ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

SENSITIVITY STUDY OF THE NUMERICAL SETUP FOR AN AUTOMATIC OPTIMIZATION PROCEDURE FOR A HYDRAULIC MACHINE

Alexander Tismer[1], **Markus Schlipf**[1] **and Stefan Riedelbauch**[1]

[1]Institute of Fluid Mechanics and Hydraulic Machinery
Pfaffenwaldring 10, 70550 Stuttgart, Germany
e-mail: alexander.tismer, markus.schlipf, stefan.riedelbauch@ihs.uni-stuttgart.de

Keywords: CFD, turbo-machine, structured-unstructured mesh, OpenFOAM, machine design

Abstract. *For an automatic optimization using Computational Fluid Dynamics, all design iterations of a turbo-machine require the generation of geometry and computational mesh as well as the simulation of the flow and the evaluation of one or more characteristic properties. All steps can be expensive regarding time or manpower. In addition, all meshes of the design iterations differ in total number of cells, percentage of a particular element type, local refinement sections or structured mesh block thickness being summarized as mesh characteristics. An object-oriented framework focused on hydraulic machinery is used to generate different geometry variants. Furthermore, the framework is extended to create hybrid meshes consisting of a structured part surrounding the blades and an unstructured part within the remaining flow channel. The structured-unstructured mesh coupling is realized with the Pyramid Open Method. In order to take care of the boundary layer flow effects along the meridional contour upstream and downstream of the runner blade rows a prismatic layer creation algorithm is also implemented. OpenFOAM software is applied to simulate the flow through the hydraulic machine. Each step in the process chain is implemented as part of the framework. The complete study is done using open source applications, libraries and tools. In terms of performing an automatic optimization procedure an assessment for the numerical behavior of the hybrid mesh in a real-engineering application is desired. To limit the number of optimization parameters and the amount of computation time the sensitivity analysis tool supports the identification of relevant geometry input parameters with significant influence on the flow field. On the one hand it is important to discover the influence of geometry variations on the predicted physical behavior (e.g. flow losses, torque and head) of the simulation results. On the other hand it is investigated if the uncontrollable changes of mesh characteristics or user controlled numerical discretization parameters, such as flux gradients affect the physical quantities on the same level as the geometric variations. The goal is to achieve a dominating geometry influence and a minor effect of mesh characteristics on the predicted flow field solution. For both, a typical Francis and a Propeller turbine runner, the fluctuations and absolute values of physical properties are shown for a number of varying meshes, numerical and geometrical setup parameters.*

1 INTRODUCTION

In almost all cases hydraulic parts of power plants are constructed individually for each power plant and hence for special operating conditions. From an engineer's point of view this means finding a geometry that fits best to the desired operating conditions. Each design iteration requires generating a geometry, meshing the domain, simulating the flow field and investigating the solution. Iterations or variants of a hydraulic machine have to be evaluated against each other to distinguish if a variant is better than the others. In addition a design process is normally limited by a project duration and in that sense it is worthwhile to automate all steps as much as possible.

The very first step of the process is to parameterize a geometry to be able to create different variations. For hydraulic machines, e.g. Francis or Propeller runners, the blade angles at leading and trailing edge, blade thickness, meridional and circumferential blade length serve as possible design degrees of freedom (DDoF). A geometry of a machine can be seen as a function that takes the DDoF as inputs and generates a geometry. Frequently, geometry and mesh generation applications act on different file formats which produces the necessity to convert output of a first application to the correct input for a second application. Meshing a hydraulic machine with a conservative block-structured grid is a highly intuitive and manpower intensive step in the optimization loop. A promising approach to reduce manual effort is the use of unstructured or partly unstructured meshes. A hybrid mesh fits more easily a wider range of geometry variants due to its variable unstructured part. On the other hand it is known that unstructured meshes also produce more inaccuracy regarding numerical errors [1]. Hybrid meshes with hexahedral block-structured elements around the blade and prisms in the boundary layer at hub and shroud are used to reduce numerical errors. The flow field in the hydraulic machine is obtained from a numerical solution of the incompressible Navier-Stokes equations on the mesh, e.g. using a finite volume approach. In the last step of a design iteration an obtained flow field has to be evaluated and classified. Characteristic properties of a design (CPoD) with either scalar or vector character have to be defined. Common physical CPoD are e.g. torque, efficiency and velocity distribution downstream the runner. The question, if a design of the hydraulic machine fits better to the operating conditions than others, is answered by evaluating those parameters.

2 FRAMEWORK

Hydraulic design environments based on applications that have a set of fixed DDoF are meaningful approaches for new designs of hydraulic components [2, 3]. Refurbishment projects need tools that are customizable and extensible. The authors believe that a framework can provide both and hence is able to be applied to both scenarios. Adaptability from the user's perspective is achieved by the use of a modifiable, human-readable construction plan of the machine. For a developer an object-oriented code structure can reduce the work for an implementation of a new feature. The already developed framework is able to handle fairly different machine types and is extended for this study [4]. A lot of open-source libraries reduce the amount of new code. The framework can be divided in different sections: geometry, mesh, pre-processing, flow simulation and post-processing. All steps of the workflow are part of the framework and thus, can be extended by modifying the source code. In order to extend the possibilities regarding geometrical modifications the underlying geometry engine of the framework is adjusted to use the well-established open source project OPENCASCADE [5]. Further used libraries remain the same (see [4] for a list).

2.1 STRUCTURE

Figure 1 shows the general structure with its main parts. The left column lists parts that a "common" user or programmer interacts with. These are implementations for geometrical objects e.g. one-, two- and three-dimensional mappings or mesh objects. The framework or rather OPENCASCADE uses non-uniform rational B-Splines (NURBS) to handle free-form geometrical shapes. It is also possible to handle compositions of NURBS or other geometrical objects [4]. The *solverKernel* provides the interface to a simulation tool that computes a numerical solution of the flow field. In order to provide a mechanism for developers to extend the framework without recompiling or modifying the original source code, a simple plugin system is also provided [6]. For this study the post-processing steps are implemented as plugins. Parts in the right column of figure 1 are generalized objects, e.g. handling string objects, exception handling or logging functions. The user interacts with the framework via a Hypertext Markup Language file that serve as a construction plan for the machine. Additionally, a very first and simple stage of parallel support using MPI (Message Passing Interface) and OpenMP (Open Multi-Processing) is implemented. The connection to the used underlying open source libraries is bundled in the *libraryKernel*. The right column of figure 1 contains an incomplete list of used libraries. The last part *applicationKernel* provides some basic methods to write new applications that use the library and as a consequence become part of the framework. For this study an application that executes geometry generation, mesh generation, simulation and post-processing for one design sample is provided. The black box that surrounds all steps in the figure visually represents the mentioned application.

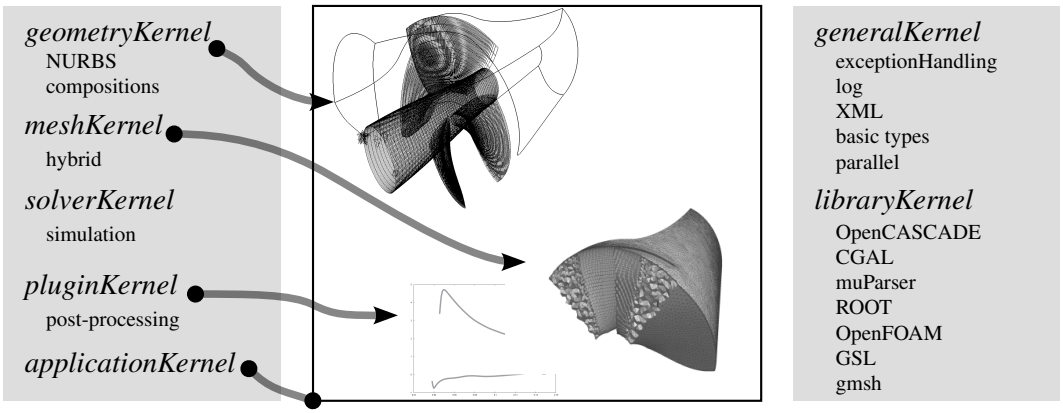


Figure 1: General framework structure

The mesh generation part uses the open source software GMSH [7]. Already implemented meshing steps of GMSH such as triangulation, tetrahedralization and creation of structured mesh blocks using a transfinite interpolation are used without modifications. The framework creates the mesh that depends on different d-dimensional geometrical objects G^d with the parameter space coordinates u, v and w in the same sequence as GMSH does:

(1) Vertices G^0 that are entities of dimension 0,
(2) Edges $G^1(u)$ that are entities of dimension 1,
(3) Faces $G^2(u, v)$ that are entities of dimension 2 and
(4) Regions $G^3(u, v, w)$ that are entities of dimension 3.

GMSH provides a bidirectional data structure of all G^d entities and, thus, enables to have all

upward $(d+1)$-dimensional and downward $(d-1)$-dimensional adjacencies of a geometrical object. Table 1 gives a list and short description of new implemented operators that are all involved in creating the hybrid meshes. The mesh is created in the same order for steps (1) - (3). Step (4) is subdivided into:

(a) create boundary layer elements on wall faces and take transfinite layers into account,
(b) add pyramid elements on quadrangles that are not part of boundary layers and
(c) fill unstructured part of regions.

Operator name	Description
dtMeshFreeGradingEdge	edge with user defined vertex distribution
dtMeshGFaceWithTransfiniteLayer	four sided face with a layer at two opposing edges
dtMeshGRegionWithBoundaryLayer	region with faces that can extrude (create a boundary layer), where vertices can slide and where vertices remain fixed (see section 2.3)

Table 1: Additionally implemented mesh operators

2.2 EDGE- AND SURFACE MESH

dtMeshFreeGradingEdge meshes an edge of length L and takes a predefined function

$$g : \mathbb{R} \to \mathbb{R} : g(x) = y \text{ with } g(0) = 0 \text{ and } g(1) = L \tag{1}$$

into account. The function space x is equidistantly divided into an user defined number of points. The i-th geometrical length l_i of the i-th mesh point is then given by

$$g(x_i) = l_i \ .$$

For this study a two dimensional B-Spline function

$$g_B : \mathbb{R} \to \mathbb{R}^2 : g_B(u) = (x_u, y_u)$$

is defined with three control points. In the sense of equation (1) g_B is forced to be a function by finding a root for the first dimension of

$$g_B(u) = (x_u - x_g, y_u) \tag{2}$$

with a given x_g. It is up to the user to make sure that the solution of equation (2) is unique. Controlling the grading by control points of B-Splines is assumed to be more intuitively from the user's perspective. But grading is not fixed to B-Spline functions only and hence hyperbolic tangent, geometric series, polynomial functions or others can be used.

The mesh operator *dtMeshGFaceWithTransfiniteLayer* acts on two dimensional mesh entities $G^2(u, v)$ according to figure 2. As first step the structured dark gray layer is created using a transfinite interpolation scheme. The second step creates the light gray unstructured part. Layers are not limited to have the same number of elements in u- and v-direction. The implementation of the operator is realized by a combination of existing operators of GMSH.

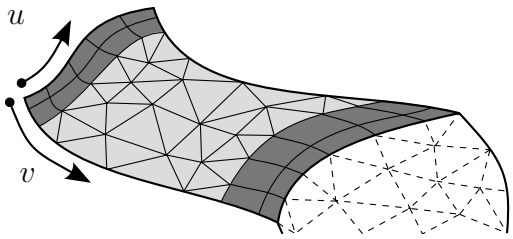


Figure 2: Face with a transfinite layer

2.3 PRISMATIC LAYER

Fluid dynamic boundary layer effects give rise to have special elements near walls. For this study a prismatic boundary layer is created on hub and shroud of the hydraulic machine. Figure 3 shows the dark gray layer as an extrusion of the unstructured light gray surface mesh. A simplified implementation of a generalized advancing layer technique is used to create the prismatic elements [8]. In the following a vertex positioned on a G^2 entity is called ordinary vertex and a vertex located at a G^1 or G^0 entity is called either sliding or fixed vertex. Firstly, a mesh manifold for each mesh vertex is created by finding a face set that shares one common mesh vertex. In figure 3 the manifold of the mesh vertex marked with a black rectangle is defined by all I light gray surface elements s that are touched by the dashed circle around the rectangle. Each surface element s_i has a unit normal vector $\underline{N}_S(i)$. The mesh unit normal $\underline{N}_V$ at mesh vertex j

$$\underline{N}_V(j) = \frac{\sum_i^I c_i \underline{N}_S(i)}{|\sum_i^I c_i \underline{N}_S(i)|} \tag{3}$$

has to be calculated for each ordinary vertex. For all ordinary points c_i is set to one. For each ordinary mesh vertex $\underline{N}_V(j)$ returns its unit normal vector and thus the direction of the extruded boundary layer. Special handling is necessary for sliding and fixed vertices. A sliding vertex is able to move along an internal mesh edge of a surface. A boundary layer that starts from the dashed surface mesh, figure 2, contains sliding vertices at mesh edge $G^1(u) = G^2(u, v = 1)$. All mesh vertices on the geometrical edge slide in negative v-direction of the internal edges located on the dark gray surface mesh of the face. In this study the boundary layer of a face is defined by the transfinite layers of its neighbor faces and consequently the normals for this constrained mesh vertices are initially not necessary to compute. But in order to prevent collisions in the mesh the normals for these vertices are approximately calculated, too. Surrounding normals of a sliding vertex are calculated using equation (3) with $c_i = 2$ if i-th mesh surface s_i contains at least one slidable vertex. A fixed vertex is a vertex that does not move and thus, the normal for this vertex is set to $\underline{0}$. In order to reduce gradients in $\underline{N}_V$ the t-th iterative smoothing step of the unit normal at vertex i with all K direct neighbors

$$\underline{N}_V^{t+1}(i) = \frac{\sum_k^K \underline{N}_V^t(k)}{|\sum_k^K \underline{N}_V^t(k)|}$$

is performed. The final extrusion creates a layer according to the dark gray elements of figure 3. On the one hand the resulting boundary layer consists of prisms that are created by pure ordinary vertex extrusions. On the other hand partly fixed vertex extrusions produce pyramids or tetrahedra for surface elements that contain one or two fixed vertices, respectively.

It is important to mention that the simplified algorithm in the framework does not provide all features mentioned in [8]. Caused by the predefined transfinite layers, it is not necessary to

calculate multiple growth curves of mesh vertices. But it also has be stated that the implemented algorithm is not as general as the algorithm found in the literature.

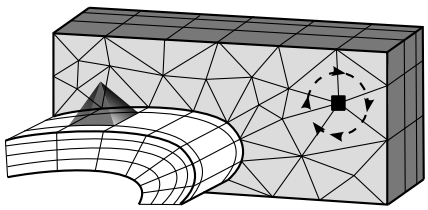


Figure 3: Prismatic layer (dark gray) and Pyramid (black shaded) as coupling element

2.4 STRUCTURED-UNSTRUCTURED COUPLING

The tetrahedral mesh in the resulting flow channel that is not part of the structured mesh block surrounding the blade has to be coupled to the structured hexahedral mesh. The Pyramid Open Method (POM) is a robust coupling algorithm. It is reported that the method is more stable compared to the Tetrahedral Transformation Method, but produces pyramids of lower quality. An advantage is that the POM does not modify or rather divide any existing hexahedra and additionally, the algorithm is easier to implement [9].

Starting with two nonconforming meshes of the tetrahedral and hexahedral part, the method opens pyramids from the coupling face. Figure 3 shows a pyramid (black transparent color) that starts to open from the white structured part. In its original formulation POM inserts a mesh vertex in the center of the quadrangle, subdivides all direct neighboring tetrahedral elements and then opens the pyramid. A quadrangle needs to have two adjacent triangle faces or rather a hexahedron has to share one face to two different faces that belong to two different tetrahedra. The implementation in the framework starts from a hexahedron that shares a face to four different faces of also four different tetrahedra. The shared face already contains the hanging vertex. After the insertion of a zero volume pyramid element the nonconforming mesh is topologically coupled. The iterative procedure takes each of the J quadrangle mesh vertices $\underline{\mathcal{Q}}$ and each from the K vertices that belong to an adjacent tetrahedron $\underline{\mathcal{T}}$ into account. The initial position ($t = 0$) of the apex vertex $\underline{C}$ of the i-the pyramid

$$\underline{\mathcal{C}}_i^{t=0} = \frac{\sum_j^J \underline{\mathcal{Q}}_j + \sum_k^K \underline{\mathcal{T}}_k}{J + K}$$

moves the apex vertex into the tetrahedral mesh and thus increases the pyramid's volume. Due to the mesh deformation it is possible that some adjacent tetrahedra become illegal (negative volume). An iterative procedure checks the volume of all adjacent tetrahedra for the t-th iteration step. If illegal elements occur the apex moving

$$\underline{\mathcal{C}}_i^{t+1} = (1 - r)\underline{\mathcal{C}}_i^t + r\underline{\mathcal{C}}_i^0$$

is constrained using a weighted average of the 0-th (initial) and the t-th vertex position. The relaxation parameter r is a user-defined scalar constant. Following the proposal of the algorithm in [9] a quality check for the shape metric is performed, too. The modified procedure does not check element quality due to multiple mesh optimization steps following the POM and

consequently creates pyramidal elements with a larger volume. Thus, the quality and volume of the deformed tetrahedral elements are lower after the implemented POM compared to the original algorithm.

2.5 PRE- AND POSTPROCESSING

For this study the open source Computational Fluid Dynamics software package OPENFOAM is used to simulate the flow field in the hydraulic machine [10]. All simulation files and data are directly written by the framework. Parts of OPENFOAM are connected to the framework as shared libraries e.g. used boundary conditions, mesh read- and write support or some general input- and output (IO) methods. It prevents avoidable conversion steps due to different formats. Additionally, the direct pre-processing of OPENFOAM via its provided libraries gives the possibility to write all necessary input files and thus is implicitly corrected if the underlying version changes. Furthermore, it is possible to use the large amount of provided functions for field operations in the sense of OPENFOAM. Therefore, also post-processing profits by the direct connection to the simulation library, e.g. the mass flow averaged circumferential velocity distribution on arbitrary user-defined surfaces or a pressure distribution on spanwise constant blade cuts are possible to calculate.

3 MODELS

For this study two typical hydraulic machines are investigated. The geometrical appearance of both, a Francis and a Propeller runner, is fairly different, but the underlying used geometrical parameterization is very similar. The template that is defined by the geometrical parameterization of the machine is either automatically or manually adjusted (change of DDoF) to fit the existing geometry. The intention of this study is not to model the existing geometry as exact as possible and not to "re-simulate" the machines. In fact, the machines serve as examples for the automatic hybrid grid generation method and in addition to discover the peculiarities of this general workflow approach.

3.1 GEOMETRICAL PARAMETRIZATION

In classical refurbishment tasks the meridional shape outside the runner is fixed and therefore not considered as DDoF for the new design. Consequently, the meridional contour needs not to be parameterized and thus, a defined number of points, picked from the contour, is imported into the framework. The blade and its center plane is transferred to the framework, too. It is important to understand that for an existing machine it is initially not possible to find its original parameterization. In reality every machine designer is free to use his own parameterization. Both machines are modeled with a three point mean line represented by a second order B-Spline defined with three control points. It is possible to define the inlet α and outlet angle β but also the meridional length of the blade H. The circumferential length of the blade is defined by the ratio r. The interested reader is referred to [4] for a detailed description. The sum of the thickness distribution $y(u,t)$ for a four-digit NACA profile with maximum thickness t_{max} and parameter coordinate u and the trailing edge thickness t_{te}

$$t = y(u, t_{max}) + t_{te}$$

defines the blade thickness [11]. The coordinate u of the blade surface points in meridional direction, whereby v represents the spanwise direction from hub to shroud. Both machines have a spanwise constant trailing edge thickness ($t_{te} \neq t_{te}(v)$). The Francis runner also has

a spanwise constant ($t_{max} \neq t_{max}(v)$) maximum thickness t_{max}. In contrast the Propeller's maximum thickness

$$t_{max} = t_0(1 - v) + t_1 v$$

linearly depends on the spanwise coordinate v with thickness t_0 and t_1 at hub and shroud, respectively. Both blades of the machines have equal thickness distributions on suction and pressure side. Table 2 lists angles and main dimensions with unit in squared brackets of the two machines. Functions b_F and b_P, with their input in brackets, represent the parameterization set of the Francis and Propeller runner. Increasing numeric index means increasing spanwise position from hub to shroud. DDoF with three positions (1, 2 and 3) are connected via a B-Spline function, two positions of a DDoF (0 and 1) are linearly connected. As already mentioned the values of table 2 are not known or able to be easily calculated for an existing blade. Both machines are approximately reparameterized with the geometrical template by reparameterizing discrete points on the imported initial blade.

	$b_F(\alpha_{1-3}, \beta_{1-3}, H_{1-3}, r, t, t_{te})$	$b_P(\alpha_{1-3}, \beta_{1-3}, H_{1-3}, r_{1-3}, t_0, t_1, t_{te})$
[°]	$(\alpha_1, \alpha_2, \alpha_3) = (88, 59, 45)$	$(\alpha_1, \alpha_2, \alpha_3) = (76, 34, 20)$
[°]	$(\beta_1, \beta_2, \beta_3) = (45, 28, 18)$	$(\beta_1, \beta_2, \beta_3) = (53, 26, 17)$
[m]	$(H_1, H_2, H_3) = (0.27, 0.24, 0.19)$	$(H_1, H_2, H_3) = (0.070, 0.070, 0.063)$
[m]	$r = 0.75$	$(r_1, r_2, r_3) = (0.400, 0.525, 0.500)$
[m]	$t_{max} = 0.005$	$(t_0, t_1) = (0.1, 0.025)$
[m]	$t_{te} = 0.005$	$t_{te} = 0.001$

Table 2: Angles and main dimensions of Francis and Propeller runner

3.2 NUMERICAL SETUP

OPENFOAM (foam-extend version 3.1) provides "MRFSimpleFoam", a solver for systems with a rotating frame of reference using the SIMPLE algorithm [12]. The investigations only include rotating parts of the machine and thus, it is not necessary to have any stator-rotor interface. Additionally, the solver produces a stationary solution. Volume or mass flow is predefined by a given velocity distribution at the inlet patch and outflow is against zero pressure. On the periodical patches the individual grid elements are nonconforming and therefore are coupled by a general grid interface. The outflow of the domain is enforced for the Francis runner by the use of the "inletOutlet" boundary condition. To take care of turbulent flow effects the simulation uses a standard k-ϵ turbulence model with a given turbulent kinetic energy intensity of 10% at inlet. The mixing length L for the calculation of the dissipation rate ϵ can be estimated based on a pipe diameter d_h

$$L = 0.038 d_h$$

for a fully developed pipe flow. Diameter d_h is replaced with the channel height of the machine. All calculations run 1000 iterations in total thereof the first 100 iterations are laminar to get a stable initial solution.

The discretization scheme can be adjusted in OPENFOAM. It is possible to change the approximation of the gradients and divergence terms in a numerical simulation. It is reported that the least square discretization scheme tends to be more oscillating on tetrahedral meshes. The gradient approximation is also not as accurate as on structured or rather hexahedral meshes and

thus, it is recommended to use gradient limiters. For this study, unless otherwise specified, a cell limited scheme with blending factor 0.33 is used that tends to be less dissipative than a face limited scheme [13].

4 SENSITIVITY ANALYSIS

For complex engineering models it is a challenging task to get knowledge about the influences of the input to the output. Especially for automatic optimization processes, it is important to ensure that the inputs like geometry variations that are adjusted by an optimizer have a predominant influence to the hydraulic properties, e.g. torque, head and efficiency. Other uncontrollable influences such as grid variations need to have less dominance on the output. A sensitivity analysis (SA) is a useful tool to investigate this behavior [14]. For the sampling and evaluation of the study the open source MATLAB/Octave toolbox "Sensitivity Analysis For Everybody" (SAFE) is used [15]. For all studies the samples are created by the Morris method that has a better economy of the sampling plan [16].

4.1 BASIC EQUATIONS

For a numerical model $\mathbb{M}$ that has an input $\underline{x}$ with K elements and an output y

$$y = \mathbb{M}(\underline{x}) = \mathbb{M}(x_1, ..., x_K)$$

an elementary effect (EE) is defined as

$$\mathrm{EE}_i(y) = \frac{\mathbb{M}(x_1, ..., x_{i-1}, x_i + \delta, x_{i+1}, ..., x_K) - \mathbb{M}(\underline{x})}{\delta} \,. \tag{4}$$

For each input i one or more EE of the output y can be determined between x_i and $x_i + \delta$. Equation (4) can be interpreted as a discrete gradient and gives the variation of the output y with respect to the i-th input x_i. A desired number of R EE for each of the K inputs require for a simple sampling method $2RK$ evaluations of $\mathbb{M}$. In the following the distribution of EE for the output y of an input i is denoted as $\mathrm{EE}_i(y)$. The chosen Morris sampling method has a better economy and requires $R(K + 1)$ evaluations of the model [16]. It is the main aim of the method to determine which input is

(a) negligible,
(b) linear or additive,
(c) non-linear or
(d) involved in interactions with other inputs.

At first, this study is not distinguishing between (c) and (d). The other distinguishing criteria are explained later. Common practice is to calculate the standard deviation $\mathrm{std}(\mathrm{EE}_i(y))$ and average $\mathrm{mean}(\mathrm{EE}_i(y))$ for i-th input. In its simplest interpretation a high mean value indicates inputs with a high overall influence and a high standard deviation is a hint for a highly dependent input that may also interact with other inputs [16]. For the application of the Morris sampling method the standard error of the mean (SEM)

$$\mathrm{SEM} = \frac{\mathrm{std}(\mathrm{EE})}{\sqrt{R}} \tag{5}$$

gives an additional indication whether an input is considered as significant or not [16]. Output values with dimensions are designated with $\mathscr{Y}$. Mean values of all samples of one run, e.g. $\mathrm{mean}(\mathscr{Y}(M))$, are used to generate dimensionless output values Y, e.g. $Y(M) = \frac{\mathscr{Y}(M)}{\mathrm{mean}(\mathscr{Y}(M))}$.

5 RESULTS

A first investigation uses the Francis runner according to table 2 and only varies numerical and mesh parameters while keeping the DDoF fixed. Table 3 gives all parameters and ranges. The symbol "⇔" means that this parameter is adjusted from the lower to the upper bound according to its column of the table.

$n_H[\%]$	$p_H[\%]$	$n_{dT}[\%]$	$n_{dH}[\%]$	$g[-]$	$g_P[-]$	$g_U[-]$	$g_{\phi,U}[-]$	$l[-]$	$s[-]$
0	0	0	0	0.20	0.20	0.20	0.20	0.20	0.20
⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔
50	3	30	50	0.40	0.40	0.40	0.40	0.40	0.40

Table 3: Parameters with ranges for a run with $R = 20$ and $K = 10$

In order to simulate different mesh variations that occur in an automatic optimization process an implemented mesh operator perturbs an initial given mesh. Perturbations are inserted in the mesh in two variations. On the one hand each edge of n_H percentage of all hexahedra is shrunk by p_H percentage of its initial length. On the other hand n_{dT} percentage of tetrahedra are divided in four new tetrahedra or n_{dH} hexahedra are split in six pyramids. Both division strategies are done by inserting a new mesh vertex in the barycenter of the initial tetrahedron or hexahedron. A random generator creates indices for mesh elements that are perturbed. Both hexahedron perturbation techniques influence the structured part and thus influences the mesh near the blade. The splitting of tetrahedrons is a local refinement and in most cases a mesh quality modification in the unstructured far field. By the use of a random generator two samples having the same values of n_{dT} differ in the position of local refinement sections while the number of elements remains unchanged. Element perturbation is only done once per element and elements that are connected to prisms at the wall or pyramids at the structured-unstructured coupling are kept unchanged. Table 3 gives the percentage and intensity of all inserted perturbations.

All g, s and l parameters of table 3 are blending factors for the gradient limiters of the discretization scheme used to stabilize the numerical flow field simulation on tetrahedral meshes. Furthermore, it is not unusual to compensate a low mesh quality with high blending values for the limiters. The blending parameters in table 3 are replaced in the discretization file

```
gradSchemes {
  default         cellLimited Gauss linear g;
  grad(p)         cellLimited Gauss linear gP;
  grad(U)         cellLimited Gauss linear gU;
}
divSchemes {
  div(phi,U)      Gauss linearUpwindV cellLimited Gauss linear gφ,U;
}
laplacianSchemes {
  default         Gauss linear limited l;
}
snGradSchemes {
  default         limited s;
}
```

of OPENFOAM. A detailed description of all limiters can be found in [13].

The fluctuations of torque $\widetilde{M}$ and head $\widetilde{\Delta_H}$ within the last 200 iteration steps can be interpreted as a physical convergence criteria. Figures 4a and 4b give the EE plots of the fluctuations. According to [16] the dashed line with slope $2\,\mathrm{SEM}$ of equation (5) is plotted. According to equation (5), SEM is the standard deviation of the distribution divided by the square root of the

sample size. In general, the mean value of a random sample is not correct or rather can only be estimated within a range of uncertainty. SEM is a confidence interval of the estimated mean value and thus, a quantity for the uncertainty. The prefactor 2 gives the 95-th percentile of the estimated mean value based on the assumption of a normal distribution. In other words, the estimated mean value $\mathrm{mean(EE)}$ varies with a probability of 95% between

$$\mathrm{mean(EE)} - 2\mathrm{SEM}$$

and

$$\mathrm{mean(EE)} + 2\mathrm{SEM}\ .$$

In that sense, a calculated mean value $\mathrm{mean(EE)}$ that lies on the left side of the dashed line can also be a mean value of zero that spreads caused by taking only a random sample into account. For that reason input parameters fulfilling the inequality

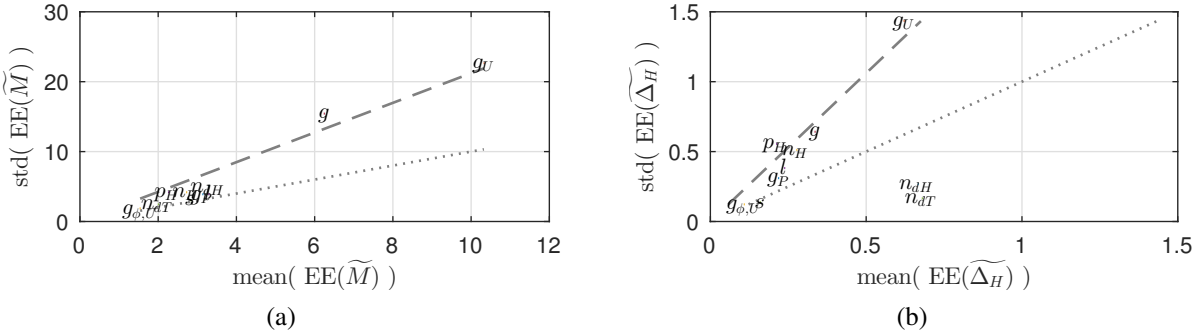
$$\mathrm{std}(EE) > 2\mathrm{SEM} \tag{6}$$

could have a mean value of zero.

The dotted line is the bisect line and hence indicates points that have equal standard deviation and mean. EE of input parameters that lie on the right side of the dotted line and additionally fulfill

$$\mathrm{std}(EE) \ll \mathrm{mean}(EE) \tag{7}$$

have a linear influence to the output and no interaction effects with other inputs. Whereby points located on the left side have non-linear and/or interaction effects [17]. Inputs that are very close to the origin are assumed to have no influence. It is obvious that g_U is separated in both figures 4a and 4b and additionally, it is located on the left side of the dashed line. The value g_U is the blending of the limiter of the $\mathrm{grad}(U)$ terms for the numerical discretization. Additionally, figure 4b gives the indication that mesh perturbations due to hexahedron and tetrahedron division are more critical regarding $\widetilde{\Delta_H}$. Or in other words $\widetilde{\Delta_H}$ seems to be more sensitive to the mesh. However, it has also to be stated that overall fluctuations of both physical quantities are quite small. The mean value of $\mathscr{Y}(\widetilde{M})$ and $\mathscr{Y}(\widetilde{\Delta_H})$ is 0.02% and 0.41%, respectively. Parameters n_{dH} and n_{dT} are on the right side of the dotted line and satisfy equation (7). For that reason $\widetilde{\Delta_H}$ linearly depends on both perturbations. All other numerical setup parameters lie close to the origin or close to the dashed line.



(a) (b)

Figure 4: EE plot for Francis runner of (a) $\mathrm{EE}(\widetilde{M})$ and (b) $\mathrm{EE}(\widetilde{\Delta_H})$

The hydraulic losses H_V are given by the energy equation

$$H_V = \Delta_H - \frac{M\omega}{\rho Q g}$$

with head Δ_H, torque M, angular velocity ω, density ρ, volume flow Q and gravitational constant g. Figure 5a gives the plot of the distribution of hydraulic losses $Y(H_V)$ versus the number of divided hexahedra n_{dH}. The gray solid lines are linear regressions that subtend the sample $\min(Y(M))$ and $\max(Y(M))$. The H_V coordinate of the intersection points with the bottom and the upper linear regression are given in figure 5a as b_{min} and b_{max}, respectively. The slope of the regression is named m and given in the figure, too. The lines are not necessarily bounding lines of all samples. Figure 5b is the equivalent plot with respect to the number of divided tetrahedra n_{dT}. Table 4 gives the number of overall, tetrahedra, pyramids and hexahedra mesh elements.

	min	max
overall $[10^3]$	621	940
tetrahedra $[10^3]$	345	446
pyramids $[10^3]$	21	338
hexahedra $[10^3]$	129	182

Table 4: Number of overall elements and different element types

It is obvious that the division of hexahedra produces more spread and lower slope compared to division of tetrahedra. It is a well known phenomena in simulation that refining a mesh may cause changes in the physical quantities. Both figures 5a and 5b show the same trend. A higher number of divided elements resulting in a finer grid resolution increases the hydraulic losses. The comparison of points with equal abscissa values gives an indication of output fluctuations, i.e. H_V, caused by an uncontrollable change of mesh characteristics due to the randomly positioned refinement of tetrahedral elements. Having this in mind the bounded distance in H_V direction of the section, or equivalent to the difference between b_{min} and b_{max}, gives the amount of fluctuations that are uncontrollable inserted due to mesh modifications. According to figures, the fluctuations of hydraulic losses H_V for hexahedra and tetrahedra splitting are approximately 9% and 3%, respectively. It has to be mentioned that hexahedra splitting is not an uncontrollable mesh change, because under normal circumstances this does not occur. But in terms of getting an indication for the behavior of unstructured meshes the observations show that physical properties are sensitive to the structured part of the mesh.

Summarizing, the influence of the uncontrollable mesh characteristics on the output results, caused by different orderings of tetrahedrons, is small and in addition of linear character. Consequently, a highly non-linear and fluctuating behavior of the mesh is not observed.

A second run using the Francis runner is created by an additional variation of geometrical DDoF namely blade inlet (α_1, α_2 and α_3) and outlet angles (β_1, β_2 and β_3). The mesh parameter λ is the thickness of the structured mesh block that surrounds the blade. In addition, the structured part that consists of four hexahedral blocks is varied. Only the hexahedral block located on the blade tip is changed by moving mesh vertices on hub ($d_{0,0}$ and $d_{1,0}$) and shroud ($d_{0,1}$ and $d_{1,1}$). Last column "[...]" in table 5 has to be seen as a link to columns g, g_P, g_U, $g_{\phi,U}$, l and s of table 3 resulting in a run that contains 17 input parameters. The symbol "$\pm$" means that a

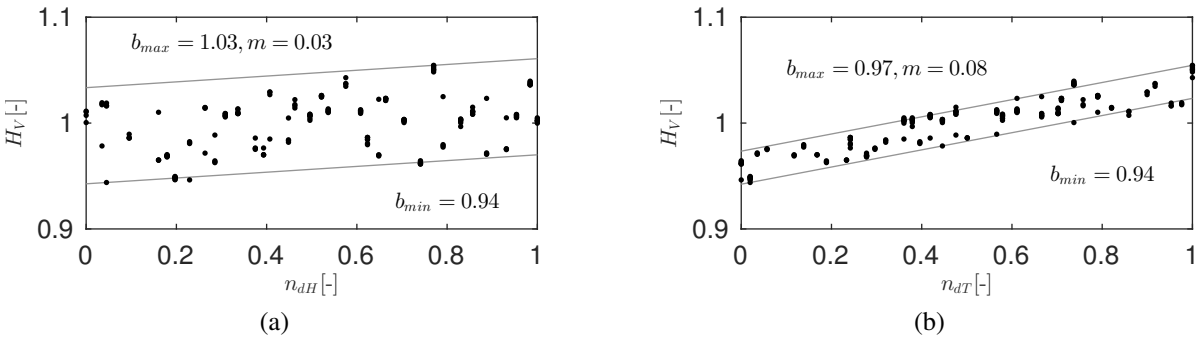


Figure 5: Hydraulic losses for Francis runner versus normalized input (a) n_{dH} and (b) n_{dT}

parameter is adjusted by addition or subtraction by the given bounds to the initial value listed in table 2. If samples fail it may be necessary to remove more than only the failed samples, caused by the Morris sampling method. Following a recommendation of [17] the value of R should be between 4 and 10. For the sake of completeness it is mentioned that each failed sample can, but must not necessarily, reduce R by one. It depends on the distribution of failed designs. A failed sample can reduce the number of samples in this case by $M + 1 = 18$ samples. In its worst case only 5% failed samples are able to reduce R to zero. A miserable distribution of 4.72% failed designs reduces R to 8 for this run. It is observed that many failed samples have small values of $d_{0,0}$ and high values of $d_{0,1}$. The combination represents a strongly distorted mesh block at the blade tip, regarding the shape from hub to shroud.

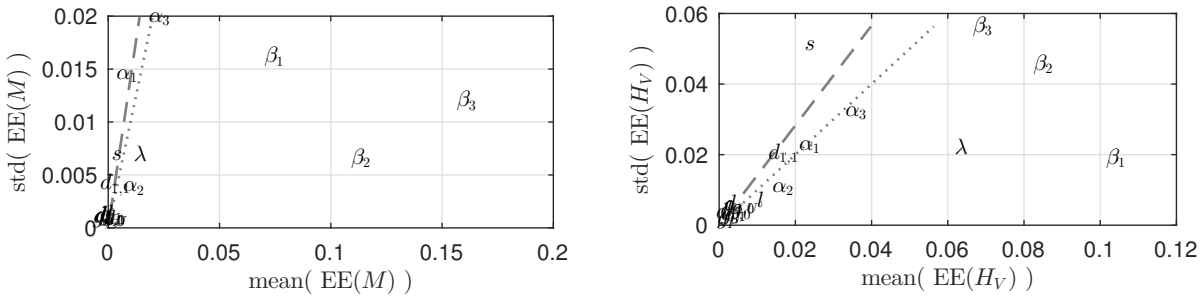
The mean value of $\mathscr{Y}(\widetilde{M})$ and $\mathscr{Y}(\widetilde{\Delta_H})$ are 0.07% and 0.25%. It is important to have in mind that for this study also geometrical changes are allowed and thus, it is stated that some samples may not have a stationary solution of the flow field. Furthermore, varying blade angles at three points for inlet and outlet may also result in an uncommon geometrical shape of the runner. In fact, it is possible that some samples have a set of DDoF that produce a geometry that does not make sense from a well skilled engineer's point of view. For that reason a simulation of the flow may produce high fluctuations if non-stationary phenomena occur. The high $\max(\mathscr{Y}(\widetilde{M}))$ and $\max(\mathscr{Y}(\widetilde{\Delta_H}))$ of 1.04% and 1.91%, respectively, may confirm the assumption.

λ[m]	$d_{0,0}[-]$	$d_{0,1}[-]$	$d_{1,0}[-]$	$d_{1,1}[-]$	$\alpha_1[°]$	$\alpha_2[°]$	$\alpha_3[°]$	$\beta_1[°]$	$\beta_2[°]$	$\beta_3[°]$	[...]
0.015	0.45	0.45	0.51	0.51	$+2$	$+2$	$+2$	$+2$	$+2$	$+2$	...
$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\pm$	$\pm$	$\pm$	$\pm$	$\pm$	$\pm$	...
0.04	0.49	0.49	0.55	0.55	-2	-2	-2	-2	-2	-2	...

Table 5: Extract of parameters with ranges for a run with $R = 20$ and $K = 17$

Figure 6a shows the EE plot of torque M. It is obvious that the three outlet angles β_1, β_2 and β_3 are separated from the other input parameters. According to equation (7) β_3 shows the strongest linear behavior on M. Inlet angles at shroud α_3 and hub α_1 are also separated. The other geometrical blade parameter α_2 lies close to origin like almost all other numerical parameters and thus, they are considered to have no influence on the output M. The decision whether an input is considered to have influence or not has always been done in relation to other inputs. Angles α_1 and α_3 have a similar amount of $\mathrm{mean}(\mathrm{EE}(M))$ but differ in $\mathrm{std}(\mathrm{EE}(M))$. The location close to or on the left side of the dashed line is an indication that the mean value is

zero. According to figure 6b mesh block thickness λ has a much higher mean value compared to figure 6a. An automatic optimization procedure to reduce hydraulic losses that is not able to distinguish between geometrical and numerical influences may produce an unwanted dependency on the numerical parameters, e.g. mesh block thickness λ. The mesh block thickness has a similar mean value as the blade angle β_3 and a similar standard deviation as β_1. Thus, the structured mesh block surrounding the blade has a significant influence on the physical properties. It is worth noting that performing the same run with an increased mesh block thickness λ of $0.035 \Leftrightarrow 0.04$ compared to table 5 shifts mean value and standard deviation of $\mathrm{EE}_\lambda(H_V)$ below 0.01. Figures 6a and 6b clearly show the essential tendency that the other numerical setup parameters do not vary the output in the same order as the geometry variants. The most critical numerical setup parameters are the surface normal blending factor are s and the mesh block position on shroud $d_{1,1}$. Inequality (6) is nearly true for both parameters.



(a) (b)

Figure 6: EE plot of (a) torque M and (b) head loss H_V for geometry variants

Table 6 gives the spread of an output o

$$\overline{o} = \frac{\max(Y(o)) - \min(Y(o))}{\mathrm{mean}(Y(o))}$$

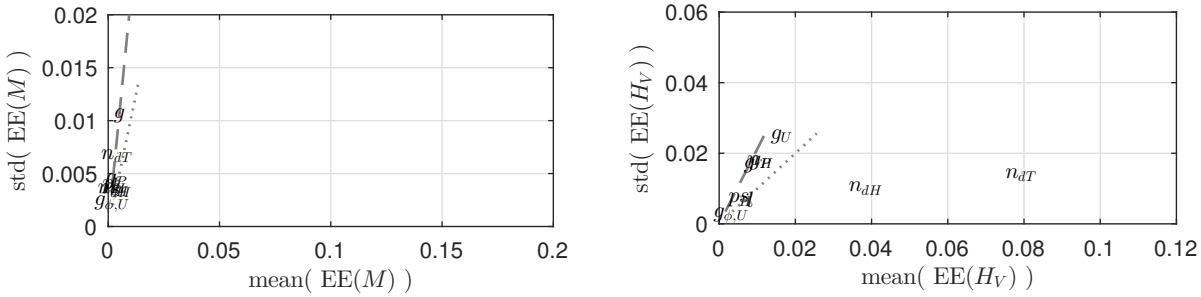
for different physical properties. All spreads are increased if geometric changes are allowed. The spread of hydraulic losses $\overline{H_V}$ is only doubled. According to the observations of figure 5b the estimated uncontrollable fluctuation is added to the table in brackets. An important point to have in mind is that the efficiency η depends on the losses H_V. The above mentioned procedure of 5b applied to η gives a corrected estimated uncontrollable fluctuation of 0.35%.

	without	with
$\overline{M}[\%]$	0.49	24.23
$\overline{H_V}[\%]$	$11.07(3.12)$	21.40
$\overline{\Delta_H}[\%]$	1.31	23.56
$\overline{\eta}[\%]$	$1.27(0.35)$	1.30

Table 6: Comparison of runs without and with geometrical modification

Figures 7a and 7b give the EE plot of torque M and Δ_H for the fixed geometrical run. The axes are equally scaled to figures 6a and 6b. For the output M all numerical or mesh parameters

are located close to the origin and consequently in the presence of allowed geometrical modifications, these inputs are considered to have no influence to the output. As expected the EE plot of Δ_H contains the linear dependent inputs n_{dH} and n_{dT}. The location of n_{dH} and n_{dT} in figure 7b is a confirmation of figures 5a and 5b. Tetrahedral division n_{dT} shows a more linear behavior than hexahedral division n_{dH}. It is equivalent to a higher slope and a lower distance of the bounded section in figure 5b compared to figure 5a.



(a) (b)

Figure 7: EE plot of (a) torque M and (b) hydraulic losses H_V without geometrical changes

The observations again show that influences by uncontrollable changes do not occur. Moreover, the observed dependency of mesh block thickness and physical properties are a controllable influence that shows in addition a linear behavior. It is again a confirmation that a comparison of samples does make sense and is not uncontrollably affected by numerical setup or mesh parameters.

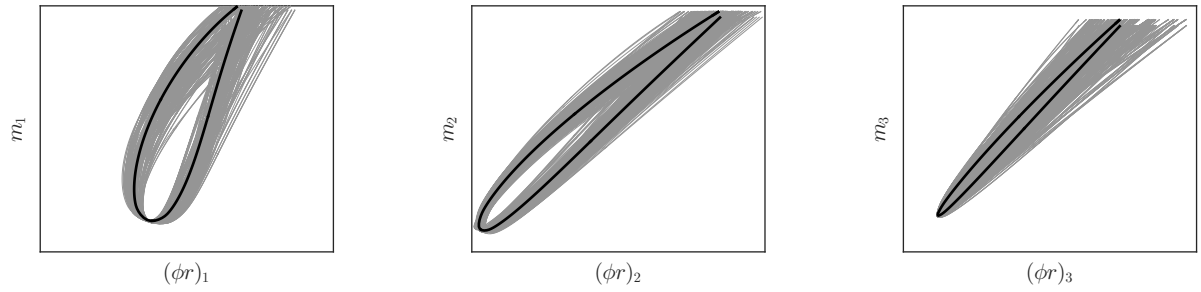
The third investigation uses the Propeller runner with parameterization given in table 2. All upper and lower bounds of the input parameters are shown in table 7. All inputs except the mesh block thickness are geometrical DDoF explained in section 3.1. Overall number of mesh elements is between 555 and 626 thousand cells and the structured part contains 95.4 thousand hexahedra ($15\% - 17\%$). The structured-unstructured coupling consists of 7.95 thousand pyramids ($1.3\% - 1.4\%$).

$\alpha_1[°]$	$\alpha_2[°]$	$\alpha_3[°]$	$\beta_1[°]$	$\beta_2[°]$	$\beta_3[°]$	$r_1[-]$	$r_2[-]$	$r_3[-]$	$t_0[\mathrm{m}]$	$t_1[\mathrm{m}]$	$\lambda[\mathrm{m}]$
$+5$	$+5$	$+5$	$+5$	$+5$	$+5$	0.3	0.3	0.3	0.06	0.0125	0.005
$\pm$	$\pm$	$\pm$	$\pm$	$\pm$	$\pm$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
-5	-5	-5	-5	-5	-5	0.7	0.7	0.7	0.14	0.0375	0.015

Table 7: Parameters with ranges for a run with $R = 20$ and $K = 12$

Figures 8a, 8b and 8c show all spanwise constant blade cuts in the circumferential-meridional ϕr-m coordinate system at three different locations. The solid black line indicates the reference state that is as close as possible to the original geometry. Each gray solid line represents one sample of the run. The figures do not have equal ϕr and m axes and hence are not conformal. For this run meridional blade length H_1, H_2 and H_3 in m-direction is fixed according to table 2. For that reason a change of $r_1 - r_3$, $\alpha_1 - \alpha_3$ or $\beta_1 - \beta_3$ requires an increment or decrement of the circumferential blade length ϕr.

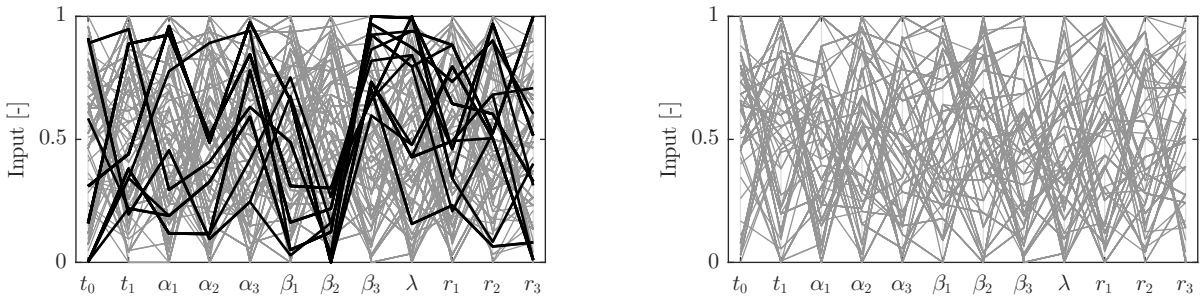
Figure 9a shows the normalized input parameter set of failed samples as black and in the background all samples with gray solid lines. In total less than 9% failed samples are produced



(a) (b) (c)

Figure 8: Spanwise constant blade cuts at (a) hub, (b) mid and (c) shroud

for this run. It is conspicuous in figure 9a that strong differences between outlet blade angles β_2 and β_3 are an indication for failing. In this case R is reduced from 20 to 14. Figure 9b gives an overview of all successful calculated and evaluated samples. It is assumed that the input parameter space is still well covered by the remaining samples shown in figure 9b.

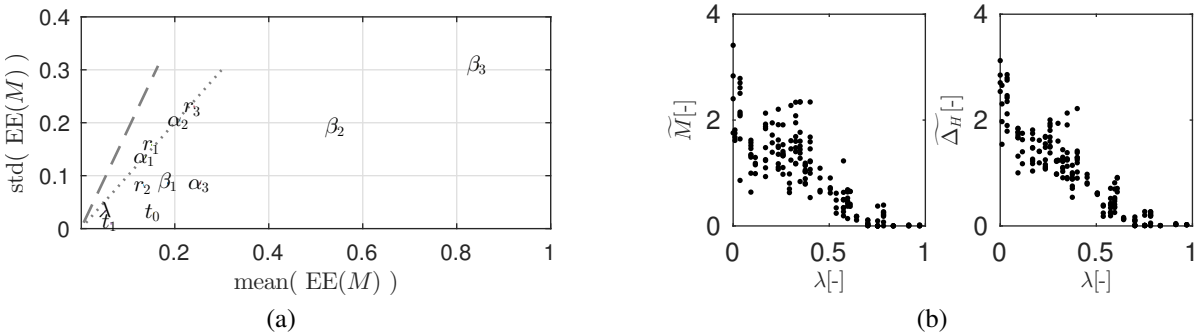


(a) (b)

Figure 9: (a) Calculated (gray) with failed (black) and (b) evaluated (gray) samples

Figure 10a visualizes the EE plot of torque M. The input parameters of figure 10a are on the right side of the dashed line which is an indication that the $\mathrm{mean(EE)}$ of the inputs are not zero. It is also clear from the figure that β_2 and β_3 are the main influencing parameters to the output and according to equation (7) do have linear effect on the output M. Moreover, inputs are clustered in three groups. Input parameters mesh block thickness λ and blade thickness at shroud t_1 are very close to each other. Other mean values of the inputs are located with a distance to origin and thus do influence M. In order to reduce mesh dependency it would be preferable to separate λ from the other inputs. Figure 10b gives $Y(\widetilde{M})$ and $Y(\widetilde{\Delta_H})$ versus normalized mesh block thickness λ. According to the figure the physical convergence criteria for this run strongly depends on the mesh block size. The average of $\mathscr{Y}(\widetilde{M})$ and $\mathscr{Y}(\widetilde{\Delta_H})$ are 4.75% and 7.02% and thus, they are very high compared to the previous observations with the Francis runner. It is important to mention that different samples of figure 10b do not only vary in the input value λ. As already mentioned a geometry modification may prevent a sample to have a stationary flow solution.

The Propeller runner is again simulated with exactly the same SA configuration except the initial range of λ of table 7 is reduced to $0.0125 \Leftrightarrow 0.015$ caused by the observations of figure 10b. The limited range corresponds to abscissa values from 0.75 to 1.00 of figure 10b. In this run R is reduced from 20 to 12 due to 8.8% failed samples. In the following the first



(a) (b)

Figure 10: (a) $\mathrm{EE}(M)$ plot of torque M and (b) fluctuations $\widetilde{M}$ and $\widetilde{\Delta_H}$ of Propeller runner

Propeller run is named "thin" and the second run "thick". According to table 8 average of $\mathscr{Y}(\widetilde{M})$ and $\mathscr{Y}(\widetilde{\Delta_H})$ are reduced to 0.30% and 0.80%. Only the 95-th percentile of $\mathscr{Y}(\widetilde{M})$ is below one percent. In addition, a too thin mesh block and consequently a flow field surrounding the blade that is discretized with more tetrahedral elements may promote smoothing effects. It is a known phenomena that tetrahedra have a more diffusive behavior than hexahedra [1]. According to table 8 mean values and percentiles of torque $\mathscr{Y}(M)$ differ significantly. The number of elements normal to the blade is not changed between the runs and thus y^+_{blade} values are increased at the blade for the thick run. Values of the dimensionless distance to the wall y^+ should be greater than 30 and smaller than 100 to model the boundary layer correctly [18]. For the Propeller runner the y^+ values on the blade are bigger than values calculated on hub and shroud. Thus overall average of y^+ is smaller than 100.

	thin				thick			
	mean	25-th	75-th	95-th	mean	25-th	75-th	95-th
$\mathscr{Y}(\widetilde{M})[\%]$	4.75	1.60	7.16	11.11	0.30	0.01	0.38	1.22
$\mathscr{Y}(\widetilde{\Delta_H})[\%]$	7.02	1.79	10.46	17.12	0.80	0.03	0.93	3.23
$\mathscr{Y}(M)[\mathrm{Nm}]$	15.6	13.2	18.8	22.5	16.7	14.4	19.8	22.8
$\mathscr{Y}(\Delta_H)[\mathrm{m}]$	2.68	2.28	3.20	3.89	2.68	2.31	3.16	3.66
$\mathscr{Y}(y^+_{blade})[-]$	98	76	118	144	131	124	138	144

Table 8: Comparison of runs with thin and thick mesh block of Propeller runner

Figures 11a, 11b and 11c give the obtained values of $\mathscr{Y}(M)$, $\mathscr{Y}(\Delta_H)$ and $\mathscr{Y}(\eta)$, respectively. Gray points represent the thin run and black points the thick run. Torque M is underestimated for the thin run and thus figure 11a gives again a graphical hint of the observations of table 8. The underestimation of M and overestimation of Δ_H leads to an underestimation of runner efficiency η, too. The plot of η in figure 11c gives the assumption that the shape of the curve obtained from the two runs remains quite similar. Hence, only an offset between both curves is assumed.

All samples of $\mathscr{Y}(M)$, $\mathscr{Y}(\Delta_H)$ and $\mathscr{Y}(\eta)$ are sorted for the thin run in ascending order for each output value separately. The ordering is then used for sorting also the thick run to get the same order of parameter sets. A loop over all elements of an output of the thick run, with the fixed ordering obtained from the thin run, checks if the current element is smaller than the next one. It gives the number of correctly ordered elements obtained in both runs. The evaluation

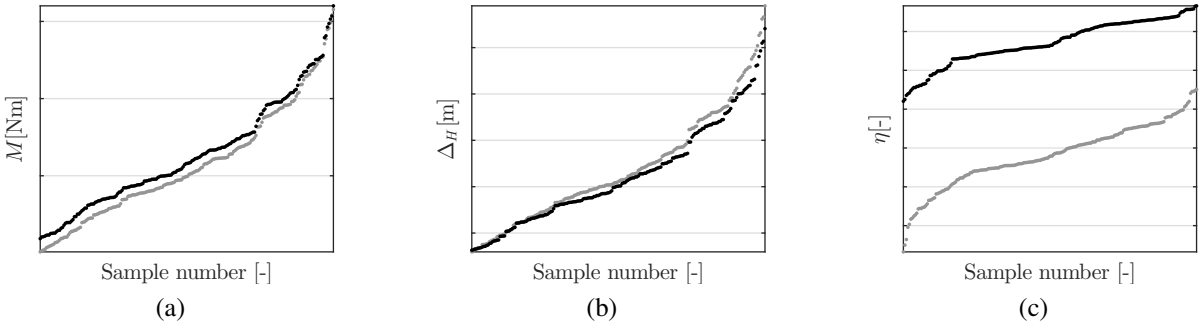


Figure 11: Samples of (a) $\mathscr{Y}(M)$, (b) $\mathscr{Y}(\Delta_H)$ and (c) $\mathscr{Y}(\eta)$ for thin (gray) and thick (black) run of Propeller runner

contains 198 samples and thus 197 checks are performed. All three outputs M, Δ_H, η and, additionally, the hydraulic losses H_V give the maximum number of successful checks. This is interpreted as a strong indication that the shapes of the curves are similar and in addition tendencies between same samples are not affected by different mesh block thicknesses.

Figure 12a, 12b and 12c shows the circumferential u_φ, radial u_r and meridional u_z mass flow averaged velocity components downstream the runner for a point close to the shroud. In order to get an indication of the flow field the velocity curves at three meridional constant positions downstream the runner are calculated and ten equidistant monitor points are used to perform the above mentioned ordering-checking procedure. For each monitor point at least 195 samples ($\approx 99\%$) are correctly ordered. An observed underestimation of the circumferential velocity u_φ for the thin run goes along with the observation of an underestimated torque M.

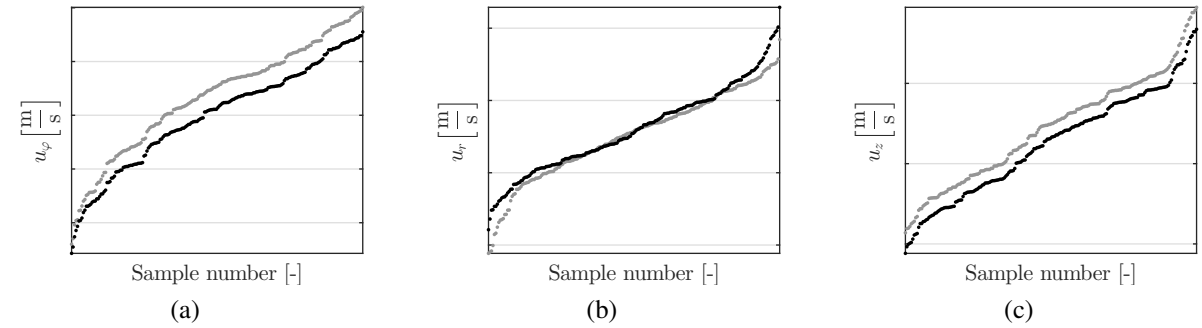


Figure 12: Cylindrical velocity components near shroud of samples (a) $\mathscr{Y}(u_\varphi)$, (b) $\mathscr{Y}(u_r)$ and (c) $\mathscr{Y}(u_z)$ for thin (gray) and thick (black) run of Propeller runner

The ordering of the inputs in figure 13 confirms the assumption of constant tendencies. Inlet angle α_2 and mean line ratio r_1 move downwards compared to figure 10a. Mesh block thickness λ has also moved in the direction of origin. All input values for the thick run are shifted to a lower std(EE(M)) value. This seems to be a consistent observation with figure 11a where torque tends to have a higher spread for the thin run. Consequently, EE have to be higher.

6 CONCLUSIONS AND FURTHER WORK

The developed framework is extended to create a hybrid mesh for different geometries. Besides existing mesh operators a free edge grading operator and a hybrid face operator is used to

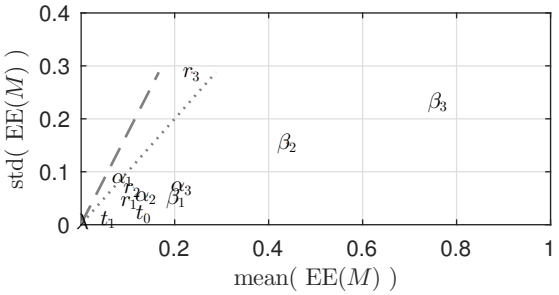


Figure 13: EE plot of torque M of Propeller runner

create the hybrid volume mesh. A modification of the original POM couples the structured with the unstructured mesh part. The prismatic boundary layer is created using a simplified advancing layer technique. Pre-processing as well as post-processing is embedded in the framework. An evaluation chain, completely implemented in the framework, is used to perform an automatic sample evaluation for different runs.

A general geometrical parameterization, that differs only in the thickness distribution and the number of DDoF, defines a typical Francis and Propeller runner. The hybrid mesh is used to perform a stationary, incompressible simulation, using periodic boundaries and a standard k-ϵ turbulence model, to obtain a solution of the flow field. Overall four runs with different modifiable input parameters of the two machines are performed. An open source sensitivity analysis toolbox is used to discover the main influences. Different input values are characterized by their effect on different output parameters. For the Francis runner it is observed that in general the effect of uncontrollable mesh variations, caused by local refinement sections or different positions of the sections in the tetrahedral mesh part, do not affect physical quantities. It is also discovered, that torque seems to be less sensitive to those variations than head or hydraulic losses. Moreover, a linear dependency is observed for the hydraulic losses. A user controlled overall increase of number of mesh elements is a critical aspect regarding the influence to the output. The use of gradient limiters for the numerical solution is more critical regarding head, but also does not affect physical properties of the same order than geometry variations. Evaluating the Propeller runner detects a linear dependency between structured mesh block thickness and fluctuations, too. A comparison of two runs with different mesh block thicknesses shows that for torque, head, runner efficiency, hydraulic losses and mass flow averaged velocities tendencies remain unchanged. In addition, the fluctuations of head are reduced significantly if mesh block thickness is increased. Only controllable, and with it avoidable, changes of mesh characteristics influence physical output values in the same range as geometry variations. The fluctuations caused by uncontrollable mesh changes for different geometry variants are negligible. Therefore, it is believed that a successful automatic optimization procedure using hybrid meshes is possible.

An interesting point is to apply the presented workflow to a whole machine and hence perform a sensitivity analysis. The simulation and evaluation of guide vanes, runner and draft tube in the presence of geometrical modifications may improve the understanding of the machine. Based on the presented observations the application of an automatic optimization procedure is of interest. Moreover, an interesting point is if a preliminary sensitivity analysis is able to reduce the amount of parameters for an automatic optimization.

REFERENCES

[1] T. J. Baker, *Mesh generation: Art or science?*, Progress in Aerospace Sciences **41** (2005), no. 1, 29 – 63.

[2] A. Ruprecht, R. Eisinger, E. Goede, *Innovative Design Environment for Hydro Turbine Components*, HYDRO (Bern), 2000.

[3] R. Eisinger, A. Ruprecht, *Automatic shape optimization of hydro turbine components based on CFD*, Seminar CFD for turbomachinery applications **6** (2001), no. 1, 101–111.

[4] A. Tismer, M. Schlipf, S. Riedelbauch, *An object-oriented approach for a highly customizable framework to design hydraulic machines*, Wasserkraftanlagen (Wien, Österreich), Innovationen und Entwicklungsbedarf für eine nachhaltige Entwicklung der Wasserkraft, vol. 18, Technische Universität Wien, Berger, Horn, 11 2014, Paper, Presentation, 213–223.

[5] MatraDatavision, *OpenCASCADE*, http://www.opencascade.com, accessed 2016-01-19.

[6] T. Bahçecioğlu, *A Simple C++ Framework to load classes from dll files in an OO way*, http://sourceforge.net/projects/pugg, accessed 2016-01-19.

[7] C. Geuzaine, J.-F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering **79** (2009), 1309 – 1331.

[8] R. V. Garimella, M. S. Shephard, *Boundary Layer Meshing for Viscous Flows in Complex Domains*, Proceedings of the 7th International Meshing Roundtable, Dearborn, Michigan, USA, October 26-28 1998, 107–118.

[9] S. J. Owen, S. A. Canann, S. Saigal, *Pyramid Elements for Maintaining Tetrahedra to Hexahedra Conformability*, Proceedings on Trends in unstructured mesh generation **220** (1997), 123–129.

[10] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, *A tensorial approach to computational continuum mechanics using object-oriented techniques*, Computers in Physics **12** (1998), no. 6, 620–631.

[11] I. Abbott, A. Von Doenhoff, *Theory of Wing Sections, Including a Summary of Airfoil Data*, Dover Books on Aeronautical Engineering Series, Dover Publications, 1959.

[12] H. Nilsson, *Turbomachinery training at OFW7*, 7th OpenFOAM Workshop (Darmstadt), June 2012, Presentation.

[13] University of Genova, *Tips and Tricks in OpenFOAM*, Introductory OpenFOAM Course (Genova, Italy), February 2015, Presentation.

[14] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, Halsted Press, New York, USA, 2004.

[15] F. Pianosi, F. Sarrazin, T. Wagener, *A Matlab toolbox for Global Sensitivity Analysis*, Environmental Modelling & Software **70** (2015), 80 – 85.

[16] M. D. Morris, *Factorial Sampling Plans for Preliminary Computational Experiments*, Technometrics **33** (1991), no. 2, 161–174.

[17] B. Iooss, P. Lemaître, *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, ch. A Review on Global Sensitivity Analysis Methods, 101–122, Springer US, Boston, MA, 2015.

[18] W. Rodi, *Turbulence Models and Their Application in Hydraulics*, IAHR Monographs, Taylor & Francis, 1993.