Conference Paper, Published Version

Taccone, Florent; Goeury, Cédric; Zaoui, Fabrice; Petralia, Adrien

Pumping station design based on shape optimization process

Pumping station design based on shape optimization process

Florent Taccone, Cédric Goeury, Fabrice Zaoui
National Laboratory of Hydraulics and Environment
EDF R&D
Chatou, France
florent.taccone@edf.fr

Adrien Petralia
Polytech' Lyon / EDF R&D
Lyon, France
Chatou, France

***Abstract*— Optimization applications with TELEMAC are increasing due to interoperability development of the system module. The present work is based on a shape optimization process apply to a real problem: the optimization of the streamline trajectories in front of a pumping station intakes. Deflectors have been designed in the model upstream of the intakes to drive the flow as perpendicular as possible to the intake entrances. The deflector's shape is defined based on two parameters controlling the size and the orientation respectively.**

In a first step, a cost function evaluating the orientation of the streamlines was defined. Then, a study was carried out on these two parameters to estimate, for each deflector, which configuration minimizes the cost function based on TELEMAC-2D runs. Finally, a statistical emulator was used to link the input parameters with the cost function residual. Indeed, this metamodeling technique allowed a simplification of the TELEMAC-2D study, drastically reducing computational times. This was particularly useful to apply an optimization process on the parameters of the shapes, requiring many TELEMAC-2D study runs.

The results of this study allowed identifying an optimal shape for each deflector, while ensuring a certain robustness of the solution.

I. INTRODUCTION

TELAPY, the API interface of the TELEMAC-MASCARET platform [1], allows the application of transverse mathematical tools for hydraulic studies with TELEMAC-2D. Its python interface offers the possibility to encapsulate a TELEMAC-2D run in OpenTURNS algorithms (www.openturns.org). Thus, [2] shows an example of uncertainty quantification on the input parameters of a TELEMAC-2D model. This type of method has also been applied for optimization processes in [3], in particular with automatic calibration applied on the bottom friction coefficient and tidal parameters, in comparison with measured data. [4] presents a schematic case of shape optimization on which the optimization process is applied. This is a fish pass for which shapes are defined to maximize the flow velocity in the central compartment. These shapes are introduced into the contour of the mesh and a cost function is evaluated on the TELEMAC-2D run to assess the relevance of the shapes to the targeted problem. As these shape optimization processes require a lot of evaluation of the numerical model, it is important, in this study with a large domain, to be interested in the metamodeling techniques developed in [5]. Indeed, the creation of a statistical model allows a quasi-instantaneous evaluation of an approximation of the TELEMAC-2D model, where the approximation error can be estimated.

The aim of this study is to optimize the efficiency of pumps in a water intake station by modifying the streamlines upstream. In this framework, this work designs deflectors in the channel, before the intakes, to allow the flow to enter perpendicularly. The deflector shape should therefore allow the best possible response to the problem, while respecting the physical constraints of the pumping station channel.

After presenting the tools used in this study in section II, the first step in this work is to carry out a parametric study. This study is organized as follow:

1. The construction of the TELEMAC-2D model and the associated hydrodynamic parameters are presented in section III.A.
2. The location and shape of the deflectors, the physical constraints associated with its shapes and the parameters describing them will be defined in III.B
3. The cost function to evaluate the perpendicularity of the current lines at the inlet of the intakes is described in III.C.
4. Finally, a large number of TELEMAC-2D simulations are run with randomly drawn deflector parameters to determine which shape minimizes the cost function, as presented in III.D.

In a second step, described in section IV, a statistical model linking the shape definition parameters and the residuals calculated during the parametric study will be created. By applying an optimization algorithm to this statistical model, it is possible to obtain optimal shapes to respond to the problem of perpendicularity of the water lines. The study of the statistical model will also give the possibility to evaluate the smoothness of the solution in order to have an idea of the robustness of the solution (how a small change in the shape parameter influence the cost function evaluation).

II. Presentation of the tools

A. TELEMAC-MASCARET Python tools

1) TelApy

TelApy [1] is a Python module that is part of the TELEMAC system. It is an API, Application Programming Interface, which allows the user to interact with the core of the TELEMAC system coded in FORTRAN using Python scripts. This module allows for example to control the execution of a simulation carried out with TELEMAC-2D by retrieving information at each time step.

2) Postel

Postel [8] is a Python module that is also part of the TELEMAC system. It allows to do post-processing by extracting data from the results file obtained from a simulation. It can be used as a visualization tool to plot the studied domain and the scalar fields representing the state variables of a simulation. In this case, it is mainly used to extract data for the cost function evaluation.

B. Tools used in the SALOME platform

1) GEOM

SALOME's GEOM module is a design software that allows you to create geometric objects (curves, surfaces, points, etc.) in 2D and 3D. This module can be used with the SALOME graphical interface or by using the associated Python library by calling GEOM class functions in a script. This module has been used to define the contour of the calculation domain and the mesh refinement zones.

2) MESH

The MESH module of SALOME is used to mesh objects created in the GEOM module. It contains several algorithms to generate uniform but also more complex meshes. For example, it allows to mesh a domain by defining properties of sub-meshes.

3) YDEFIX

Ydefix is a C++ library of the SALOME platform that allows distributing a series of calculations on the resources of a machine. In this study, it allows us to distribute the calculations of the simulations on a computing cluster. This library can be used through the SALOME platform as a Python module. It allows to execute a Python script sequentially on a large number of processors in at the same time. It is therefore possible, for example, to launch a large number of TELEMAC-2D calculations via the module TelApy simultaneously. However, technical limitations of this module does not permit to run several TELEMAC-2D computations with the internal parallelism of the code.

4) OpenTURNS

OpenTURNS (Open source initiative to Treat Uncertainties, Risks'N Statistics) is a C++ library developed by EDF R&D, Airbus Group, IMACS Engineering, ONERA and PHIMECA allowing the treatment of uncertainties. This library can be controlled via a Python module or via the graphical interface of the SALOME platform. It allows a large number of applications, from data analysis to the creation of statistical models. It also contains several optimizers to solve minimization problems.

III. Parametric study

A. TELEMAC-2D model

The pumping station studied pumps its water from the sea and is subject to tides. The domain represents the sea offshore over a 3 km zone with a 200 m mesh size, then in the channel the mesh size is 2 m. A third refinement zone is defined along the wall on which the pumps are located with a mesh size of 20 cm. A new mesh is generated for each TELEMAC-2D run, keeping these characteristics, including the deflectors in the outline. A single mesh generation takes approximately 20 seconds.

The area of interest for the study is at the end of the channel where there are 20 intakes. The suction flow rate in the pumps are distributed as follows:

- 2 pumps with a flow discharge of 1.1 m^3/s;
- 8 pumps with a flow discharge of 8.125 m^3/s;
- 2 pumps with a flow discharge of 1.1 m^3/s;
- 8 pumps with a flow discharge of 8.125 m^3/s.

Regarding boundary conditions, there are two types in the model. The sea boundary conditions, allowing the representation of the tide in the model, and the pump boundary conditions. Concerning the tidal boundary conditions, in order to have a good representation of both the free surface dimension in the model and the the current velocity, a water height on the edge facing the channel entry and velocities, uniform over the depth, on the lateral edges are imposed. The imposed values are computed using the TPXO database and represent schematic tide of coefficient 120. For the boundary conditions of the pumping station, the suction flow rate imposed is constant, corresponding to the values described previously. In TELEMAC-2D the processing of boundary conditions is done in such a way that the velocity field is imposed perpendicularly to the boundary condition segments. As the objective is to study the direction of the flow at the entrance of the intakes a slight indentation has been created for each intake. They represent the beginning of the openings so that the entrance is not completely forced to be perpendicular. Figure 1 shows the position of boundary conditions and the definition of boundary conditions in the limits for the first 10 intakes (the last 10 being similar).

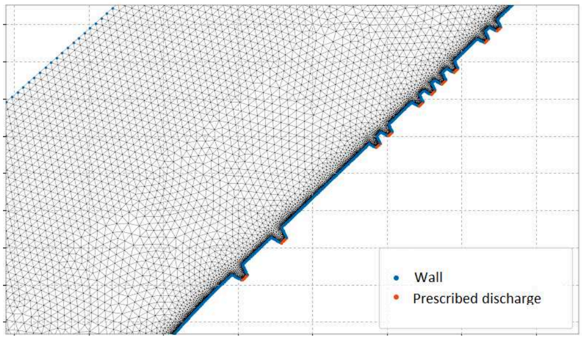

Figure 1: Visualization of the imposition of the pumping station intakes in the model

The calculation starts in a high sea state and lasts one and a half tidal cycle, i.e. 18 hours, and thus 64800 seconds. The beginning of the simulation, during the first half cycle, allows the establishment of the model to be independent from the initial condition. The results are therefore taken into account over the entire tide, in the last two-thirds of the simulation. To represent the friction effects of the bottom, a Manning-Strickler friction law is used, with a Strickler's coefficient set at 57.6 $m^{1/3}/s$. This value has been calibrated in a precedent study using velocity flow measurements with on several ADCP profiles.

B. Definition of the deflector shapes

In order to have a shape that is more adjustable than a simple circle, one opts for the catenary curve equation, a function well known in physics that describes the shape taken by a heavy and flexible cable. This shape is governed by the equation:

$$f(x) = a \cdot \cosh\left(\frac{x}{a}\right),$$

where the parameter a has the dimension of a length, and can be interpreted in its physical sense as the length of the suspended cable.

This function has been modified in this study to represent the orientation of the catenary curve. It becomes:

$$f(x) = a' \cdot \cosh\left(\frac{x+b}{a'}\right) + c,$$

with:

$$a' = -ab^2 \text{ and } c = a\cosh\left(\frac{b}{-a}\right) \text{ if } b \in [0.5; 1],$$

$$a' = -a(1+b)^2 \text{ and } c = a\cosh\left(\frac{1+b}{-a}\right) \text{ if } b \in [0; 0.5].$$

Figure 2 shows how the length parameter a influences the shape of the deflector. This parameter represents the amplitude of the shape, and it is used to define whether it will take up more or less space in the channel. It is important to note that the smaller is a, the larger is the shape.

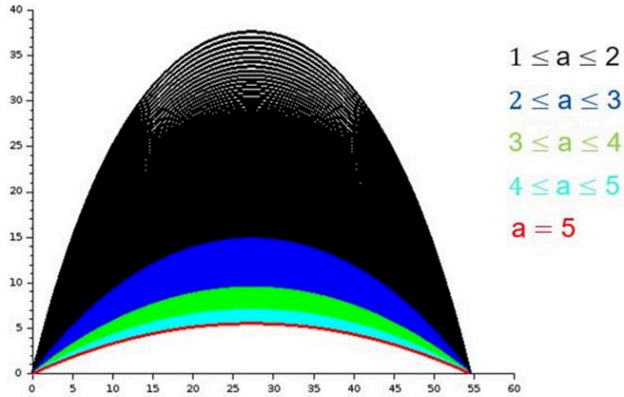

Figure 2: Illustration of the influence of the a parameter on the deflector shape (b=0.5).

The second parameter b influences the orientation of the deflector. Figure 3 illustrates its shape, depending on the value of b.

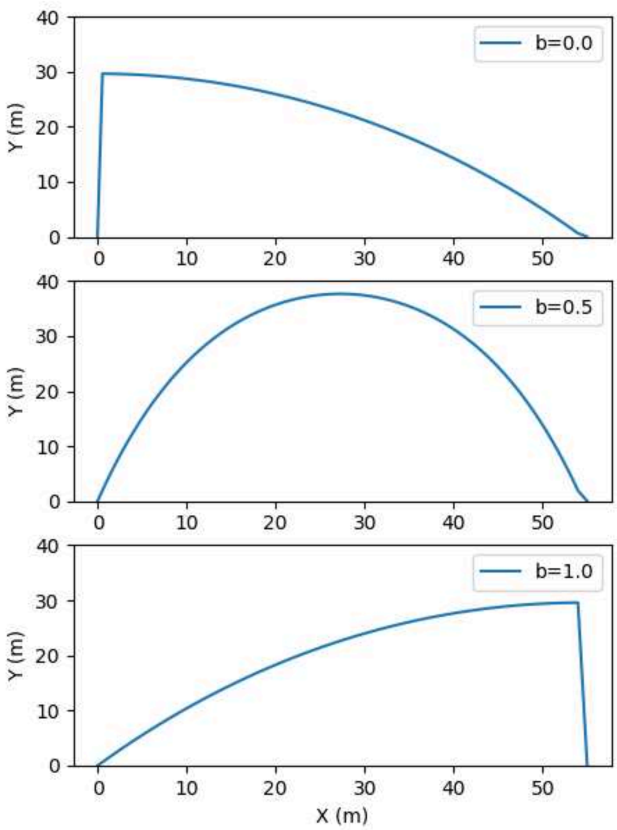

Figure 3: Illustration of the influence of the b parameter on the deflector shape (a=1).

These deflectors are inserted in the mesh contour, on the segment preceding each group of 10 pumps. Figure 4 shows their location in the domain.

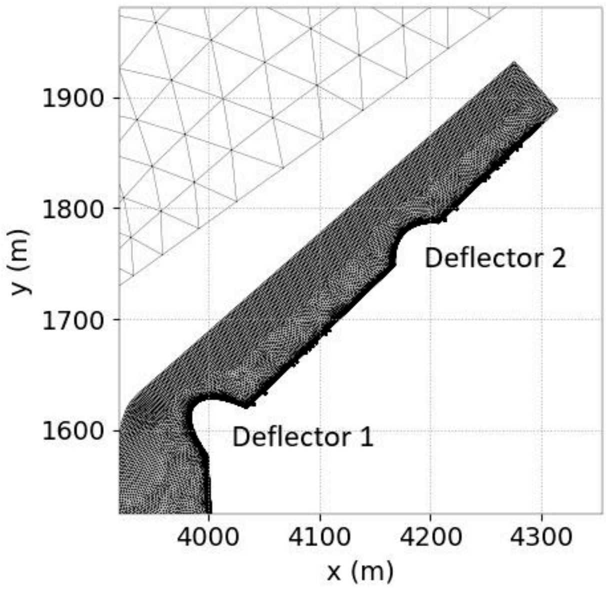

Figure 4: Location of the deflectors in the domain, here with the parameters $(a_1 = 1, b_1 = 0.5, a_2 = 2, b_2 = 0.5)$.

Finally, we have four parameters that govern the optimization problem. Thus, we have two parameters a and b and two deflectors, so the entry points of the study are written as $X = [a_1, b_1, a_2, b_2] \in [1;5] \times [0;1] \times [2:5] \times [0;1]$.

C. Cost function

In order to optimize the pumping station efficiency, the streamlines should run as perpendicular as possible to the inlet of the intakes. The aim is to reduce the α angle, shown by the black arc in Figure 5, between the normal line at the inlet and the line defined by the water velocity vectors. For every simulation, the goal is to give a single value of the residual to be able to classify the shapes of the deflectors and evaluate which one is the most appropriate for this physical issue.

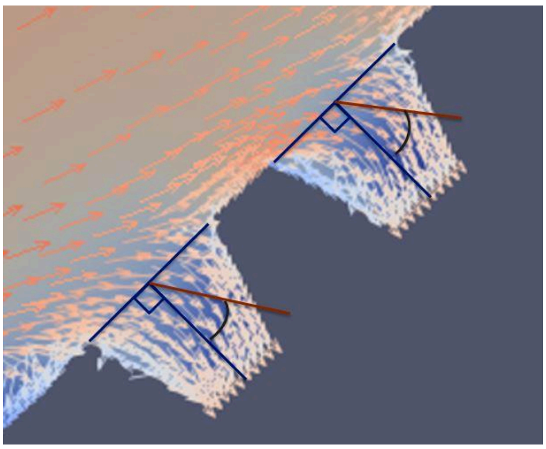

Figure 5: Schematic representation of the optimization problem.

One defines the function $\text{atan2}(y, x)$, with the formula:

$$\text{atan2}(x, y) = 2\arctan\left(\frac{y}{\sqrt{x^2 + y^2} + x}\right),$$

with $x \neq 0$. The interest of using this function lies in the fact that it allows to calculate the angle between two vectors.

With a reference system $(\vec{\imath}, \vec{\jmath})$ defined at the intake, as shown in Figure 6, one can note $\vec{V} = \binom{x}{y}, \overrightarrow{AB} = \binom{x_0}{y_0}$ and $\phi = \text{atan2}(y - y_0, x - x_0)$. We thus have ϕ the angle between $\vec{V}$ representing the velocity vector at the entrance of the intake and $\overrightarrow{AB}$, where A and B are the two points located at the intake entrance extremities.

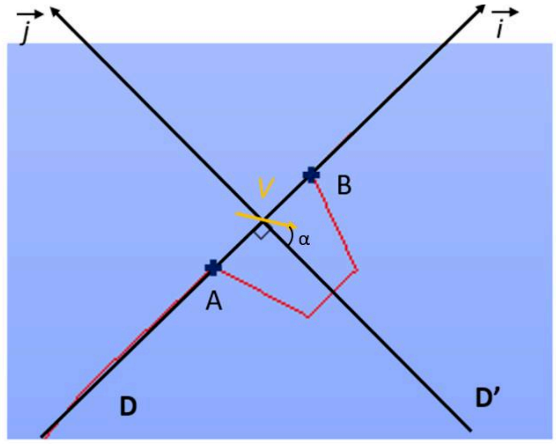

Figure 6: Illustration of the notation to define the cost function.

One then distinguishes two cases:

- If $\phi < 0$, the velocity vector enter the intake and we set $\alpha = \left||\phi| - \frac{\pi}{2}\right|$;
- If $\phi \geq 0$, the velocity vector enter the intake and we set $\alpha = \pi - \left|\phi - \frac{\pi}{2}\right|$.

We therefore have a residual angle α between 0 and π which takes into account the direction of arrival of the vector. The minimum 0 is reached when the vector enters perpendicularly into the intake. The maximum is π and it is reached when the vector exits perpendicularly from the intake.

For a fixed simulation, for a time iteration in the TELEMAC-2D computation i and an intake $j \in [1, n]$, with $n = 20$, we calculate for each of the N interpolated vector $V_{i,j}^k$ on the segment $[A, B]_j$, the angle between the vector $V_{i,j}^k$ and the line normal to the segment $[A, B]_j$. Averaging on the N interpolated vector, one have:

$$S_{i,j} = \frac{\sum_{k=1}^{N} \alpha_{i,j}^k}{N},$$

with:

- $\alpha_{i,j}^k = \|V_{i,j}^k\| \left||\phi_{i,j}^k| - \frac{\pi}{2}\right|$ si $\phi_{i,j}^k < 0$;
- $\alpha_{i,j}^k = \|V_{i,j}^k\| \left(\pi - \left|\phi_{i,j}^k - \frac{\pi}{2}\right|\right)$ si $\phi_{i,j}^k > 0$.

Averaging then over the iteration number during all the computational time, and over all the intakes, one obtain the final residual estimation:

$$Res = \frac{\sum_{j=1}^{n} \frac{\sum_{i=0}^{n_{iter}} S_{i,j}}{n_{iter}}}{n}.$$

D. Sampling and results

In order to carry out the parametric study, it was necessary to sample the shape parameters in an optimal way. Indeed, it is necessary to obtain as much information as possible on the behaviour of the model output for each point $X = [a_1, b_1] \times [a_2, b_2]$ evaluated by the TELEMAC-2D numerical model.

There are several ways to explore a given domain: divide it in a regular way (all the points in the domain are equally distributed), sampling it randomly or using quasi-random methods, known as "space filling". Given that the domain has 4 dimensions, using a regular grid would require a lot of points to evaluate in order to hope to explore the domain in an optimal way. On the other hand, the use of a random sampling method such as the Monte-Carlo method following a probability law would not allow to cover the whole space, some areas would remain unexplored. This is due to the independent nature of each point draw. There are therefore several quasi-random methods for obtaining an optimized sampling allowing exploring the domain in an optimal way without having to generate a too large sample. In our case, the LHS method (Latin Hypercube Sampling) has been used. This method is based on a uniform probability law, in our case, for each parameter. Each sample is then positioned based on the precedents so that they do not have common coordinates between them.

In order to further optimize the space coverage, 500 LHS samples are drawn are made and the one with the best space coverage is selected, minimizing the centered L_2 discrepancy [7]. The OpenTURNS module was used to sample the shape parameters. To carry out the parametric study, a sampling of 720 points was carried out.

All 720 TELEMAC-2D runs to obtain the results files were launched using the YDEFIX module of the SALOME platform. Each calculation is carried out on a processor and the duration of a calculation is between 12 and 15 hours. The YDEFIX module also allows calculating the residuals associated with each simulation, by launching both the post-processing of the results files using Postel [8] and the evaluation of the cost function. This process takes about 2 h 30 min on 36 processors with YDEFIX, compared to more than 10 times more if the results had been evaluated sequentially. Indeed, this post treatment process is very costly because it requires the extraction of the results on 20 intake entries, for 720 results file, each one containing more than 200 frames.

Once for each calculation, a residual is identified. The minimum residual the cost function is retained and the parameters of the forms that make it possible to obtain this minimum residual id extracted. The shape obtained is described by the point (a_1=2.47, b_1=0.15, a_2=2.65, b_2=0.15). Figure 7 shows the shapes with the smaller residual. Both shapes are large and oriented towards the upstream side of the channel. This causes an acceleration of the flow on the opposite side of the channel. This also creates a small recirculation after the shapes, orienting the flow in the desired direction in the two first intakes with low suction rate. The minimum residual is 0.259345 and the maximum residual is 0.5120612 obtained for the shapes with the parameter set (a_1=4.43, b_1=0.53, a_2=4.36, b_2=0.53).

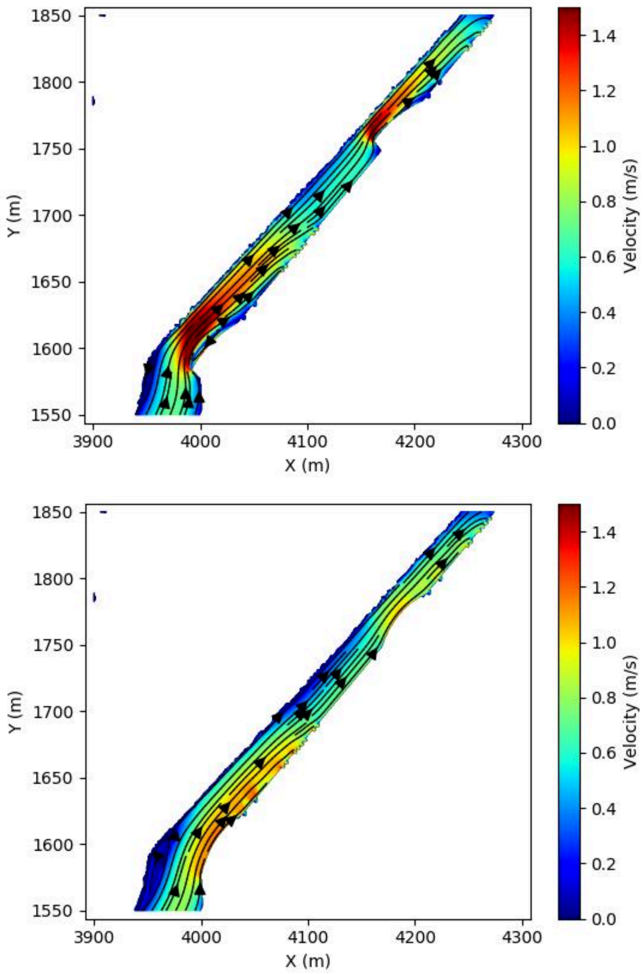

Figure 7: Shapes of the deflector minimizing (top) and maximizing (bottom) the cost function.

VII. METAMODELING AND OPTIMIZATION

A. Creation of the metamodel

A metamodel, is a mathematical function that allows replacing a very time-consuming model. It makes it possible to approximate the answers of a complex model, while having a very negligible calculation cost in comparison. In our case, a metamodel has been created to approximate the residuals of the cost function defined above. To generate a metamodel, the 720 points already evaluated by the TELEMAC-2D model are used as a learning data base.

As input for the model, one defines $X_s \in \mathbb{R}^{720\times4}$, all the points of the LHS experimental design. At the output, one defines $Y_s \in \mathbb{R}^{720}$, the set of values of the residuals of a cost function. One then can note:

$$X_s = \begin{pmatrix} X^1 \\ \vdots \\ X^{720} \end{pmatrix} = \begin{pmatrix} a_1^1 & b_1^1 & a_2^1 & b_2^1 \\ \vdots & \vdots & \vdots & \vdots \\ a_1^{720} & b_1^{720} & a_2^{720} & b_2^{720} \end{pmatrix};$$

$$Y_s = \begin{pmatrix} Y^1 \\ \vdots \\ Y^{720} \end{pmatrix} = \begin{pmatrix} G(X^1) = Res^1 \\ \vdots \\ G(X^{720}) = Res^{720} \end{pmatrix},$$

which is the learning database for the metamodel.

The metamodel interpolation has been done with the kriging method via OpenTURNS. It consists in considering the deterministic output $G(X^i) = Y^i, i \in [1, N]$ and $N = 720$ obtained using the TELEMAC-2D model as the production of the random field described as:

$$G(X^i) = \beta F(X^i) + W(X^i),$$

where βF is the regression part of the model and W the stochastic part. The regression function used is linear. Thus, it reads:

$$\beta F(x) = \beta_0 + \sum_{i=1}^{N} \beta_i x_i.$$

W is also called Gaussian process and one can write $W \sim N(0, c)$ where c is a covariance function such that $c = \sigma^2 r$ with R the correlation function. The Matérn covariance function is used. It is described as:

$$c(x, u) = \sigma^2 \frac{2^{\alpha-1}}{\Gamma(\alpha)} - \left(\sqrt{2\alpha} \left\| \frac{x-u}{\theta} \right\|_2 \right)^{\alpha} K_\alpha - \left(\sqrt{2\alpha} \left\| \frac{x-u}{\theta} \right\|_2 \right),$$

with Γ the gamma function, K_α the modified Bessel function of the second kind and α, θ and σ three strictly positive parameters respectively set to 1.5, 1 and 1.

Let x^* be a new unsampled point, to determine its value using the metamodel. The result by Gaussian kriging is given by the Best Linear Unbiased Prediction:

$$\hat{G}(x^*)_{|X_s, Y_s} \sim N\left(\mu(x^*), \hat{\sigma}^2(x^*)\right),$$

with:

$$\mu(x^*) = \beta F(x^*) + r(x^*) R_s^{-1}\left(Y_s - \beta F(X_s)\right),$$

$$\hat{\sigma}^2(x^*) = \sigma^2(1 - r(x^*)^T R_s^{-1} r(x^*),$$

where one has:

$$r(x) = (R(x_1, x), \ldots, R(x_N, x))^T,$$

$$(R_s)_{i,j} = R(x_i, x_j),$$

with:

$$R(x, u) = \frac{c(x, u)}{\sigma^2}.$$

[9] shows that, with a kriging interpolation, a sample size of a few hundred is sufficient to get a good estimation of the metamodel, even with high dimensional inputs (superior to 10). The size of the learning sample chosen here then appears to be sufficient. In this study, a convergence study showed that a sample of less than 100 inputs is enough in this case.

B. Validation

In order to assess the predictivity of the metamodel, the criterion $Q2$ is used. It requires a test database from the numerical model that was not used to create the metamodel. We therefore use 90% of the database to learn the metamodel and 10% to validate it through the $Q2$ estimation. This coefficient quantifies the part of variance of $\bar{Y}$, which is the set of output values of the test database, described by the metamodel. The criterion writes:

$$Q2(Y, \hat{Y}) = 1 - \frac{\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{N}(\bar{Y} - Y_i)^2},$$

where:

- Y_i are the observed values from the test database,
- $\hat{Y}_i$ are the values predicted by the metamodel,
- $\bar{Y}$ is the calculated average of the observed values from the test database.

The closer the Q2 coefficient is to 1, the better the fit of the model to the observations in the test database.

For the metamodel constructed with the cost function above presented, the predictivity criterion Q2 obtained is 0.859. Figure 9 shows the differences between the evaluation by the model and the metamodel on the validation sample. It can be seen that the metamodel reproduces the TELEMAC-2D model well in most cases.

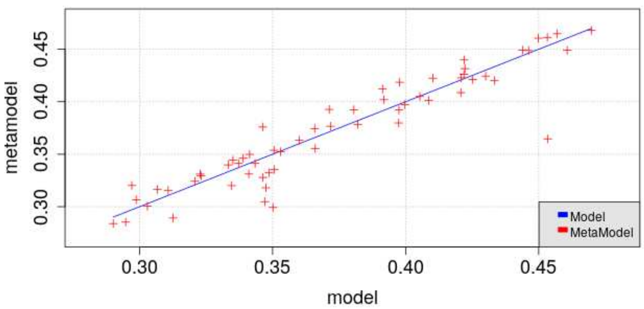

Figure 8: Validation test of the metamodel.

C. Sensitivity analysis

Sobol analysis is a sensitivity analysis technique based on variance decomposition. Given X_i^k a random vector, $i \in [1, n]$ with n the number of input parameters per evaluation, 4 in our case, and k the model dimension. Let Y^k be the output random variable of the model obtained with the input vector X_i^k. Then for a fixed k, one quantifies the dependence of the parameter Y on the input variables of the vector X_i. One considers the numerical model G such as:

$$Y^k = G\left(X_i^k\right).$$

Using the Sobol index method, one therefore tries to evaluate the part of the variance of the vector Y^k due to the different components of the vector X_i^k. To do this, the method is based on a decomposition of the variance $\mathrm{Var}[Y]$ using the Sobol-Hoeffding formula [10]:

$$\mathrm{Var}[Y] = \sum_{i=1}^{n} V_i + \sum_{i<j} V_{i,j} + \sum_{i<l<j} V_{i,j,l} + \cdots + V_{1,2,\ldots,n},$$

with $V_i = Var[\mathrm{E}[Y|X_i]]$ and $V_{i,j} = \mathrm{Var}\big[\mathbf{E}[Y|X_i]\big] - \mathrm{Var}\left[\mathbf{E}[Y|X_j]\right]$, where E represents the expected value.

One can then define the Sobol indices:

- the first-order Sobol index, denoted S_i, which quantifies the impact caused by the variable X_i on the variance $\mathrm{Var}[Y]$, independently of the interactions that X_i can exert on the other variables $X_j,\ j \neq i$. It is expressed as:

$$S_i = \frac{Var[\mathrm{E}[Y|X_i]]}{\mathrm{Var}[Y]};$$

- the total Sobol index, denoted ST_i, which defines the sum of all interactions in which variable X_i is involved. It is written:

$$ST_i = \frac{Var[\mathrm{E}[Y|X_{-i}]]}{\mathrm{Var}[Y]},$$

with X_{-i} the vector X without its i^{th} component.

The Sobol sensitivity analysis is applied to the metamodel constructed with the cost function evaluating the orientation of the flow at the intakes. 5000 evaluations of the metamodel are used to estimate the Sobol's indices. Figure 9 shows the influence of each parameter defining the shapes by displaying their first-order and total Sobol indices. Note that the most influential parameter is parameter b_1 representing the orientation of the first deflector. It can therefore be thought that this parameter is very influential in minimizing the residual of the cost function through recirculation created by this orientation, greatly influencing the direction of flow at the two first intakes. The second most important variable is the size of the first deflector, which can be explained by the fact that a shape that is too small does not generate sufficient recirculation to have a significant effect at the first intakes. Finally, it can be seen that the parameters of second deflector have little influence on the residuals. This little influence on the residuals can be explained by the fact that the second deflector is the closest to the end of the channel. The pumping here is playing a bigger role in hydraulic forcing and thus less flow is drawn to the end of the channel. The flow is therefore less likely to be parallel to the intakes and a deflector has less influence on the flow orientation.

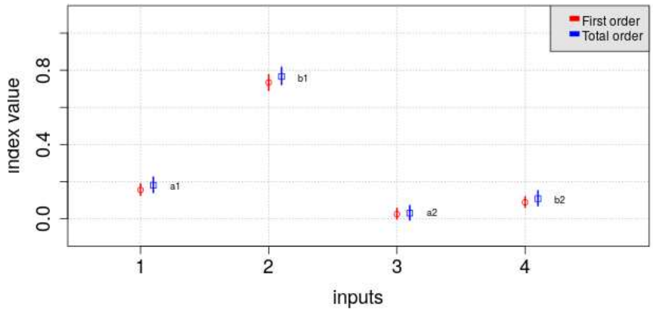

Figure 9: Sensitivity analysis of each shape parameter using Sobol indices.

D. Optimizer

The COBYLA optimizer is used via OpenTURNS [6]. This algorithm works on an incrementally principle: for each x, the optimizer evaluates $f(x)$ then, from an x_i such that $|x - x_i| < \rho, \rho \in \mathbb{R}$, calculates $f(x_i)$ and compares its value to $f(x)$. If the new value x_i gives $|f(x_i)| < |f(x)|$ then $x = x_i$ is the new starting point for the next iteration. The control distance ρ is decreased during iterations in order to evaluate the minimum of the function as accurately as possible. Optimization algorithm finishes when the control radius ρ becomes less than a predefined value ρ_{end} (fixed to 10^{-5}). In other words, the initial input of the algorithm are a starting point x_0, a control radius initial ρ_{beg} and a target radius ρ_{end}. The diagram in Figure 10 can schematize it.

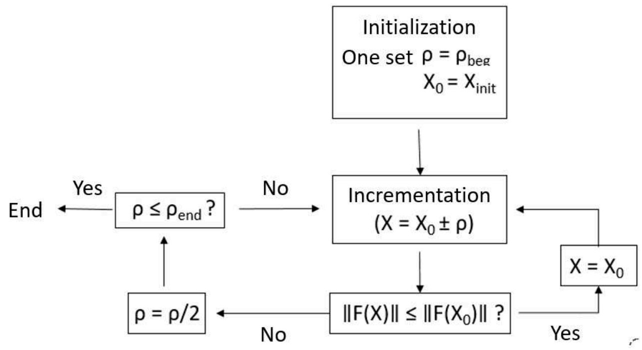

Figure 10: scheme of the COBYLA optimizer algorithm.

The main weakness of this algorithm is its inability to distinguish a local minimum from a global minimum. To overcome this disadvantage, it is necessary to make a multistart. This consists in running the algorithm several times with different starting points evenly distributed in space. The number of starting points is fixed to 500 in this study.

The optimizer has been applied to the metamodel created from the cost function. Its application gives an optimal point of (a_1=2.4, b_1=0.16, a_2=2.2, b_2=0.16). The shape found with the parametric study thus gives a lower residual than the one found with the optimizer. However, the values of the parameters obtained are close ((a_1=2.47, b_1=0.15, a_2=2.65, b_2=0.15) and (a_1=2.4, b_1=0.16, a_2=2.2, b_2=0.16)) and the residuals are low for both configurations 0.259345 and 0.292294. In particular, the parameters b_1 and a_1, which are the parameters identified as having the most influence during the sensitivity analysis are close. This seems to show that a small perturbation of the parameters of the first form leads to a small increase of the residual and therefore this could be a sign that the obtained solution is robust. Figure 11 shows that the streamlines in the channel obtained with the shape from the optimizer are quite close to those observed with the shape from the parametric study in Figure 7.

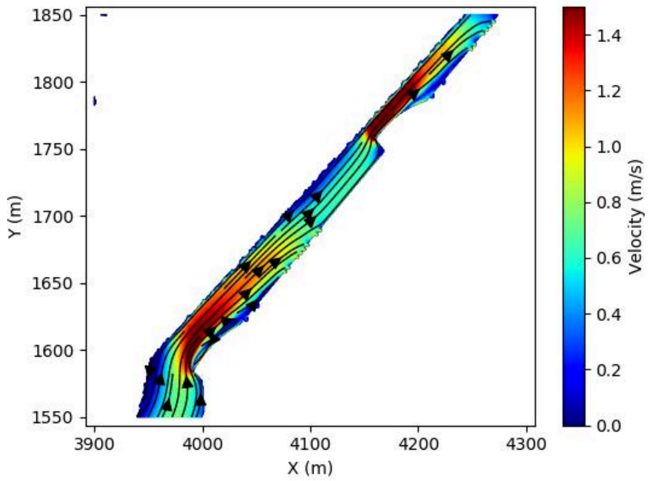

Figure 11: Velocities and streamlines for the shape obtained with the COBYLA optimizer.

VIII. Conclusion

This paper presents an optimization study of the shape of deflectors in a pumping station in order to orient the flow as well as possible in its intakes.

First, after having built the TELEMAC-2D model, the shapes of the two deflectors had to be described with two parameters each, allowing to control their size and orientation. Then, a cost function was defined to evaluate the orientation of the flow lines at the intakes entrance. Then, automatic mesh generation methods to run 720 calculations with shape parameters drawn randomly, and to calculate a residual for each run. It was then possible, to identify for which parameters the minimum residual was observed. In this case, the deflectors obtained are of medium size, with a shape oriented upstream, creating a recirculation small the first one.

Then, a metamodel was created with a kriging method, to interpolate the input parameters of the database of the 720 calculations and the output residuals. This model was evaluated by a quality criterion, Q2, to ensure that the metamodel represents the responses of the numerical model. It then allowed a sensitivity analysis to be performed on the input parameters, giving the orientation of the first deflector and, to a lesser extent, its size, as the most important parameter. An optimization of the deflector parameters was then carried out on the metamodel and the results gave parameters close to those obtained with the minimum of the parametric study. A TELEMAC-2D calculation with the parameters resulting from the optimization also gave a low residual, close to the minimum of the parametric study. The solution found during the parametric study is therefore retained, and its robustness has been demonstrated by an optimization study giving very similar results.

One perspective considered in this work is to evaluate a second cost function assessing the sedimentation potential in the channel, and to define the shapes of the deflectors by optimization on multi-objective criteria.

References

[1] TelApy User Manual, V8P1

[2] C. Goeury, T. David, R. Ata, S. Boyaval, Y. Audouin, N. Goutal, A.-L. Popelin, M. Couplet, M. Baudin, R. Barate, “Uncertainty Quantification on a real case with TELEMAC-2D”, Proceedings of the 22nd Telemac-Mascaret User Club, 2015

[3] C. Gœury, A. Ponçot, J.-P. Argaud, F. Zaoui, R. Ata, and Y. Audouin, “Optimal calibration of TELEMAC-2D models based on a data assimilation algorithm”, 24 th Telemac-Mascaret User Conference, Graz, Austria, pp. 73- 80, 2017

[4] C. Goeury, F. Zaoui, Y. Audouin, P. Prodanovic, J. Fontaine, P. Tassi, R. Ata, “Finding Good Solutions to Telemac Optimization Problems with a Metaheuristic”, XXV th Telemac & Mascaret User Club, 2018

[5] Amsallem, D., Zahr, M., Choi, Y., & Farhat, C. (2015). Design optimization using hyper-reduced-order models. Structural and Multidisciplinary Optimization, 51(4), 919-940.

[6] OpenTURNS documentation, http://openturns.github.io/openturns/latest/index.html

[7] Fang, K. T., Ma, C. X., & Winker, P. (2002). Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs. Mathematics of Computation, 71(237), 275-296.

[8] Audouin Y., Fontaine J., Fouquet T., Goeury C., Leroy A., Pham C.-T., Souillé F., Taccone F., Duron, L., Daou, M.-P. and Sécher, M. “A new Python3 module for TELEMAC-MASCARET dedicated to post-treatment: Postel” XXVI th Telemac & Mascaret User Club, 2019

[9] Maurel, A., Iooss, B., Van Dorpe, F., & Volkova, E. (2008). An efficient methodology for modeling complex computer codes with Gaussian processes. Computational Statistics & Data Analysis, 52(10), 4731-4744.

[10] Olivier Roustant. Sobol-Hoeffding decomposition: bounds and extremes. 2018. hal-01953080