inducer included, and using the computationally demanding transient simulations. In the end, the simpler model worked out better. The results of the numerical simulations of a complete pump suggest the model with suction optimized for $H_{Suction}$ displays both the best efficiency and cavitational properties. The objective defined by velocity uniformity did not work very well, possibly because of lower accuracy on the rotor-stator interface. Overall, results of this optimization were not convincing.

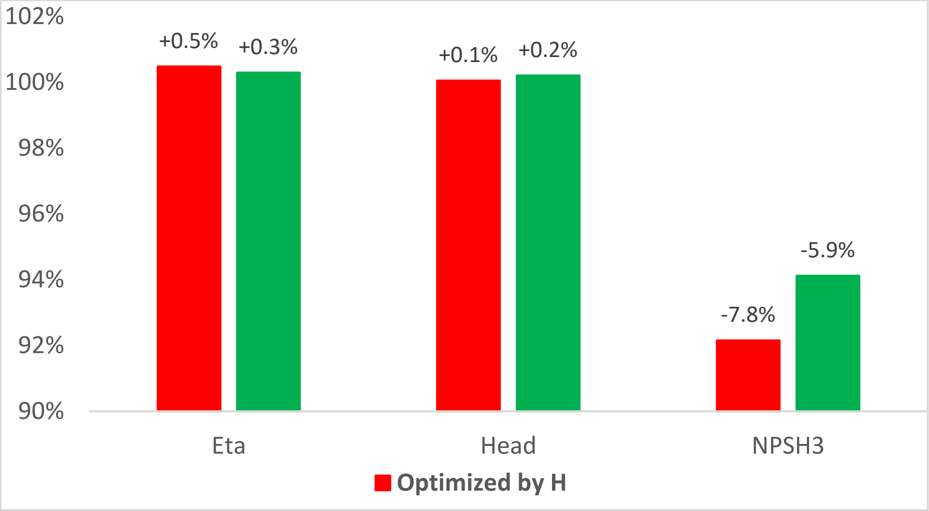

Figure 17: Hydraulic and cavitational performance of the optimized designs. The results are related to the original suction design.

6. Case 3 - multi-objective optimization of a stator

Objectives: Pump efficiency at working range.
Solution: ANSYS Workbench parametric model driven through text script files, transient simulation (full geometry). Three objectives (efficiencies at 90%, 100% and 110% of Q_{OPT}). Stochastic RBF optimization method + scalarization.
Results: Efficiency increased from 2% up to 5% in the working range.

6.1. Introduction

The next case for stator optimization was a diagonal pump with adjustable blades, of specific speed n_s = 360. It was an already existing, but outdated pump from SIGMA. When compared to the newer designs, the efficiency was not competitive. The numerical simulations identified the diffuser to be responsible for significant part of the efficiency loss.

The impeller efficiency was very high (over 90 %), but the circumferential component of the velocity remained very high through the diffuser. This increased the hydraulic losses in the outflow part, lowering the hydraulic performance of the pump. Based on previous experience with diffuser optimization, it was decided to do another diffuser optimization. Unlike in the first studied case, the plan was to perform an optimization in multiple flow rates – to ensure a good performance in the whole working range.

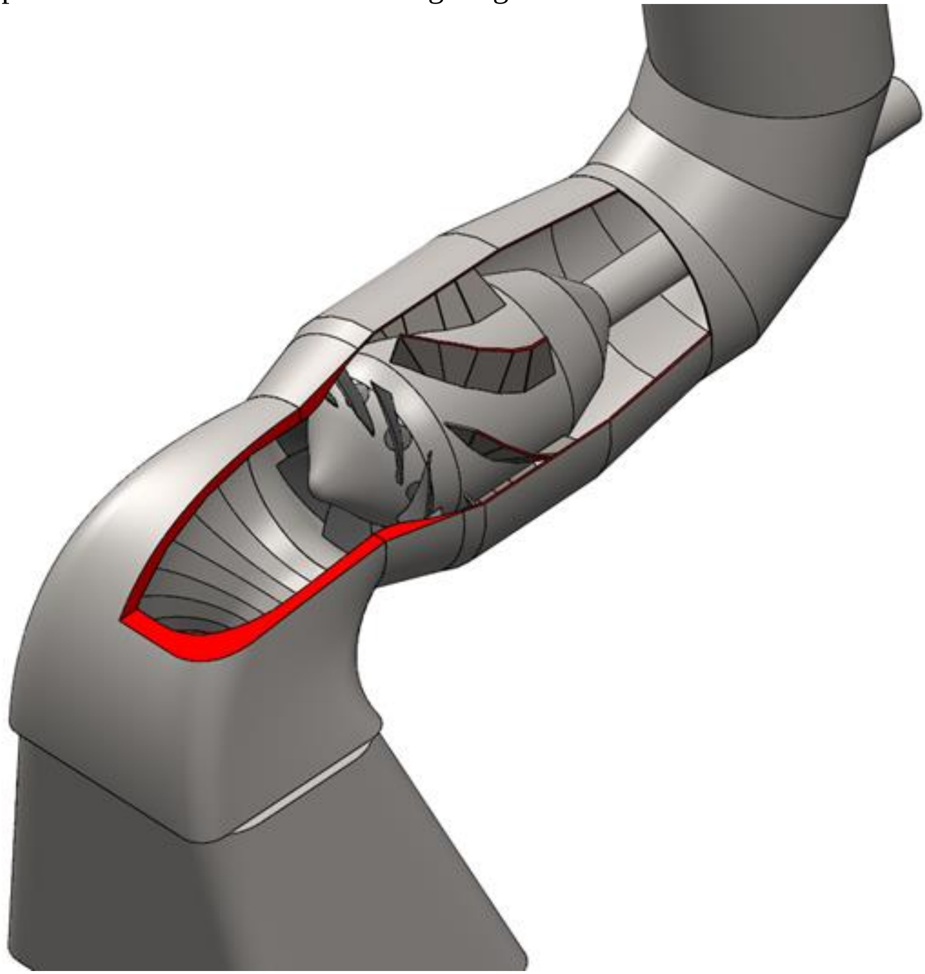

Figure 18: 3D model of the pump with bent diffuser blades

Just as in the previous case, the plan was to produce a design that can be manufactured by the so-called *metal sheets bending*. I.e. relatively simple blades (with constant thickness, defined on two camber lines) and "straight" meridional profile. But this time the intention was to consider multiple flow rates and the Stochastic RBF method. It should be noted that in the ideal case, the actual bent shape should be considered for optimization. The bending in general has a small negative impact on the hydraulic performance, as sharp edges are created on the blade. For each blade shape, there are practical limits for the bending. As a result, it can happen that one shape of the blade is deformed more than another, resulting in bigger performance drop. However, for the optimization this needs to be ignored, as doing the optimization with respect to the bent shape would pose a major challenge. This is because the software tools used for the parametric model creation are not suitable for such option. I.e. both geometry creation and mesh generation cannot be automated in the bent shape. Thus, the optimization was based on the assumption that the resulting design will display performance good enough to have a reserve for the decrease caused by the bending.

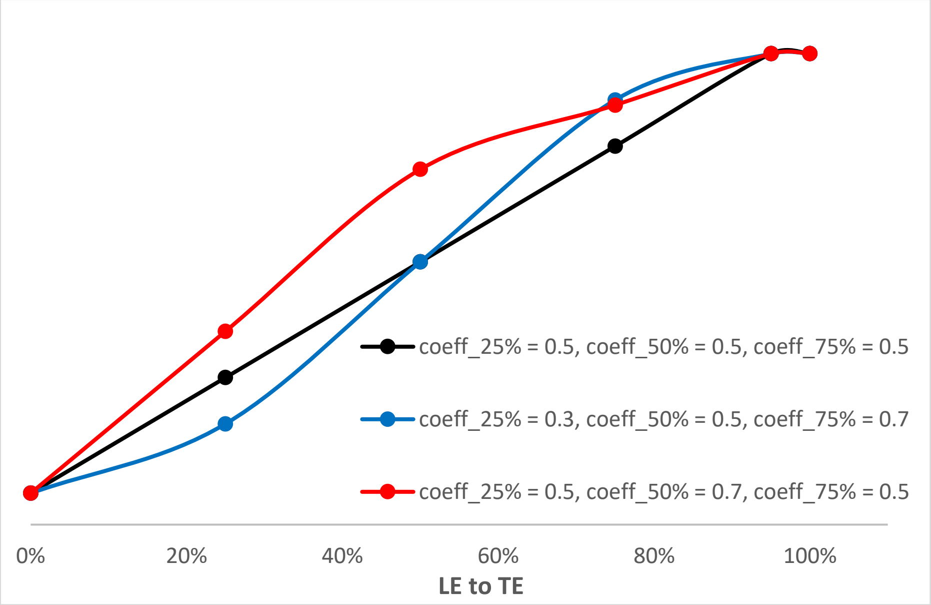

Figure 19: Examples of angle distribution for selected coefficient settings

Another empirically verified assumption was that only one selected impeller blades adjustment can be considered for the optimization. I.e. if the pump displays good performance for one setting, then it performs well for a range of adjustments.

6.2. Parametric model and automation

For geometry and mesh parametric creation, a WorkBench project was created. The process of setting the parameters, geometry and mesh creation, and consequent CFD model assembly was recorded as a template script. The objective function, used for the optimization, then simply replaces keywords in the template by the input parameters, calls WorkBench in batch mode, does the CFD simulations, extracts and evaluate the results and sends the objective value(s) back to the optimization routine. The geometry shape was driven by 18 parameters. The beta angles (at both hub and shroud) were defined at leading edge, trailing edge and three inner points (25%, 50% and 75% of the blade length). As it is often recommended to “hold” the angle towards the blade end, the value at 95 % was considered the same as at the trailing edge.

Table 3: Parameter ranges

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
1st sampling	30	0.3	0.3	0.3		15	0.3	0.3	0.3		700	700	15	20	20	-10	25	25
	40	0.7	0.7	0.7		35	0.7	0.7	0.7		1300	1300	25	200	200	15	150	150
2nd sampling	31	0.3	0.25	0.25	80	17	0.5	0.4	0.5	80	1100	1100	15	20	20	-10	25	25
	37	0.5	0.45	0.4	90	23	0.7	0.6	0.7	90	1300	1300	25	200	200	15	100	100

ANSYS DesignModeler was used for diffuser and outflow geometry creation. The structured impeller mesh was generated by TurboGrid, and ANSYS Meshing was used for elbow meshing. The mesh size was ca 1.5 mil. Nodes, and solution time ca. 20 hours. With a WorkBench script, the CFD model is then updated and solver input file created. The solver is run for each desired setting, defined in a ccl file. Once the run finishes, the results are extracted, processed and sent as output. As each solver setting runs independently on the others, it is possible to run multiple setting at once to speed up the optimization process.

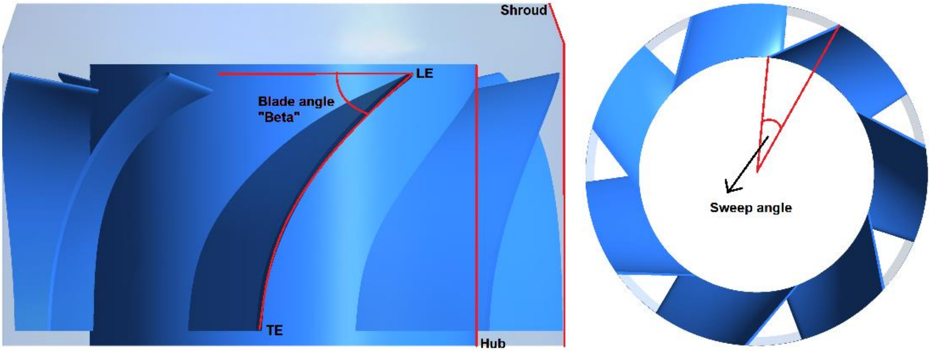

Figure 20: Visualisation of selected parameters

6.3. Testing the parametric model

First, the trailing edges were set as fixed to 90 degrees. For the 16 remaining parameters, the initial sampling was created. The default size of the Stochastic RBF method was used, i.e. $2 * (N + 1) = 34$ samples, where N = 16 is the number of parameters. Out of the 34 samples, 25 were successfully created. The remaining samples failed for geometry reasons.

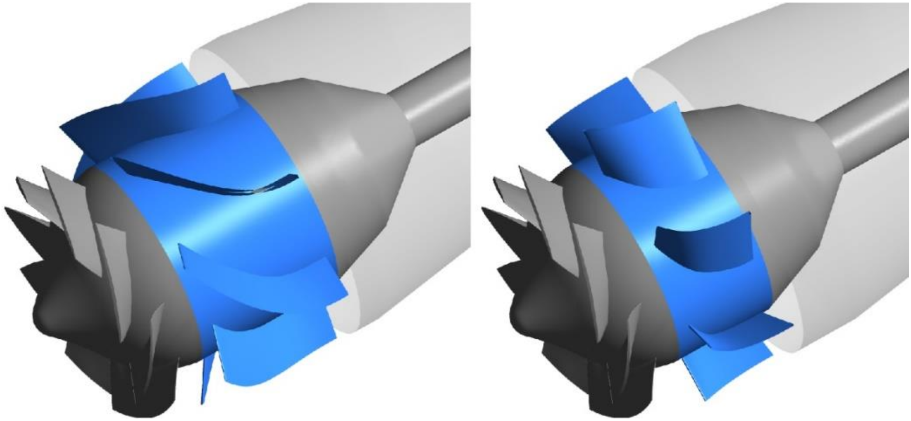

Figure 21: Examples of diffuser geometry (in blue).

All the samples were evaluated at three different points (flow rates) – *90%*, *100%* and *110%* of the design point Q_{OPT}. After careful evaluation and consideration of the results, the parametric model was adjusted. First, the beta angles at trailing edge were enabled as parameters, instead of being fixed to a constant value. Second, the ranges of the parameters were modified, mostly narrowed. The comparison can be seen in Table 3. With these changes, a new sampling of size 38 was created. Out of it, 36 solver input files were successfully generated. After CFD simulations at the three specified flow rates, this initial sampling was used for starting the optimization.

6.4. Optimization and results

The same method as in the previous case (“Case 2”) - *Stochastic RBF* – was used. As it is a single objective optimization method, the scalarization of the three objectives (efficiencies at 90%, 100% and 110% of Q_{OPT}, further denoted as *η_{90}, η_{100} and η_{110}*) had to be used. Based on recommended scalarization methods in (13), the objective function was defined like this:

$$J = max(90 - \eta_{90}, 91 - \eta_{100}, 88 - \eta_{110}) + \frac{\eta_{90} + \eta_{100} + \eta_{110}}{1000}$$

90, *91* and *88* were “thresholds”, i.e. efficiencies required to be exceeded (at the respective flow rates). By considering the maximum of the “missing efficiencies”, the optimization is basically forced to try to meet all the required thresholds. Unlike a scalarization by summing the objectives, where gain in one objective can outweigh the loss in another. The small additive member in the scalarized function is usually recommended (13) to help guiding the optimization in a case when an objective is improved, but the maximum remains the same.

The optimization was performed at an HPC cluster. The limiting factor for the parallelization was the number of available ANSYS CFX licences, there were 25 of them. Thus, for maximum performance, Stochastic RBF was set to generate 8 new samples in every iteration. Three flow rates mean three working points per sample, this means 24 CFD simulation per optimization iteration – each utilizing 16 CPU cores. There were 8 Stochastic RBF iterations in total. Out of the 64 samples, 11 failed and 53 was successfully generated and evaluated. The optimization was stopped once there were no significant improvements over successive iterations. The optimization record (efficiency objectives and the single scalarized objective) can be seen in Figure 22.

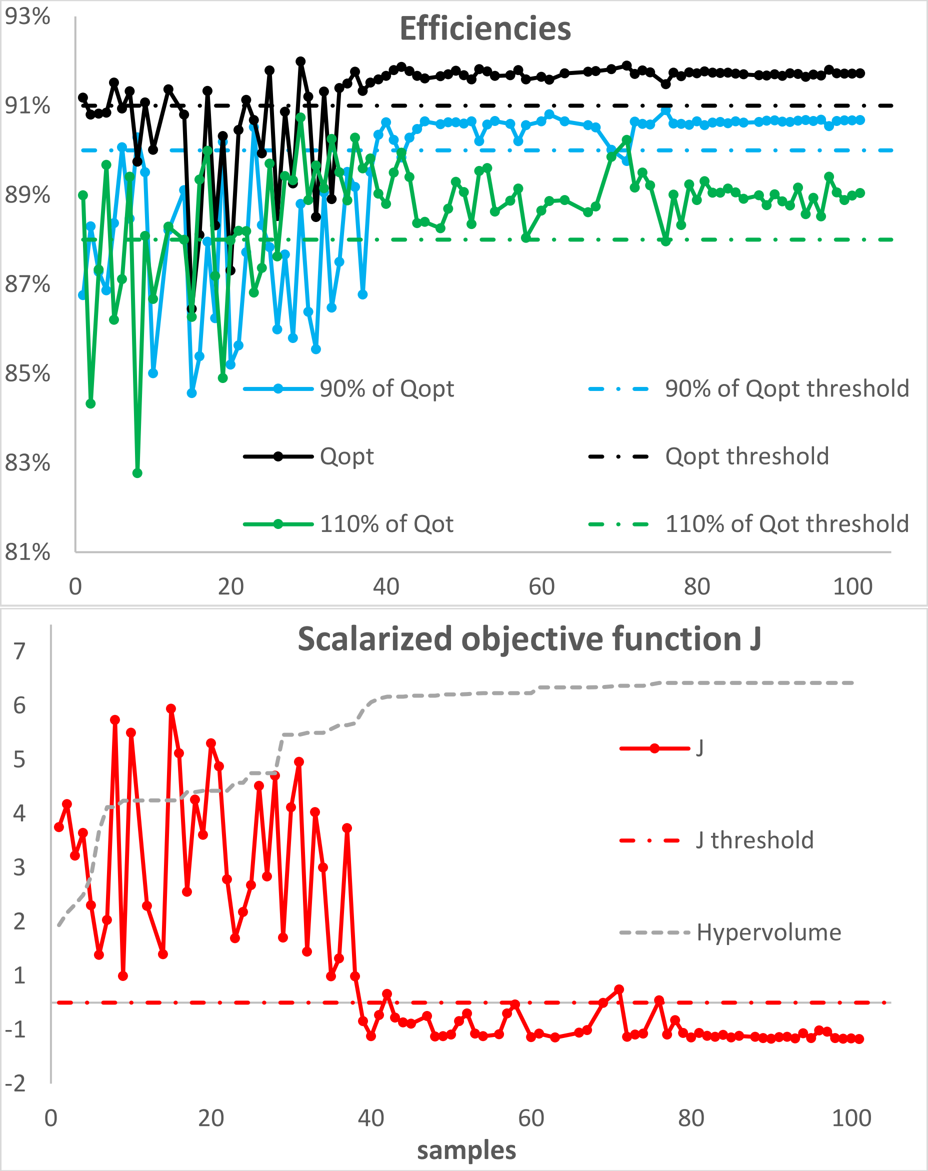

Figure 22: The objectives and scalarized function during the optimization. The grey dashed line shows the hypervolume of the Pareto front.

The optimization successfully exceeded the efficiency thresholds in all three flow rates. For the final comparison, the design found by the Stochastic RBF was then modified for the manufacturing (by sheet metal bending technology). For all considered blade adjustments (-3, 0, 3, 6, 9 and 12 degrees), the meshes were created, and full sets of CFD simulations were performed. The comparison of *Q-Efficiency* characteristics between the pump with original diffuser and optimized diffuser with the bent blades can be seen in Figure 23.

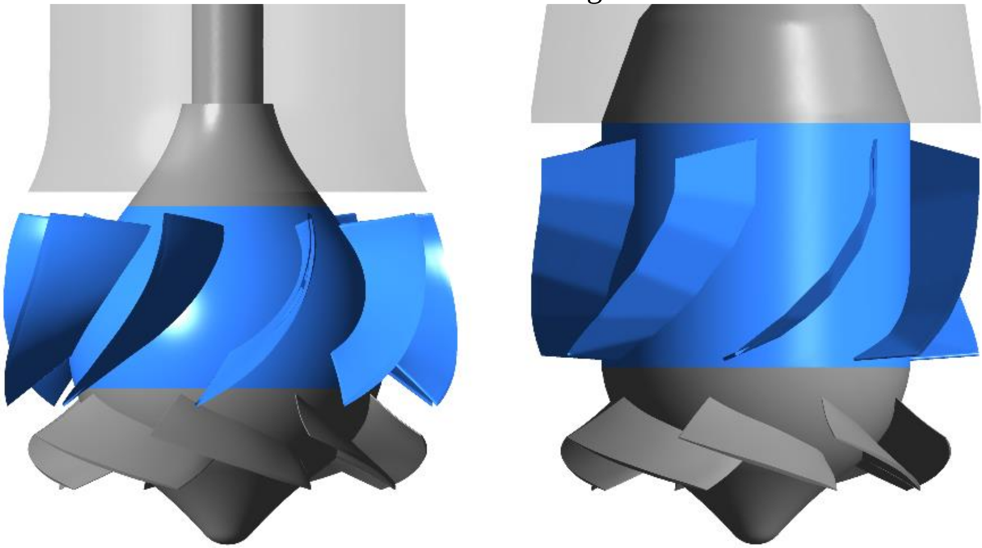

Figure 23: Comparison of original (left) and optimized (right) design.

The improvements vary from 1% up to ca. 7%. The optimized design has a bit narrower operation area with high efficiency. For a pump with adjustable blades this is not a problem, as the setting can be adjusted to the optimal configuration for any scenario. Thus, in practical application, the performance improvements are in a range from 4 to 7%.

6.5. Conclusion

The automated optimization was successfully used as part of the hydraulic design process. The parametric model was created, connected to a Matlab optimization code and run in an HPC environment. The scalarization approach worked reasonably well, and the efficiency was improved for the whole relevant working range. The final sets of numerical simulations for various impeller blades adjustments (and with the bent diffuser blades) confirmed the simplifications of the CFD model (not considering the bent blades, only one impeller blades adjustment) as justified. Overall, we can conclude such approach to diffuser optimisation works reasonably well, if the necessary computational resources are available.

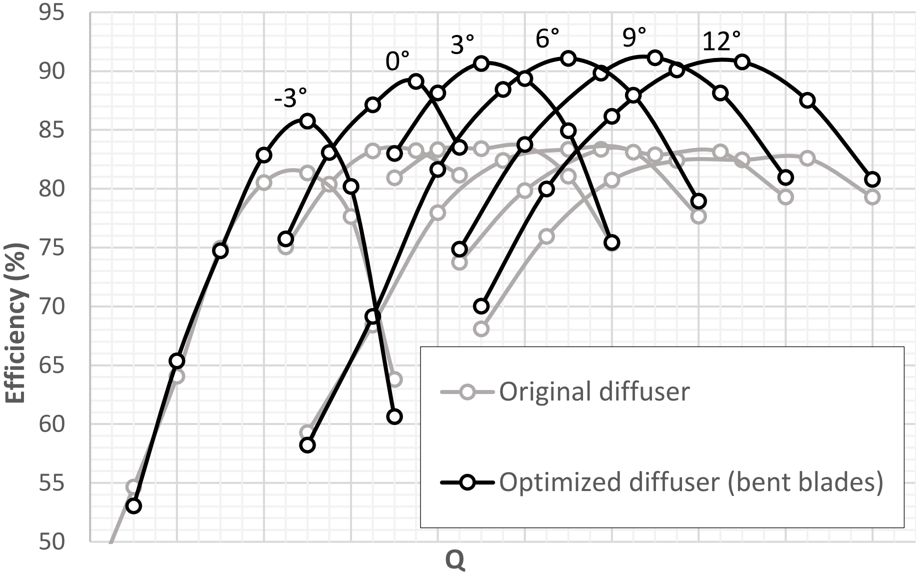

Figure 24: Efficiency comparison with the old design

7. Case 4 - multi-objective optimization of a stator

Objectives: Improving efficiency for the sub-optimal flow rates, maintaining it at the higher flow rates.
Solution: ANSYS Workbench parametric model driven through text script files, transient simulation (full geometry). Three objectives (efficiencies at 80%, 100% and 120% of Q_{OPT}). Stochastic RBF optimization method + scalarization, K-RVEA.
Results: Efficiency increased from 2% up to 5% in the working range.

7.1. Introduction

Another case for stator optimization was a diagonal pump with axial diffuser, of a specific speed n_s = *200*. The hydraulic design was challenging, as the operation site constraints forced relatively short total length of the pump. This caused some troubles, as the short meridional length did not really fit into the recommended ranges for this specific speed. The customer also required high efficiency in a wide working range. The resulting design, created by an experienced hydraulic expert, met these demands. However, the CFD simulations suggested a sharp decrease of performance at ca. 80 % of the design point Q_{OPT}. It was thus decided to perform an optimization of the design.

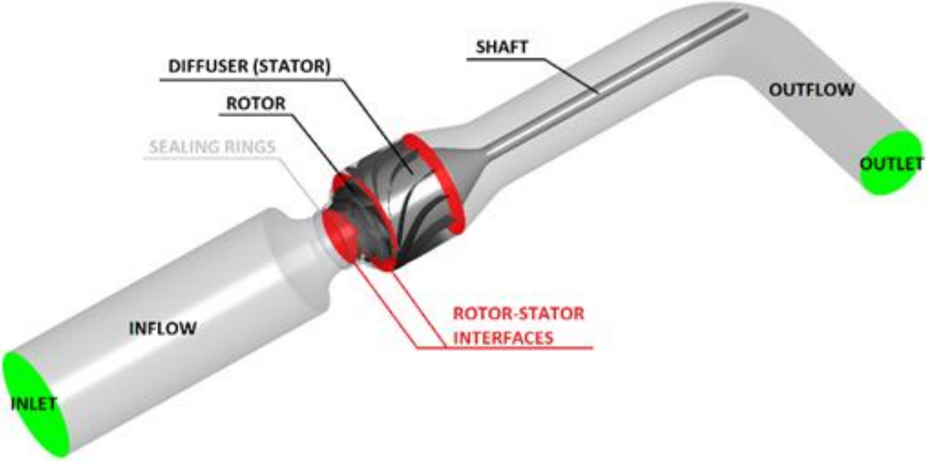

Figure 25: CFD model of the pump.

Based on the previous experience, the optimization was focused on the diffuser. Efficiencies at three different points were chosen as the objectives. First, the Stochastic RBF method and scalarized objective function were used. Next, a more modern, Kriging-based multi-objective method, K-RVEA, was used. For the optimization, a simplified CFD model without sealing rings was considered. Then, a more complex model was assembled, and the optimized design performance was thoroughly evaluated in the whole working range of the pump. In the conclusion, there results were compared with the old design.

7.2. Parametric model and automation

The parametric model was created in a similar way as in the previous diffuser optimization case. I.e. an ANSYS Workbench project, driven by Python-controlled scripts. As can be seen in Figure 26, the sealing rings were also included in the model. The flow from the sealing rings can enter the passage in a direction perpendicular to the "main" flow, and influence the rotor-stator interaction. The sealing rings are rarely included in CFD simulations, as the geometry and mesh creation are challenging. Instead, the effect on pump performance is estimated empirically.

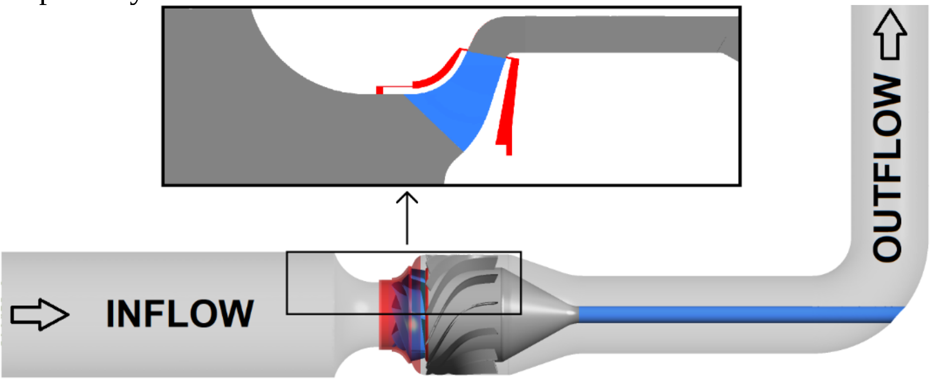

Figure 26: Meridional section of the CFD model. Rotating parts (impeller and shaft) are displayed in blue, the sealing rings are in red.

For the parametric model, the leakage part had to be omitted. Creating a parametric model of the leakage and its automation would be very difficult, as TurboGrid cannot be used for this. It was also assumed that this would not affect the comparison of different design significantly. For these reasons, the CFD model for optimization was simplified in this way.

The CFD model was set as fully-transient, with 154 time-steps per impeller rotations. This equals to 2.34 degrees of an impeller rotation during each time-step. The pump has 995 rpm; thus, the time-step value was $\Delta t = 60 / (154 \cdot 995) = 0.00039157s$. Boundary conditions (BC) were set as Mass Flow Rate at the outlet and Total Pressure at the inlet. As the turbulence model, *SST k-Ω* was used.

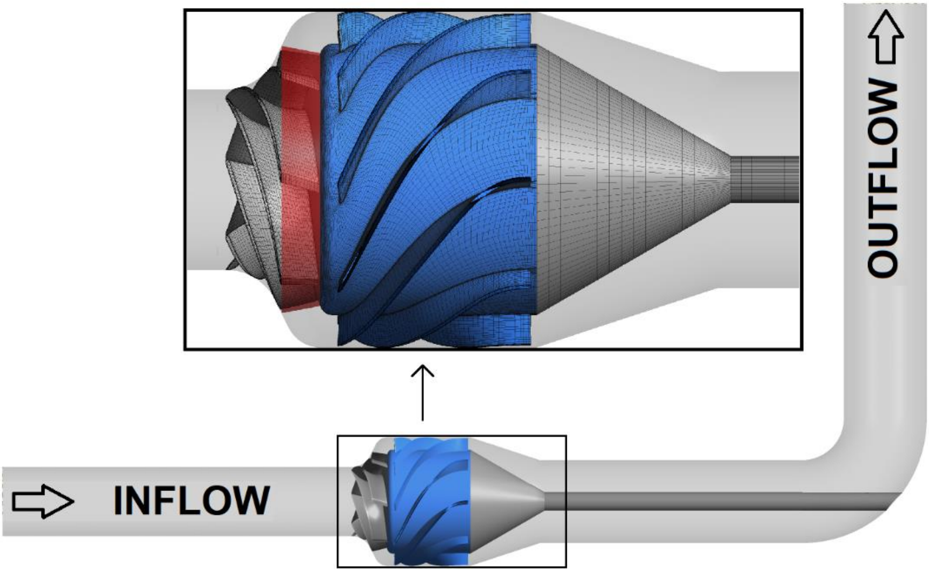

Figure 27: CFD model used for optimization. The diffuser is in blue, impeller-diffuser interface is in red.

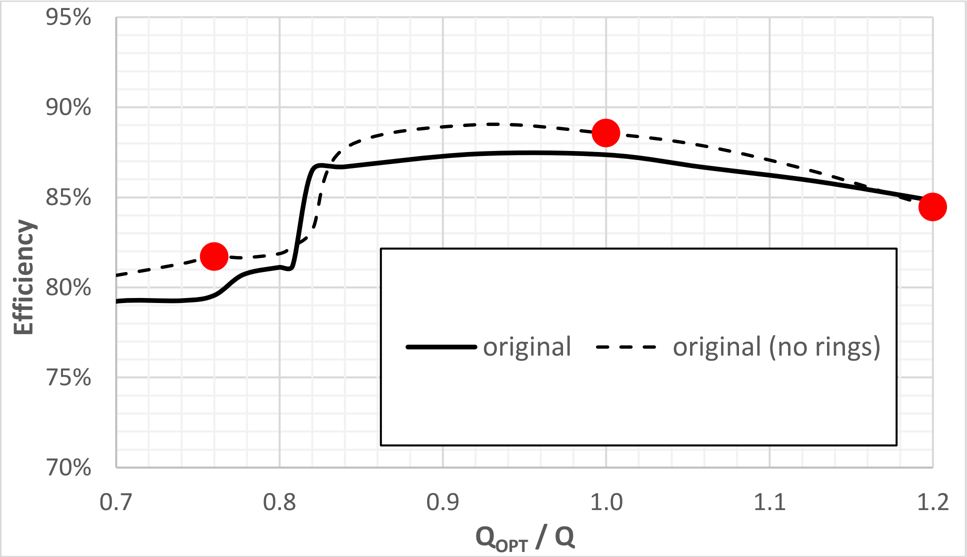

Figure 28: CFD results – with and without rings. The red dots show the three flow rates at which the efficiency was optimized

The simplified CFD model with diffuser detail can be seen in Figure 27, and the comparison of results between the models with and without sealing rings can be seen in Figure 28. In total, there were 22 geometric parameters. Some of the parameters are displayed in Figure 29.

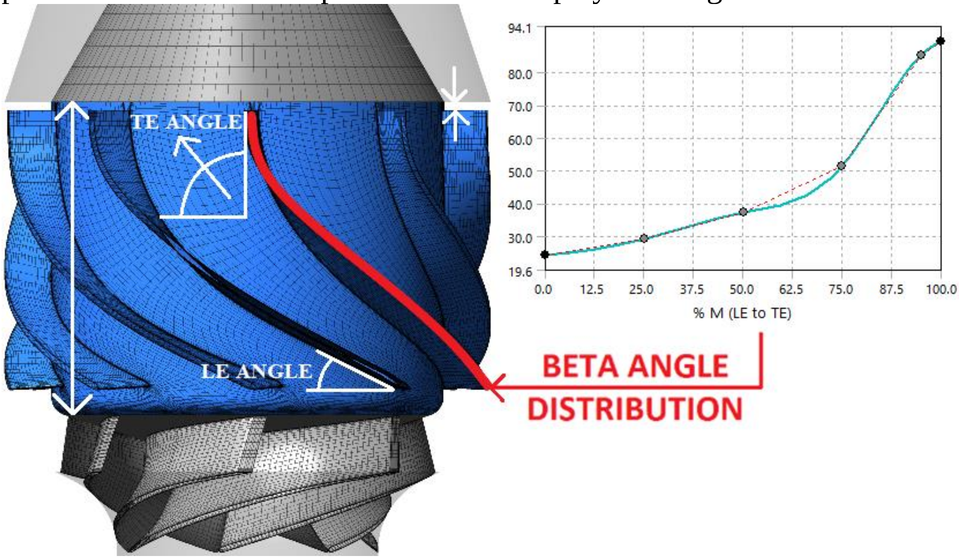

Figure 29: Parameters driving the diffuser shape

7.3. Optimization run 1 – Stochastic RBF

For the designs of the initial sampling (created by the “premade-topology” approach), the CFD simulations were run for the three selected flow rates (76 %, 100 % and 120 % of Q_{OPT}). Just as in the previous case, the three objectives were selected as reasonable compromise between computational costs and ensuring good performance (of the pump) in the whole working range. 76 % of Q_{OPT} was placed right below the problematic in-stability flow rate of the original design. After checking these initial results, the optimization was continued with a scalarized objective function defined like:

$$J = max(84 - \eta_{76}, 90 - \eta_{100}, 84 - \eta_{120})$$

The *threshold* values *84*, *90* and *84* (per cents of efficiency) for the respective flow rates were believed to be enough to ensure improvement over the original design.

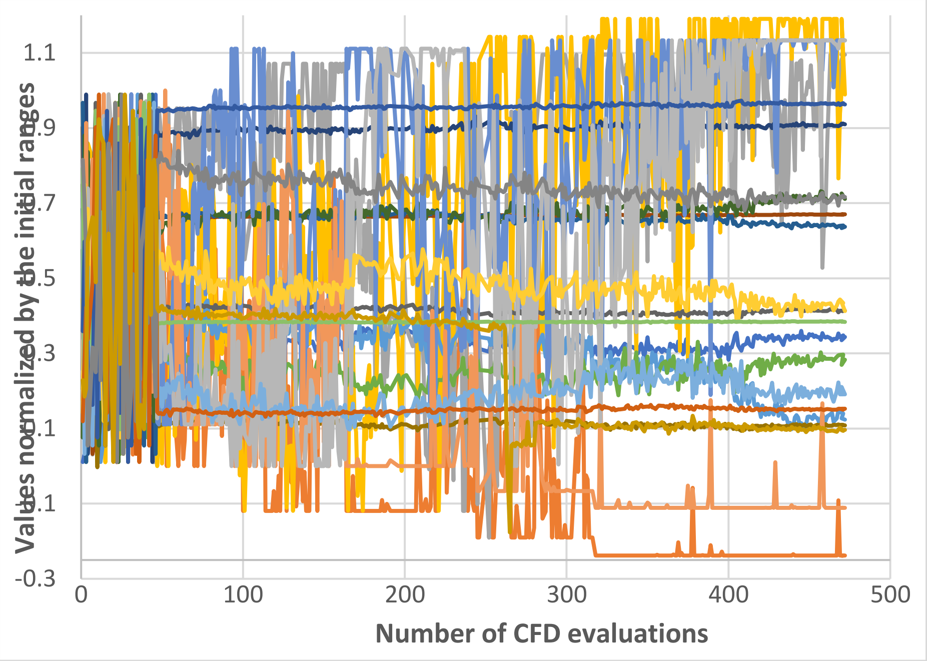

Figure 30: Parameter ranges normalized by the starting values.

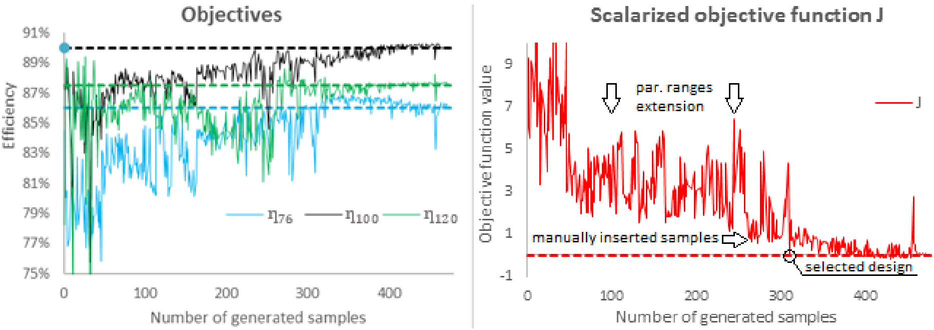

Figure 31: Objectives during optimization. The dashed lines show the optimization thresholds.

During the course of the optimization, some modifications were made to the settings. First, some of the parameters seemed to converge to the boundary values – indicating the results could be further improved outside of the originally specified hypercube. As the Stochastic RBF method performs update of the surrogates every iteration, extending the bounds is relatively straightforward. Of course, it can be assumed that RBF approximation of the extended areas is sub-optimal. Still, for a lack of better information, such option was preferred to running a new initial sampling and optimization. Overall, the parameter ranges were modified twice during the course of the optimization. Further, analysing the available results, the objective *thresholds* were modified, too. The new objective function was:

$$J_{(2)} = max(86 - \eta_{76}, 90 - \eta_{100}, 87.5 - \eta_{120})$$

Also, as an additional information, a manually tuned sample was inserted to the list of computed samples. As the final design, sample number *311* was selected. The results were compared with the initial design, on both simplified and more complex geometry. The Stochastic RBF optimization took more iterations than expected, but the results were promising. The newly found design dominated the old one, and the more thorough CFD simulations confirmed the results. In the whole working range, the efficiency was improved over the old design.

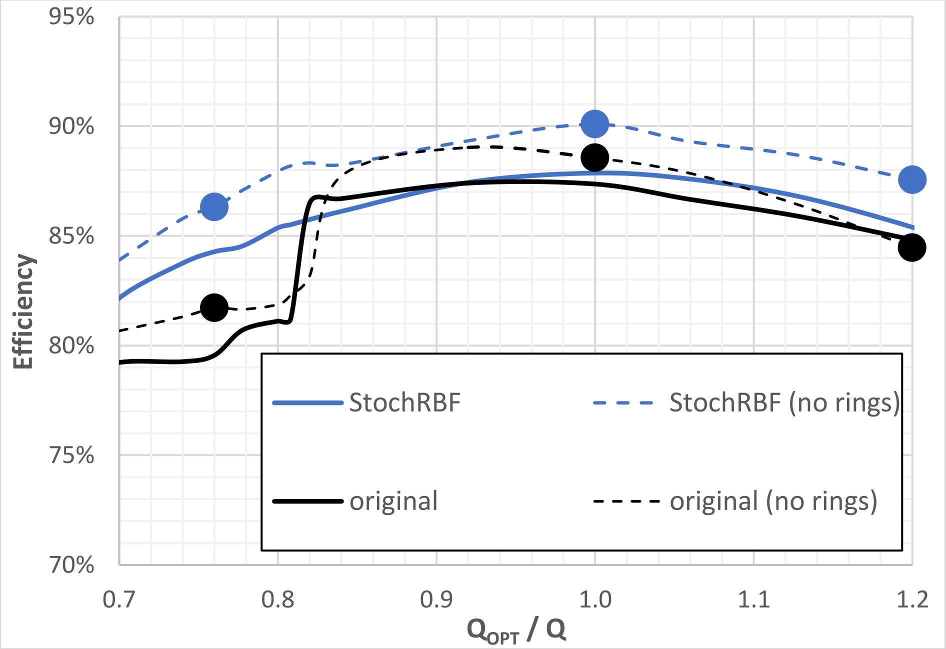

Figure 32: Performance of the original and Stochastic RBF - optimized designs. The dots show optimization objectives

7.4. Optimization run 2 – K-RVEA

As the next step, a more advanced, truly multiobjective optimization method was used. Kriging-assisted evolutionary multiobjective optimisation (K-RVEA) was introduced in (14). It is an optimization algorithm designed to solve computationally expensive multiobjective optimization problems. It is based on RVEA, introduced in (15). RVEA uses the reference vectors in the objective space in order to guide the population development. The goal of this is to promote exploration in the search, and to ensure an efficient search of the Pareto front. In K-RVEA, this approach is further enhanced by Kriging model for the surrogates. The objective spaces are approximated (separately for each objective) and RVEA operates on these approximations. This helps to speed up the optimization process, and less computationally expensive evaluations of the objective functions is then required (16).

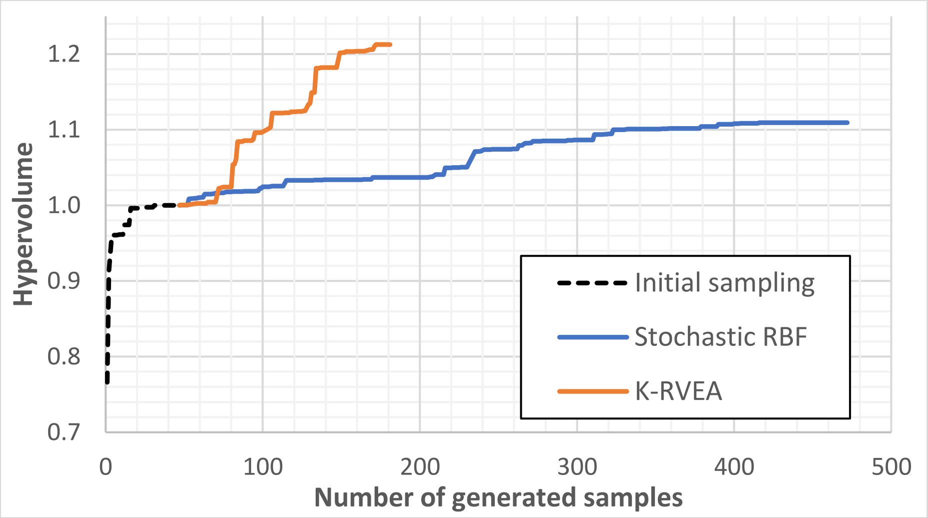

Figure 33: Normalized hypervolume in the objective space - Stochastic RBF vs K-RVEA

For a better comparison, the optimization was started from the same initial sampling. In each iteration, three new samples were generated. This number was set with an intention to avoid getting situations where the geometry or mesh generation fails for all the new samples, and also for improving the utilization of available computational resources.

Thanks to its more sophisticated design, K-RVEA displayed superior performance to Stochastic RBF. As can be seen in Figure 33, the hypervolume of objective values (of the design found by the optimization) grows much faster for K-RVEA. This means that K-RVEA is able to guide the search much more efficiently, probably due to more sophisticated way of maintaining the balance between exploitation and exploration, as it uses the reference vectors, i.e. information from the objective space, to its advantage. Stochastic RBF, on the other hand, only works with the information about the samples in the parametric space. Thus, it needs more time to get from a *local-minima trap*. Of course, with the scalarization approach to the objective function it is also more dependent on the scalarization coefficients.

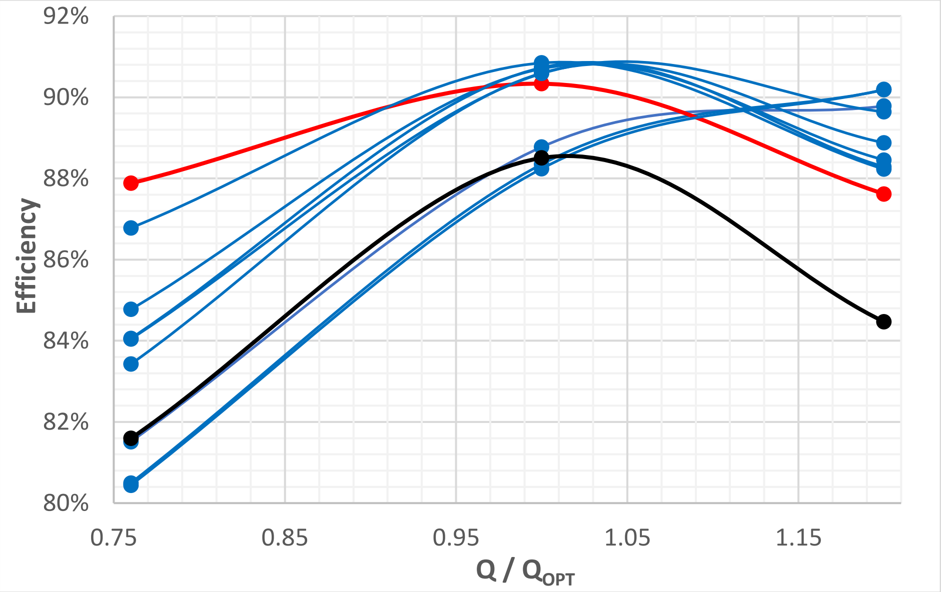

Figure 34: Nondominated solutions found by K-RVEA. Black line is for the old design, and the red line shows the design selected as the best one.

The nondominated solutions found by K-RVEA are shown in Figure 34. Multiple designs displayed performance superior to the old one. As the main goal of the optimization was to improve the efficiency of the pump in the sub-optimal flow rates (and to at least maintain the efficiency in the remaining parts of the working range), the design with the highest efficiency at 76 % of Q_{OPT} was selected for further investigation. As far as the results from the simplified optimization CFD model are considered, it clearly dominates the old design. The efficiencies at Q_{OPT} and at 120 % of Q_{OPT} are excellent, too. For this design, the complete CFD model, including the sealing rings, was assembled. Next, the full set of CFD simulations for multiple flow rates was performed. The results can be seen in Figure 35. Unfortunately, the results did not meet expectations. The performance with the effect of the sealing rings is considerably worse than expected. While it is still higher than the performance of the original design, the efficiency drop in the sub-optimal flow rates is severe (when compared to the simplified CFD model). While the K-RVEA design dominated the solution found by Stochastic RBF, it does not hold true for the more complex model. The peak efficiency at Q_{OPT} is still higher, but overall, the Stochastic RBF design is preferable. The probable explanation is that that the "side-flow", caused by the leakage at the impeller inflow and outflow, can disturb the flow in the passage.

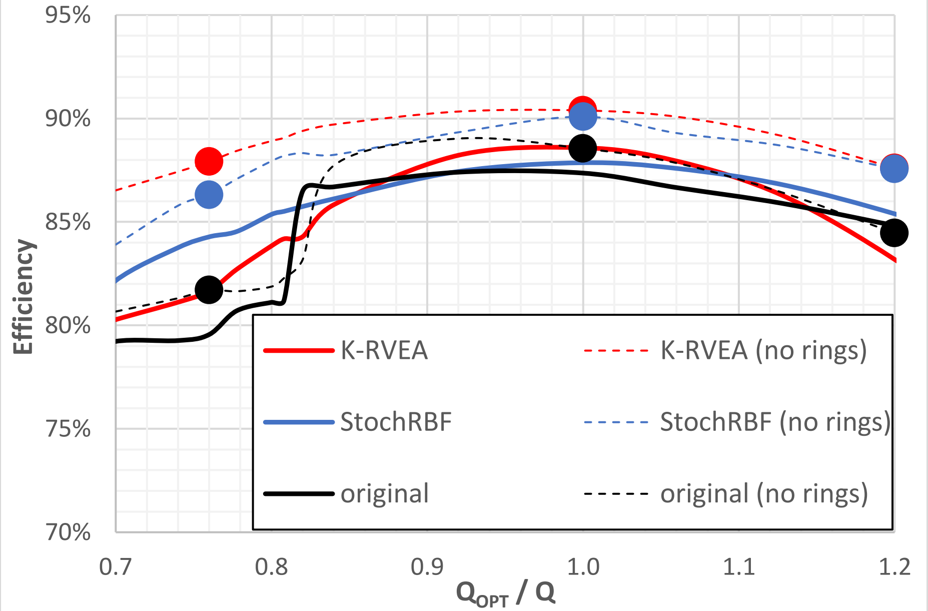

Figure 35: Performance comparison between original and optimized designs. The dots show optimization objectives.

7.5. Conclusion

The pump performance was successfully improved by shape optimization of the diffuser. Two methods were employed for the task – the previously used (single-objective) Stochastic RBF, and more modern and sophisticated K-RVEA, designed for computationally expensive multi objective optimization. Due to technical limitations, the sealing ring were not considered during the optimization. With this optimization CFD model, K-RVEA displayed superior performance, clearly outperforming Stochastic RBF. Both methods also found an improvement over the original design. However, the original assumption, that the leakage can be ignored for the optimization, has been proven to be incorrect. Instead, the superior (by the simplified CFD model) design, found by K-RVEA, did not perform well once the leakage was added to the model.

8. Case 6 - multi-objective optimization of a complete pump

Objectives: Improving the pump performance, meet prescribed head at Q_{OPT}.
Solution: Python codes + premade templates (ANSYS BladeGen, TurboGrid and CFX), transient simulation (full geometry). Six objectives (efficiency at $\mathrm{Q_{OPT}}$, efficiencies at lower flow rates, efficiency at higher flow rates, head and cavitational properties). Stochastic RBF optimization method + scalarization, K-RVEA.
Results: Efficiency increased from 2% up to 5% in the working range.

8.1. Introduction

In this case, an axial pump with diagonal diffuser (specific speed $n_S=680$) was considered. The 3D model of the pump can be seen in Figure 36. The goal was optimizing both rotor and stator, while considering hydraulic performance in multiple flow rates. Impeller optimization increases the optimization complexity considerably, as – unlike in the stator optimization – the cavitation properties cannot be ignored anymore. As

the multiphase head-drop curves simulations (necessary for NPSH3 curve determination) are too computationally expensive, an estimation of cavitation performance, based on “Blade Loading” (static pressure distribution along the impeller blades, further denoted as BL) was used. Another problem is ensuring the design head at the Q_{OPT}.
A complex parametric model, with a parametric description of both impeller and diffuser, was created. Next, multiple objectives formulations and optimization approaches were used and tested against a classically created hydraulic design (by human expert).

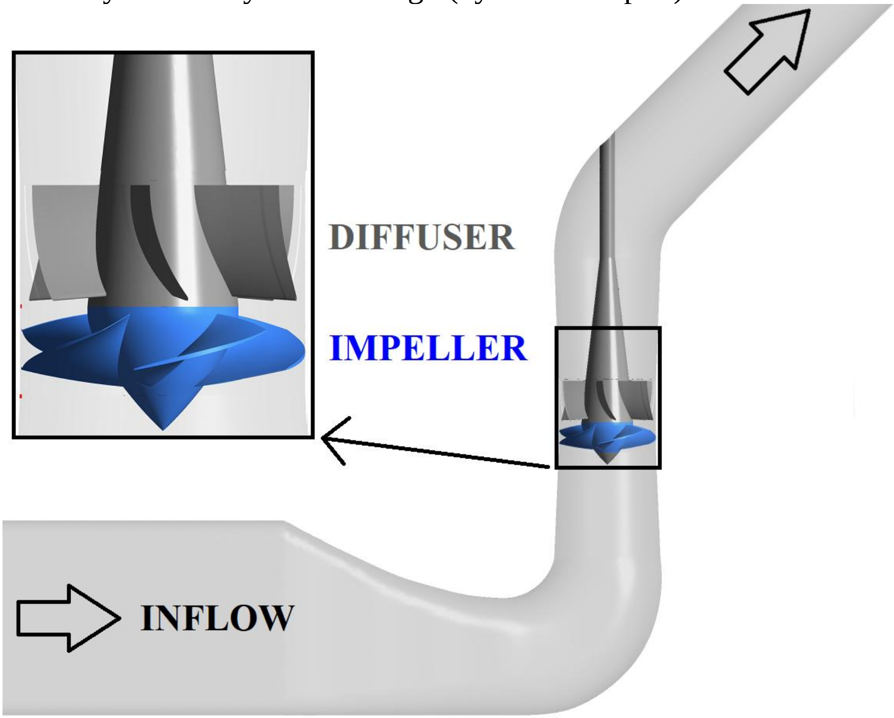

Figure 36: The optimized pump geometry, with a detail of the impeller and diffuser.

8.2. Parametric model and automation

The parametric model was completely new and different from the previous cases, based on premade ANSYS templates and various Python codes.

The model creation works as follows:

1. Generate the meridional section and beta angles distribution (for both rotor and stator).
2. Pass the data to a pre-made BladeGen *bgi* templates and create the inputs for TurboGrid.
3. Generate the meshes calling TurboGrid in batch mode.
4. Call CFX-Pre with a premade model and replace the rotor and stator meshes. Positioning the inflow and outflow meshes is also required in order to match the interfaces. Then create a solver input (.def) file.
5. Call CFD-Post and create graphical / text outputs – meridional section, mesh appearance and quality criterions.

The Python codes allow for a lot of variability. In total, there were 59 geometric parameters (decision variables). Some of these were passed directly to the BladeGen template, other serve as an input to subroutines. Overall, 71 parameters were used for generating the geometry. Due to various reasons, some of these were fixed, and 59 were "active" and used for the optimization.

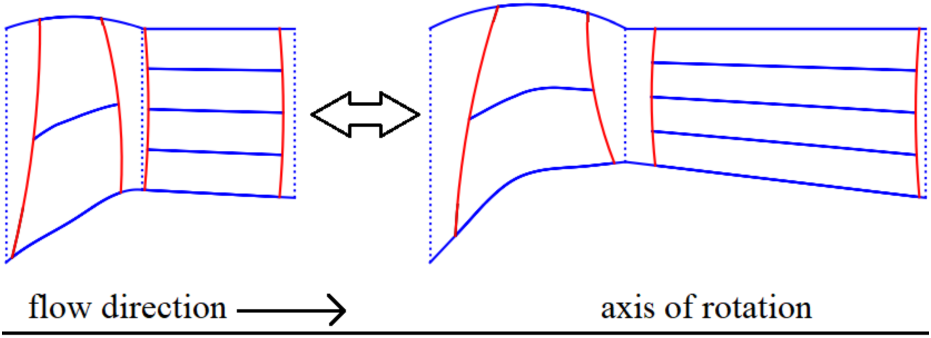

Figure 37: Visualisation of selected parameters

An example of a comprehensive visualization of a generated geometry, created automatically by the framework Figure 39. It displays slice of the CFD model, meridional sections, blade views and beta angles (for both rotor and stator), mesh information and normalized (in the objective space hypercube) parameter values – in comparison with a reference design. Such visualization serves for a quick inspection of the model.

8.3. Initial sampling and objectives

To perform the optimization, the original hydraulic design was used as a reference one. I.e., the ranges of the input parameters were set (after a discussion with the hydraulic expert) as reasonable offsets to the values used for the reference design. For the start, the initial sampling was generated. Its size was decided by the 2 * (N + 1) formula, same as in the previous cases. For 59 parameters this meant 120 samples. Out of these 120, 3 failed for geometric reasons, and another 2 had a very poor mesh quality. I.e. in total there were 115 samples left for the expensive CFD evaluation. Every sample was evaluated in five pre-set flow rates.

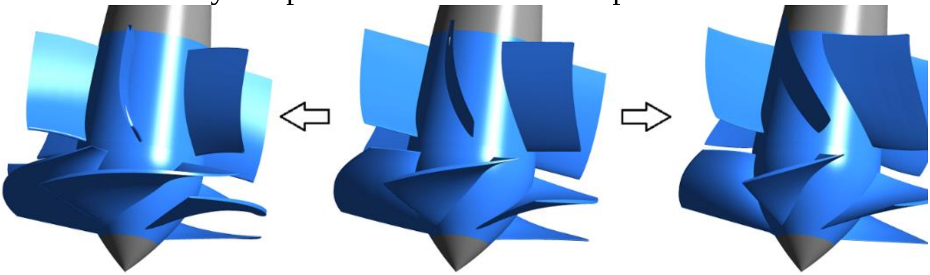

Figure 38: Impeller and diffuser blades for various settings of the parameters. The parametrized part of the geometry is displayed in blue.

The simulations were run on an HPC cluster. To capture the BL transient behaviour, it was necessary to analyse the data for every single timestep. As the intermediate *trn* file has ca. 1 GB, almost 2 TB od data had to be processed for each sample and flow rate. To avoid excessive data transfers, the results were processed on-the-fly, with only csv files with values being transferred.

An example of the BL data can be seen in Figure 41. The LE to TE direction is displayed from left to right, red colour is for the suction side and blue for pressure side. The pressure data were processed at 19 layers (*0.05*, *0.10*, … of Hub-to-Shroud distance) and averaged to a single curve, with weights *0.05*, *0.10*, … This reflects the fact that the character of the flow at the shroud is more important for the overall performance, as the velocity increases with the increasing distance to the axis of pump rotation.

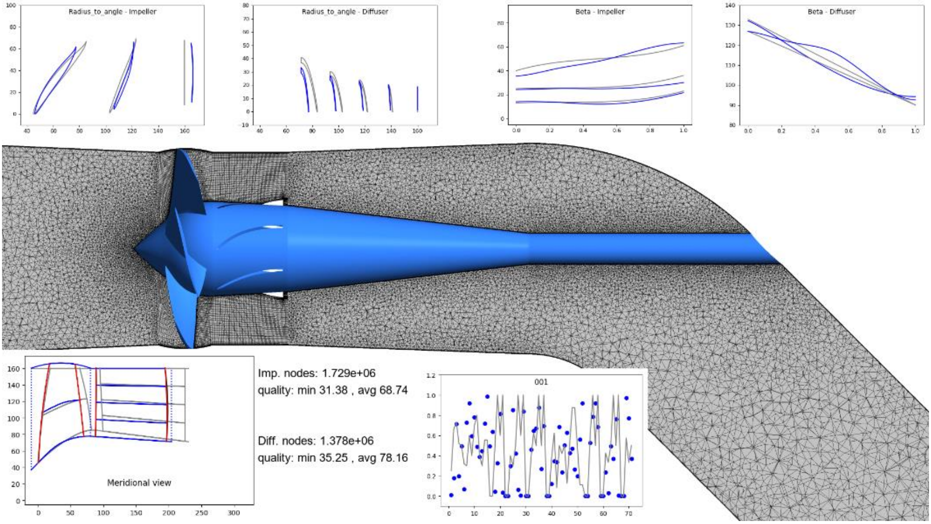

Figure 39: An example of graphical output of the parametric model

For practical reasons, the CFD runs were split into a sequence of shorter 2-hours runs, continued “until ending criterion” (“convergence” or maximum number of timesteps). At the end of each 2-hours run, the BL data (for both rotor and stator and every timestep) were extracted to csv, packed (to prevent unnecessary load on the file servers) and sent to the central storage. Once the CFD simulations for the initial sampling was finished, the results were analysed. There were multiple objectives to consider and evaluate:

1. The peak efficiency, respective the efficiency at the design point. (The actual optimum can occur at a different flow rate.)
2. The overall efficiency at the working range.
3. The NPSH$_3$ (cavitational) properties in the working range.
4. Meeting the desired head at the design point.

Evaluating the efficiencies and head is trivial. The cavitation properties, however, are more challenging, as the correlation between NPSH$_3$ and BL properties is not completely known. Based on our hydraulic experts’ opinions and (1), three main qualities of the BL shape were considered:

- The percentage of the curve where the pressure drops below the water vapour pressure (3169 Pa). In our case the pump was designed to operate at water level, i.e. at the atmospheric pressure. Thus, the threshold was set to approximately -1e5 Pa. The pressure drop occurs at the leading edge, and the designs with a good NPSH3 typically have this peek very narrow. The problem with this criterion is that once the cavitation develops, the BL changes. Thus, this metric only gives a hint about the true cavitational behaviour.

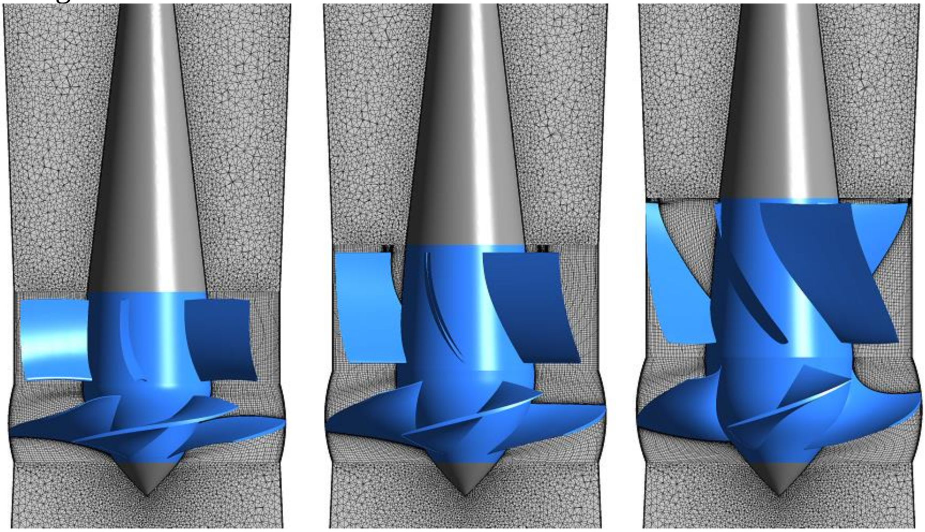

Figure 40: Examples of generated designs, including a slice through the computational mesh, for three different parameters settings. The parametrized parts of the pump are in blue.

- A "linearity" of the curve at the main passage, i.e. between the leading edge and trailing edge. As the total pressure difference between TE and LE is basically given (by the head), its distribution needs to be as even as possible. The numeric value was obtained as Pearson correlation coefficient between the (passage part of the) BL curve and the linear curve fit to it by Least squares method.
- The "transient" stability of the BL curves. This one is supposed to be related to the stability of other hydraulic characteristics (efficiency, head), which is desirable in general. It was evaluated as Pearson correlation coefficient between BL curves for each subsequent pair of timesteps. For all the samples, the efficiency, head and BL properties were evaluated.

8.4. Optimization – DYCORS

For the first optimization run and testing, the single-objective DYCORS (17) method was used. This method is an improvement over its predecessor, Stochastic RBF, and uses a more sophisticated approach for generating the *Candidate points*. According to the numerical testing, it is more efficient for higher dimensions, especially if the number of parameters exceeds *30*.

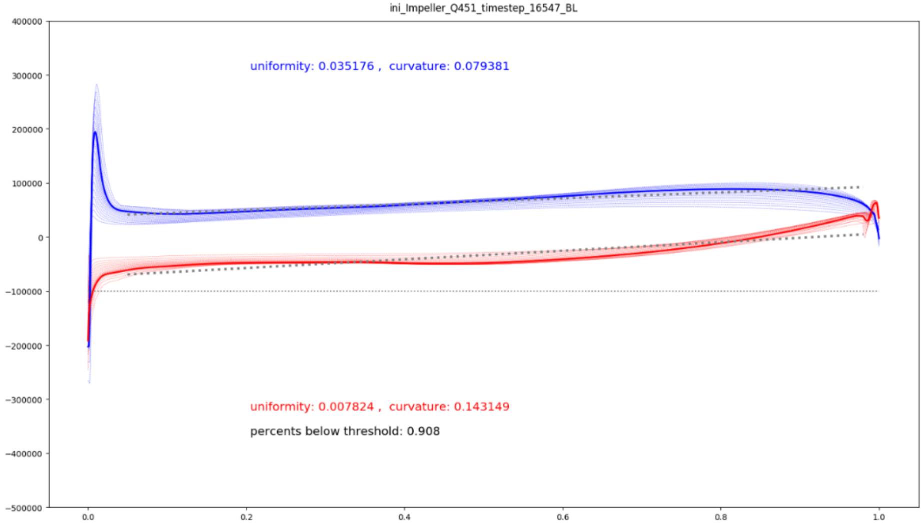

Figure 41: Blade loading plot example

The considered objectives were:

1. Efficiency at the design point – the goal was to ensure meeting efficiency objective at the design point Q_{OPT}.
2. Efficiency curve in the working range – multiple flow rates were considered to ensure a good performance overall.
3. BL curve – percentage below the threshold and suction shape – these metrics are often used by the hydraulic experts.
4. Difference from the head (at the design point) – this was to ensure that the optimized design meets the required value of head, instead of finding a geometry with great efficiency, but with an unacceptable head.

For the scalarization, the approach recommended in (13) was used. For each objective Ji, its value is normalized as

$$J_{i\_norm} = \frac{J_i - J_{i\_threshold}}{J_i - J_{i\_utopian}}$$

Where $J_{i\_threshold}$ is a selected threshold value, and $J_{i\_utopian}$ an ideal (but unreachable) value. The scalarized objective function is then assembled as

$$J_{total} = \max_{i=1,\dots,N} J_{i\_norm} + \sum_{i=1,\dots,N} \frac{J_{i\_norm}}{"large\ number"}$$

The “*large number”* can be selected arbitrarily, but such, that the sum is relatively small when compared to the *max* part of the formula. Thanks to this, the scalarized objective can capture changes to objectives that are lost in the *max*, and the desired properties of the scalarization are still preserved. This way it is possible to capture the Pareto front more efficiently. The choice of the threshold and utopian values influence the “weights” of the objectives. In our case, the results of the initial hydraulic design were used for the thresholds.

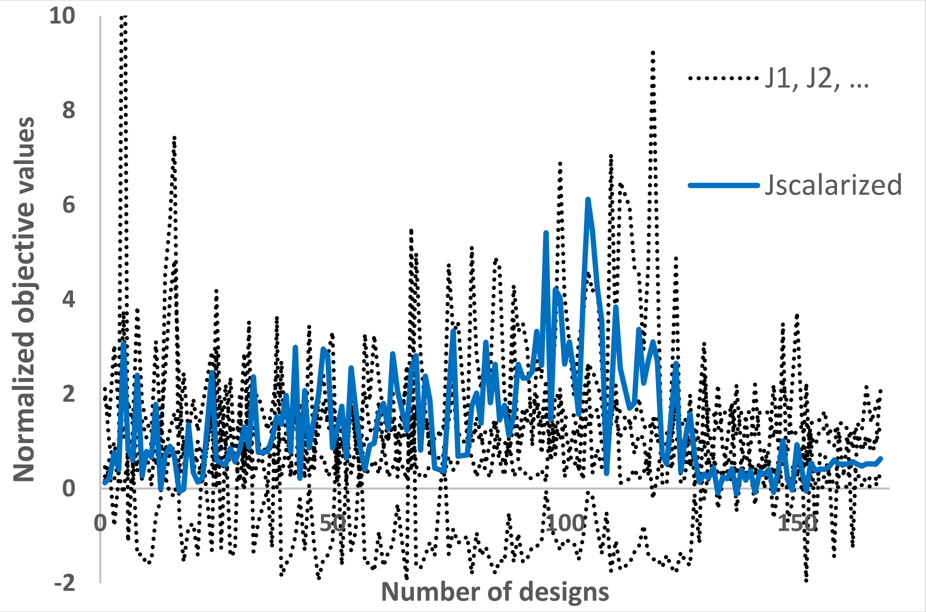

Figure 42: Objective functions and the scalarized objective function for DYCORS.

The DYCORS optimization was stopped after 12 iterations. The performance comparison between the “baseline” design and the result of the optimization can be seen in Figure 43 and Figure 44.

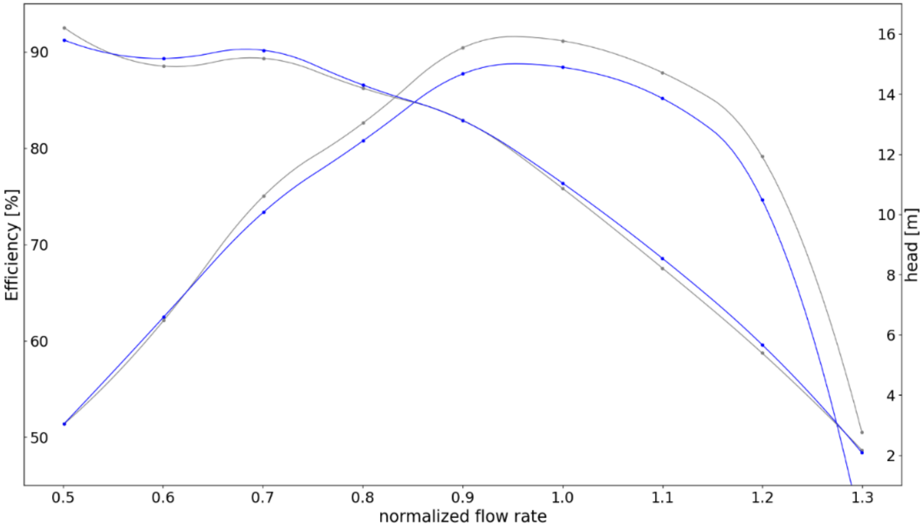

Figure 43: Optimization results. Comparison of hydraulic performance (efficiency and head) with the baseline design.

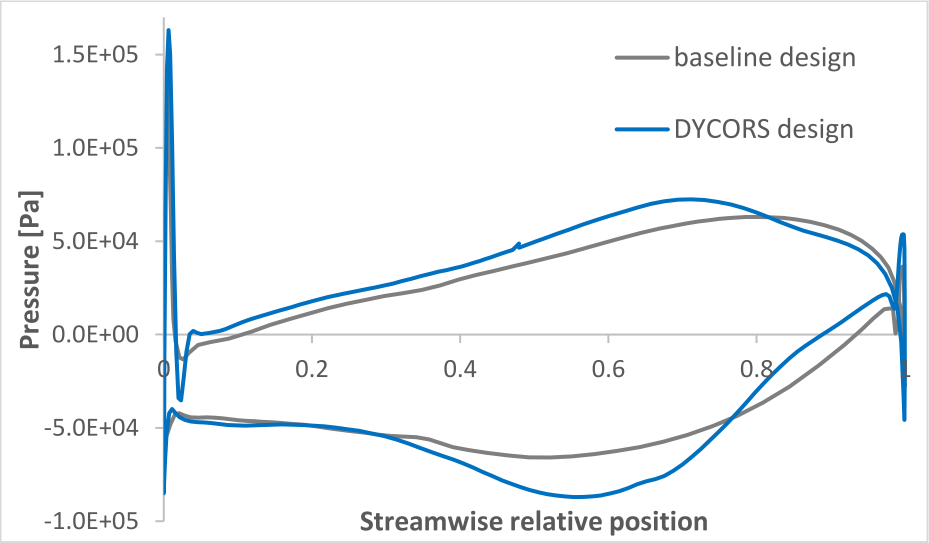

Figure 44: Optimization results. Comparison of BL at 110 % of Q_{OPT}.

8.5. Optimization – K-RVEA

For the second run, a multi-objective K-RVEA method, described in the previous chapter, was used. Six different objectives were considered:

1. Efficiency at the design point
2. Efficiency in the range of 60 to 100 % of the design point
3. Efficiency in the range of 100 to 120 % of the design point.
4. BL curve – percentage below the threshold
5. BL curve at the suction
6. Difference from the design head (at the design point)

Instead of scalarization, each objective was considered independently. Also, overall efficiencies for the sub-optimal, and "above-optimal" flow rates were considered separately. The intention was to have better control on optimizing the design for lower or higher flow rates.

To start the K-RVEA optimization, data from the same initial sampling as in the previous case was used. Overall, ca. 60 samples were created – the non-dominated solutions can be seen in Figure 46, hypervolume of the objective space is shown in Figure 46.

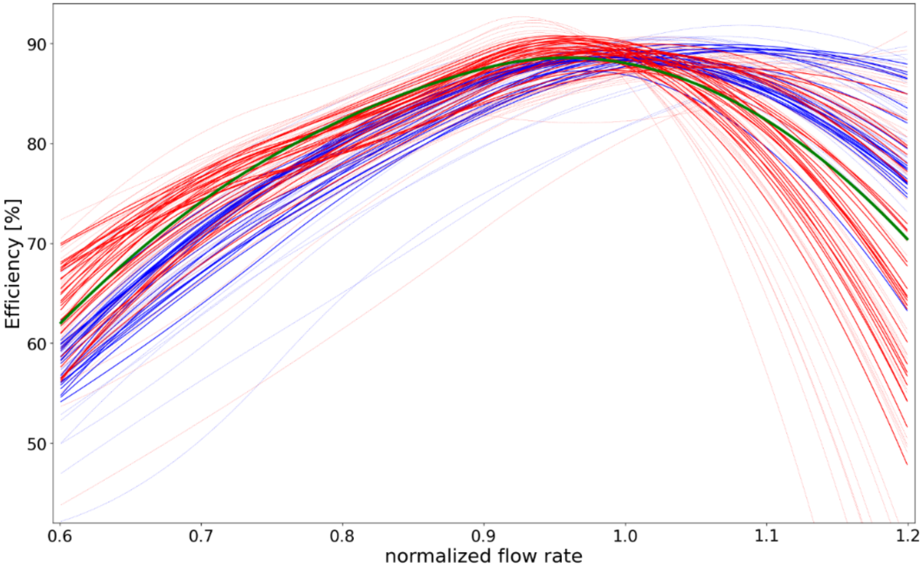

Figure 45 Efficiency curve approximated from five points (flow rates). The K-RVEA samples are in red, DYCORS in blue and the baseline design in green. The lines are bolder for designs that are within the tolerance for the design head.

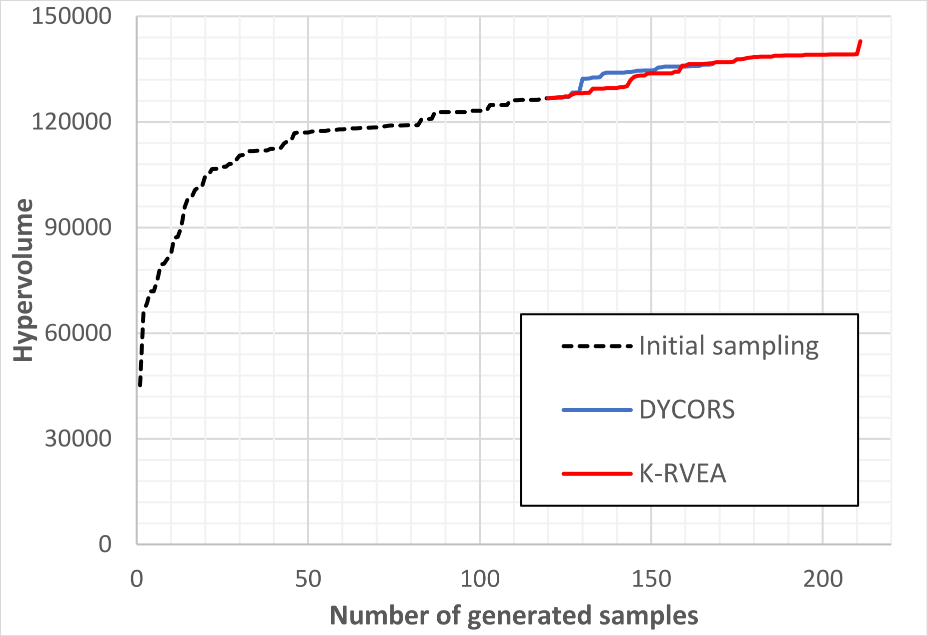

Figure 46 Hypervolume for both DYCORS and K-RVEA. For the comparison, the DYCORS samples were evaluated by the same objectives that were used for K-RVEA optimization.

8.6. Lessons learned and future plans

First, setting the difference from the design head as one of the objectives is a valid, but very inefficient way of ensuring this design objective. As this objective is ”binary”, i.e. either *passed* or *failed* (the designs with a wrong head are of no use – only to guide the optimization), a lot of computational resources is spent on undesired designs. Unfortunately, penalization is a problematic approach for surrogates-aided-optimization, as it distorts the surrogates.

Second, considering the BL qualities as objectives competing with efficiencies was a mistake, too.

Generally, only the efficiencies can be considered as the “true” objectives, as it makes sense to ask the “Pareto” questions such as “*How much efficiency am I willing to sacrifice at flow rate 1 to improve the efficiency at flow rate 2?*”. But asking like this in the *efficiency vs meeting the design head* scenario does not make sense.

9. Conclusion

When implemented properly, automated optimization of hydraulic shapes can be a valuable addition to the tools available for hydraulic design of pumps. While theoretically such approach could replace the classical approach altogether, in practice there are several limitations. First, the computational cost of the necessary numerical simulations is very prohibitive and severely limits possibilities of exploring the parametric space. When considering the relatively high number of parameters, necessary for describing the hydraulic shapes, and lack of derivative information, Surrogate-aided optimization is the only viable solution. Creation of automated parametric models, and suitable formulation of the optimization objective(s) have also proven to be very challenging.

In this thesis, multiple real-world optimization cases were presented – ordered from lower to higher difficulty / complexity. The first presented case is not an actual optimization, but only uses iterative sampling in the parametric space. Next, the automated parametric model was created and connected with a suitable surrogate-based, single objective method. In the following cases, multiobjective optimization of pump stator was presented. Finally, in the last case, the optimization of the rotor and stator parts with respect to multiple objectives was presented.

Parametric models and their automation proved to be technically difficult. When optimizing stator parts, the optimization results were promising. The newly found designs exceeded the performance of the original models. Considering the rotor hydraulic parts posed bigger challenge, though. First, the hydraulic shapes are more complex and require even more geometric parameters, making the optimization more complex. Second, more objectives need to be considered in impeller design, such as cavitation properties.

The results of the impeller optimization indicate that further development is needed. The goal is to have a method that combines the “fast-available” information (geometry features, analytical formulas) and “slow” (computationally expensive) objectives and can consider “classification” objectives – the same way as human designers do.

10. References

1. **Gülich, Johann Friedrich.** *Centrifugal Pumps.* s.l. : Springer-Verlag Berlin Heidelberg, 2014. 978-3-642-40114-5.

2. **Storn, R., Price, K.** Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization.* 1997, Vol. 11, pp. 341–359.

3. **Kennedy, J. and Eberhart, R.** Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks.* 1995, pp. 1942–1948.

4. **Preuss, Mike.** *Multimodal Optimization by Means of Evolutionary Algorithms.* s.l. : Springer International Publishing AG, 2015. 9783319074061 .

5. **Chugh, T., Sindhya, K., Hakanen, J. et al.** A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing.* 2019.

6. **Witteck, D, Micallef, D, & Mailach, R.** Comparison of Transient Blade Row Methods for the CFD Analysis of a High-Pressure Turbine. *Proceedings of the ASME Turbo Expo 2014: Turbine Technical Conference and Exposition.* June 16–20, 2014, Vol. 20: Turbomachinery.

7. *Study of Steady State and Transient Blade Row CFD Methods in a Moderately Loaded NASA Transonic High-Speed Axial Compressor Stage.* **Qizar, Mohammed & Mansour, Mahmoud & Goswami, Shraman.** 2013. ASME Turbo Expo 2013: Turbine Technical Conference and Exposition.

8. **Giannakoglou K.C., Papadimitriou D.I.** Adjoint Methods for Shape Optimization. [book auth.] Janiga G. (eds) Thévenin D. *Optimization and Computational Fluid Dynamics.* s.l. : Springer, Berlin, Heidelberg, 2008.

9. **Liu, H., Ong, Y. & Cai, J.** A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization.* 2018, Vol. 57, pp. 393–416.

10. **Jin, Yaochu.** Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation.* 2011, Vol. 1, 2, pp. 61-70.
11. **Regis, Rommel & Shoemaker, Christine.** A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. *INFORMS Journal on Computing.* 2007, Vol. 19, pp. 497-509.
12. —. Parallel Stochastic Global Optimization Using Radial Basis Functions. *INFORMS Journal on Computing.* 2009, Vol. 21, pp. 411-426.
13. **Miettinen, K.** Introduction to Multiobjective Optimization : Noninteractive Approaches. *Multiobjective Optimization : Interactive And Evolutionary Approaches.* 2008, pp. 1–26.
14. **Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., & Sindhya, K.** A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-objective Optimization. *IEEE Transactions on Evolutionary Computation.* 2018, Vol. 22, 1, pp. 129-142.
15. **R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff.** A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation.* 2016, Vol. 20, 5, pp. 773-791.
16. *Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system.* **T. Chugh, K. Sindhya, K. Miettinen, Yaochu Jin, T. Kratky and P. Makkonen.** San Sebastian : s.n., 2017. IEEE Congress on Evolutionary Computation (CEC). pp. 1541-1548.
17. **Shoemaker, Rommel G. Regis & Christine A.** Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization.* 2013, Vol. 45, 5, pp. 529-555.
18. **S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera,.** *S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera,.*

19. **S.R. Shah, S.V. Jain, R.N. Patel, V.J. Lakhera.** CFD for Centrifugal Pumps: A Review of the State-of-the-Art. *Procedia Engineering.* 2013, Vol. 51, pp. 715-720.
20. **R. Spence, J. Amaral-Teixeira.** A CFD parametric study of geometrical variations on the pressure pulsations and performance characteristics of a centrifugal pump. *Computers & Fluids.* 2009, Vol. 38, 6, pp. 1243-1257.
21. *Shape Optimization of Turbomachinery Blade Using Multiple Surrogate Models.* **Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei.** 2006. ASME 2006 2nd Joint U.S.-European Fluids Engineering Summer Meeting Collocated With the 14th International Conference on Nuclear Engineering.
22. **Jingwei Jiang, Haopeng Cai, Cheng Ma, Zhengfang Qian, Ke Chen & Peng Wu.** A ship propeller design methodology of multi-objective optimization considering fluid–structure interaction. *Engineering Applications of Computational Fluid Mechanics.* 2018, Vol. 12, 1.
23. **Derakhshan, Shahram & Mohammadi, Bijan & Nourbakhsh, Ahmad.** The comparison of incomplete sensitivities and Genetic algorithms applications in 3D radial turbomachinery blade optimization. *Computers & Fluids.* 2010, Vol. 39, pp. 2022-2029.
24. **ones, D.R.** A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization.* 2001, Vol. 21, pp. 345–383.
25. **Shahram Derakhshan, Maryam Pourmahdavi, Ehsan Abdolahnejad, Amin Reihani, Ashkan Ojaghi.** Numerical shape optimization of a centrifugal pump impeller using artificial bee colony algorithm. *Computers & Fluids.* 2013, Vol. 81, pp. 145-151.
26. **Xiao Tang, Jiaqi Luo, Feng Liu.** Aerodynamic shape optimization of a transonic fan by an adjoint-response surface method. *Aerospace Science and Technology.* 2017, Vol. 68, pp. 26-36.
27. **C.A. Gilkeson, V.V. Toropov, H.M. Thompson, M.C.T. Wilson, N.A. Foxley, P.H. Gaskell.** Dealing with numerical noise in CFD-based design optimization. *Computers & Fluids.* 2014, Vol. 94, pp. 84-97.

28. **V.O. Lomakin, P.S. Chaburko, M.S. Kuleshova.** Multi-criteria Optimization of the Flow of a Centrifugal Pump on Energy and Vibroacoustic Characteristics. *Procedia Engineering.* Vol. 176.
29. **Huang, R., Luo, X., Ji, B. et al.** Multi-objective optimization of a mixed-flow pump impeller using modified NSGA-II algorithm. *Sci. China Technol. Sci.* 2015, Vol. 58, pp. 2122–2130.
30. **Yongsheng Lian, Akira Oyama, Meng-Sing Liou.** Progress in design optimization using evolutionary algorithms for aerodynamic problems. *Progress in Aerospace Sciences.* 2010, Vol. 46, 5-6, pp. Pages 199-223.
31. **K.-H. Wu, B.-J. Lin & C.-I. Hung.** Novel Design of Centrifugal Pump Impellers Using Generated Machining Method and CFD. *ngineering Applications of Computational Fluid Mechanics.* 2008, Vol. 2, 2, pp. 195-207.
32. **Siddique, M Hamid & Afzal, Arshad & Samad, Abdus.** Design Optimization of the Centrifugal Pumps via Low Fidelity Models. *Mathematical Problems in Engineering.* 2018, pp. 1-14.
33. **Ivo Marinić-Kragić, Damir Vučina & Zoran Milas.** 3D shape optimization of fan vanes for multiple operating regimes subject to efficiency and noise-related excellence criteria and constraints. *Engineering Applications of Computational Fluid Mechanics.* 2016, Vol. 10, 1, pp. 209-227.
34. **Xiao Tang, Jiaqi Luo, Feng Liu.** Aerodynamic shape optimization of a transonic fan by an adjoint-response surface method. *Aerospace Science and Technology.* 2017, Vol. 68, pp. Pages 26-36.
35. **Kang, I. S. Jung and W. H. Jung and S. H. Baek and S.** Shape Optimization of Impeller Blades for a Bidirectional Axial Flow Pump using Polynomial Surrogate Model. *Transactions of the Korean Society of Mechanical Engineers B.* 2012, Vol. 6, 6.
36. **H. Safikhani, A. Khalkhali & M. Farajpoor.** Pareto Based Multi-Objective Optimization of Centrifugal Pumps Using CFD, Neural Networks and Genetic Algorithms. *Engineering Applications of Computational Fluid Mechanics.* 2011, Vol. 5, 1, pp. 37-48.
37. **V.O. Lomakin, P.S. Chaburko, M.S. Kuleshova.** Multi-criteria Optimization of the Flow of a Centrifugal Pump on Energy and

Vibroacoustic Characteristics. *Procedia Engineering.* 2017, Vol. 176, pp. 476-482.
38. *Optimal design of multi-conditions for axial flow pump.* **Shi, L & Tang, F & Liu, C & Xie, R & Zhang, W.** 2016. IOP Conference Series: Earth and Environmental Science. Vol. 49.
39. **Huang, R., Luo, X., Ji, B. et al.** Multi-objective optimization of a mixed-flow pump impeller using modified NSGA-II algorithm. *Sci. China Technol. Sci.* 2015, Vol. 58, pp. 2122–2130.
40. **Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei.** Multiple Surrogate Modeling for Axial Compressor Blade Shape Optimization. *Journal of Propulsion and Power.* 2008, Vol. 24, pp. 302-310.
41. **Han, X., et al.** Impeller Optimized Design of the Centrifugal Pump: A Numerical and Experimental Investigation. *Energies.* 2018, Vol. 11, p. 1444.
42. **Xing'an Zhao, Biao Huang, Tairan Chen, Guoyu Wang, Deming Gao, Jing Zhao.** Numerical simulations and surrogate-based optimization of cavitation performance for an aviation fuel pump. *Journal of Mechanical Science and Technology.* 2017, Vol. 31, 2, pp. 705-716.
43. **Rajkumar Roy, Srichand Hinduja, Roberto Teti.** Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals.* 2008, Vol. 57, 2, pp. 697-715.
44. *Shape Optimization of Turbomachinery Blade Using Multiple Surrogate Models.* **Samad, Abdus & Kim, Kwang-Yong & Goel, Tushar & Haftka, Raphael & Shyy, Wei.** 2006. ASME 2006 2nd Joint U.S.-European Fluids Engineering Summer Meeting Collocated With the 14th International Conference on Nuclear Engineering.
45. **Siddique, M Hamid & Afzal, Arshad & Samad, Abdus.** Design Optimization of the Centrifugal Pumps via Low Fidelity Models. *Mathematical Problems in Engineering.* 2018, pp. 1-14.
46. **Baoshan Zhu, Xuhe Wang, Lei Tan, Dongyue Zhou, Yue Zhao, Shuliang Cao.** Optimization design of a reversible pump–turbine runner with high efficiency and stability. *Renewable Energy.* 2015, Vol. 81, pp. 366-376.

47. **Milas, Zoran & Vučina, Damir & Marinic-Kragic, Ivo.** Multi-Regime Shape Optimization of Fan Vanes for Energy Conversion Efficiency using CFD, 3D Optical Scanning and Parameterization. *Engineering Applications of Computational Fluid Mechanics.* 2014, Vol. 8, pp. 407-421.
48. **X.D. Wang, C. Hirsch, Sh. Kang, C. Lacor.** Multi-objective optimization of turbomachinery using improved NSGA-II and approximation model. *Computer Methods in Applied Mechanics and Engineering.* 2011, Vol. 200, 9-12, pp. 883-895.
49. **Škerlavaj, Aljaž & Morgut, M & Jošt, Dragica & Nobile, Enrico.** Optimization of a single-stage double-suction centrifugal pump. *Journal of Physics: Conference Series.* 2017, Vol. 796, 1.
50. **Iuliano, Emiliano.** Global optimization of benchmark aerodynamic cases using physics-based surrogate models. *Aerospace Science and Technology.* 2017, Vol. 67, pp. 273-286.
51. **KIM, Jin-Hyuk & Choi, Jae-Ho & Husain, Afzal & Kim, Kwang-Yong.** Multi-Objective Optimization of a Centrifugal Compressor Impeller through Evolutionary Algorithms. *Proceedings of the Institution of Mechanical Engineers Part A Journal of Power and Energy.* 2010, Vol. 224, pp. 711-721.
52. **Erik Tengs, Pål-Tore Storli & Martin Holst.** Optimization procedure for variable speed turbine design. *Engineering Applications of Computational Fluid Mechanics.* 2018, Vol. 12, 1, pp. 652-661.
53. **Wang, Yuqin & Huo, Xinwang.** Multiobjective Optimization Design and Performance Prediction of Centrifugal Pump Based on Orthogonal Test. *Advances in Materials Science and Engineering.* 2018, Vol. 15, pp. 1 - 10.
54. **Škerlavaj, Aljaž & Morgut, M & Jošt, Dragica & Nobile, Enrico.** Decoupled CFD-based optimization of efficiency and cavitation performance of a double-suction pump. *Journal of Physics: Conference Series.* 2017, Vol. 813, 1.
55. **Shaaban, S.** Design optimization of a centrifugal compressor vaneless diffuser. *International Journal of Refrigeration.* 2015, Vol. 60, pp. 142-154.

56. **Feliot P., Bect J., Vazquez E.** User Preferences in Bayesian Multi-objective Optimization: The Expected Weighted Hypervolume Improvement Criterion. *Lecture Notes in Computer Science, vol 11331.* 2019.
57. **S.N. Skinner, H. Zare-Behtash.** State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing.* 2018, Vol. 62, pp. 933-962.
58. **Safikhani, Hamed & Khalkhali, Abolfazl & Farajpoor, M.** Pareto Based Multi-Objective Optimization of Centrifugal Pumps Using CFD, Neural Networks and Genetic Algorithms. *Engineering Applications of Computational Fluid Mechanics.* 2011, Vol. 5, pp. 37-48.
59. **Jinya Zhang, Hongwu Zhu, Chun Yang, Yan Li, Huan Wei.** Multi-objective shape optimization of helico-axial multiphase pump impeller based on NSGA-II and ANN. *Energy Conversion and Management.* 2011, Vol. 52, 1, pp. 538-546.
60. *A multi-response adaptive sampling approach for global metamodeling.* **Liu, Haitao & Xu, Shengli & Wang, Xiaofang & Yang, Shuhua & Meng, Jigang.** 2016. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.
61. **Zhihui Li, Xinqian Zheng.** Review of design optimization methods for turbomachinery aerodynamics. *Progress in Aerospace Sciences.* 2017, Vol. 93, pp. 1-23.
62. **Yin Yu, Zhoujie Lyu, Zelu Xu, Joaquim R.R.A. Martins.** On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Aerospace Science and Technology.* 2018, Vol. 75, pp. 183-199.
63. **J. Tian, Y. Tan, J. Zeng, C. Sun and Y. Jin.** Multiobjective Infill Criterion Driven Gaussian Process-Assisted Particle Swarm Optimization of High-Dimensional Expensive Problems. *IEEE Transactions on Evolutionary Computation.* 2019, Vol. 23, 3, pp. 459-472.
64. **Zhang, Qi-hua & Cao, Li & Yan, Zhaoxu & Zhang, Weidong.** Hydraulics and Blading of Centrifugal Pump Impellers: A Systematic

Review and Application. *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering.* 2018.
65. **Claudia Schillings, Stephan Schmidt, Volker Schulz.** Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids.* 2011, Vol. 46, 1, pp. 78-87.
66. **L. Bonfiglio, P. Perdikaris, S. Brizzolara, G.E. Karniadakis.** Multi-fidelity optimization of super-cavitating hydrofoils. *Computer Methods in Applied Mechanics and Engineering.* 2018, Vol. 332, pp. 63-85.
67. **Wei-xue Cao, Xue-yi You.** The inverse optimization of exhaust hood by using intelligent algorithms and CFD simulation. *Powder Technology.* 2017, Vol. 315, pp. 282-289.
68. **Antonio Posa, Antonio Lippolis.** A LES investigation of off-design performance of a centrifugal pump with variable-geometry diffuser. *International Journal of Heat and Fluid Flow.* 2018, Vol. 70, pp. 299-314.
69. **Md. Abdul Raheem Junaidi, N.B.V Laksmi Kumari, Mohd Abdul Samad, G.M. Sayeed Ahmed.** CFD Simulation to Enhance the Efficiency of Centrifugal Pump by Application of Inner Guide Vanes. *Materials Today: Proceedings.* 2015, Vol. 2, 4-5, pp. 2073-2082.