Benefits of a metamodel for automatic calibration of 1D and 2D fluvial models

Rui YANG[1,2]
[1] Ecole Centrale de Lyon
36 Avenue Guy de Collongue, 69134 Écully, France

Luc DURON[2], Guillaume BENEFICE[2], Pierre NUNES[2]
[2] Compagnie Nationale du Rhône
2 rue André Bonin, 69009 Lyon, France
l.duron@cnr.tm.fr

***Abstract*— Construction of hydraulics models are often time-consuming in terms of calibration, which is an essential step. In this paper, two methods (kriging and XGBoost) are used to build intermediate metamodels for 1-D and 2-D hydraulic models to calibrate friction coefficients defined by zones. The resulting metamodels were hereafter applied for sensitivity analysis and other applications. The benefits of an intermediate metamodel were assessed on multiple real applications and this method was compared to the other semi-automatic calibration method (meta-heuristic method, see [12]). The developed approach involving a metamodel appears to be highly computationally less intensive and leads to similar results.**

I. INTRODUCTION

The calibration of a hydraulic model is an essential phase, which consists typically in adjusting friction coefficients. The common Strickler coefficients are indeed often hard to estimate, because they depend on the materials of the riverbed, the shape of the riverbed, etc. and they may account for different terms of energy losses depending of the flow rate. That's why it is necessary to calibrate it to minimize the error between simulations and observations. Usually for channel with simple cross section geometry, if the error of water level between hydraulic model and observations is around a few centimetres and the error of discharge is 5 to 10% of the nominal discharge, the hydraulic model is considered as correctly calibrated. But if the error is too large at some locations, it is possible that the model parameters are wrong, the observations are wrong, or it is necessary to add new features to the hydraulic model [13]. In this paper, it is assumed that observations and discharges are correct. Moreover, it only focuses on 1D and 2D steady flow models.

The classic calibration approach needs to repeat hydraulic calculations which is a time-consuming process. An interesting method is to build an intermediate and abstract metamodel and apply it to multiple purposes, as presented by Zaoui et al. (see [1]). This methodology is presented in detail in the next section.

This automatic calibration was applied firstly on a 1D simple test case with multiple design of experiments (DOE) methods and metamodels, as presented in section IV.A. Once the method was validated, it has been applied and compared to a more complex model in 1D and 2D. The computations have been performed on a 1D computation model named Crue10 (property of CNR) and a TELEMAC-2D model. Moreover, some sensitivity analysis have been conducted in order to consolidate the calibrated parameters and give some complementary information for the modeler.

The developed scripts are done with Python and are based on the scikit-learn module. They consist of a set of functions which the modeler can combined.

II. METHODOLOGY

This paper proposes to use an intermediate metamodel to realize the calibration. Figure. 1 shows the general workflow for the automatic calibration.

Firstly, the input space is sampled to create a matrix called design of experiments (DOE). There are several DOE methods (LHS, PhiP, MinDist and C2), which are described hereafter. Then hydraulic computations are performed for each sample. Input variables are Strickler coefficients for different zones of a river and output variables are water levels at different sections and discharges. Intervals of Stricker coefficient are chosen by experience without further information, based on the principle of maximum entropy the uniform law is used [14].

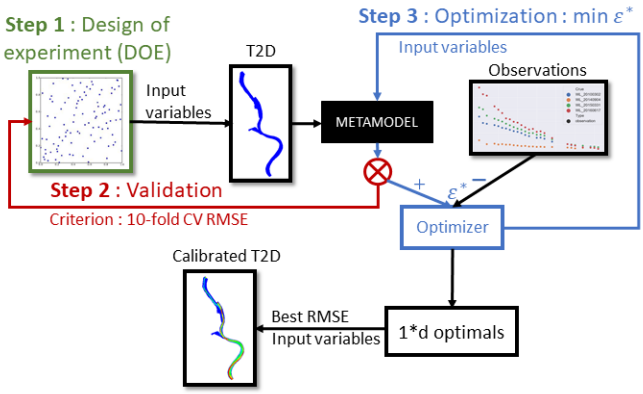


Figure 1. Workflow of the automatic calibration

The second step is the construction of the metamodel and its validation. For the metamodel definition, the DOE and the output variables are used, and multiple methods can be applied

(e.g. kriging, XGBoost etc.). The validation process consists in a cross-validation to define the accuracy of the metamodel If the error criteria is too high it is recommended to increase the size of the DOE or adjust the hyperparameters of the metamodel.

Finally, the third step is the optimization step, which is based on a minimization approach of root-mean-square error between metamodel predictions and observations. Water level elevation measurements are realized at different stations and one discharge corresponds to one water line.

Because of the equifinality problem and the use of a meta-heuristic method, this process is repeated 1*d (d is the number of input variables) times, which lead to the best configuration of input variables. We observed that d repetitions is a good compromise and leads in practice to around 10-20 repetitions.

A. Design of experiment (DOE)

The definition of the DOE is an essential step, because it determines calculation costs. There are mainly three types of DOE methods [3]:

- Classical DOE (factorial, composite, etc.),
- Law-discrepancy sequences,
- Latin Hypercube Design (LHS).

When using a non-parametric metamodel (the number of parameters can grow with the sample size), there are only two criteria to assess a DOE [3]:

- No redundancy by projection,
- Space filling designs.

In our study cases, the hydraulic model often depends on more than ten Strickler coefficients. But the water level measured at a station only depends on a few of them. That's why we expect the experiment points won't be confused when we make the projection onto the subspace constructed by a few of Strickler coefficients. That's also the reason that the property of no redundancy by projection is important [4]. Otherwise, if we assume that we don't have any prior knowledge about the response surface, the only thing we can do is to explore the input space. It corresponds to the space filling designs [3].

For engineering applications, the classical DOE is not convenient because the size of the DOE will be huge when the number of input variables is large (e.g. the factorial design needs 2^d points where d is the number of inputs) [4]. For the Law-discrepancy sequences, when the number of input variables is large, it performs badly on the projection [6]. The Latin Hypercube Design (LHS) can guarantee the property of no redundancy but it is often poor on the space filling [4]. That's why we need LHS and space filling designs. The objective is to generate a DOE created by LHS method X_d^N ($X_d^N = (x_j^{(i)})_{(i=1\dots N, j=1\dots d)}$ where N is the size of DOE, d is the number of input variables and $x_j^{(i)}$ represents the j-th input of the i-th experiment), that minimizes a space filling criterion φ. For example, the space filling criterion φ can be [7]:

- The minimum average distance between 2 elements according to L2 norm:

$$\phi(X) = \min|x^{(i)} - x^{(j)}|_{L^2, \forall i \neq j=1,\cdots N} \quad (1)$$

- PhiP: The p-moment of the xi distribution

$$\phi_p(X) = \left(\sum_{1\leq i<j\leq N}|x^{(i)} - x^{(j)}|_{L^2}^{-p}\right)^{\frac{1}{p}}, p = 50 \text{ by default} \quad (2)$$

- C2: The centred L2-discrepancy

$$C^2(X_d^N) = \left(\frac{13}{12}\right)^d - \frac{2}{N}\sum_{i=1}^N\prod_{k=1}^d\left(1+\frac{1}{2}\left|x_k^{(i)}-0.5\right|-\frac{1}{2}\left|x_k^{(i)}-0.5\right|^2\right) + \frac{2}{N^2}\sum_{i,j=1}^N\prod_{k=1}^d\left(1+\frac{1}{2}\left|x_k^{(i)}-0.5\right|+\frac{1}{2}\left|x_k^{(j)}-0.5\right|-\frac{1}{2}\left|x_k^{(i)}-x_k^{(j)}\right|\right) \quad (3)$$

After sampling, we can have a sample matrix:

$$K = \begin{pmatrix} K_{s11} & \cdots & K_{s1d} \\ \vdots & \ddots & \vdots \\ K_{sN1} & \cdots & K_{sNd} \end{pmatrix}_{N\times \mathrm{d}} \quad (4)$$

K_{sij} represents the j-th Strickler coefficient of the i-th sample.

Then we pass it to the hydraulic model to get a response matrix with n stations and m water lines:

$$Z = \begin{pmatrix} Z_{11} & \cdots & Z_{1(n\times m)} \\ \vdots & \ddots & \vdots \\ Z_{N1} & \cdots & Z_{N(n\times m)} \end{pmatrix}_{N\times(n\times m)} \quad (5)$$

Z_{ik} represents the k-th water level of i-th sample.

A water line is a set of observed water elevations at different stations for the same boundary conditions (assumed to be steady flow).

B. Metamodel

Understanding of the dataset structure is very useful for metamodeling. The properties of our dataset:

- Tens input variables that are continue and assumed to be non-correlated and independent,
- Hundreds of output variables that are continue and correlated,
- The size of dataset is ten times the number of input variables.

In terms of data science, it's a typical regression problem with small dataset.

It is not recommended to use parametric model (e.g. linear regression, polynomial, exponential etc.) because these models are often too simple to recognize.

In the field of mechanics, we use often kriging which is a powerful method for metamodeling when the size of dataset is not too large. In fact, the prediction of kriging interpolates the real output and is probabilistic so that one can compute empirical confidence intervals. But they lose efficiency in high dimensional spaces (when the number of input variables exceeds a few dozens) and they use whole samples information to perform the prediction [8].

Generally, the only thing we need to define is the kernel matrix which describes the covariance of the Gaussian process (i.e. kriging). In our case, the output (water level) comes from a hydraulic model, therefore the response surface is smooth (i.e. not noisy). Additionally, it is not periodic. The common choice for the kernel matrix is Matérn 5/2 [2].

Another method is XGBoost (Extreme Gradient Boosting). It's very popular and powerful in the field of machine learning [9]. XGBoost method is based on the decision tree ensembles. The tree ensemble model consists of a set of classification and regression trees (CART). Usually, a single tree is not enough to make a prediction. That's why an ensemble model which calculates weighted average of multiple trees is used [9]. It has been proved very powerful over the Kaggle competition [9]. But there exists a lot of hyperparameters for the training process. In practice, we need to tune the XGBoost to fit better our dataset. Based on our study cases, a fine-tuned XGBoost metamodel can perform at least as well as kriging.

In fact, we do not make a complete model for all outputs but one single model for one output and then combine all of them. Thus, we used univariate metamodels because:

- The sample size is not large, and the number of outputs variables is acceptable.
- We don't know exactly the correlation among the outputs. If we create one complete model this can't help us to increase the quality of model [10].

Of course, we can make some transformations over outputs to decrease the number of outputs. For example, we can calculate the RMSE error between metamodel predictions and observations [1]. But some information will be lost. It's better to make a single model for each output.

$$M(K_{s1},\cdots,K_{sd}) = \begin{cases} M_1(K_{s1},\cdots,K_{sd}) \\ \vdots \\ M_{n\times m}(K_{s1},\cdots,K_{sd}) \end{cases} \quad (6)$$

where:

- M_1 is the metamodel for the first output;
- K_{s1} is the first input variable.

C. Validation

In the field of machine learning, k-fold cross-validation is commonly used to estimate the performance of machine learning models. Because it doesn't take any additional data and generally results in a less biased or less optimistic estimate of the model skill than other methods [8].

Usually, if the size of dataset is less than 100, we take $k = 20$, else we take $k = 10$.

In the hydraulic field, the criteria below are often used:

Mean signed error (MSE): $\frac{\sum_{i=1}^{N}\sum_{j=1}^{n\times m}(M_j(K_{si1},\cdots,K_{sid})-Z_{ij})}{N\times n\times m}$ (7)

Mean absolute error (MAE): $\frac{\sum_{i=1}^{N}\sum_{j=1}^{n\times m}|M_j(K_{si1},\cdots,K_{sid})-Z_{ij}|}{N\times n\times m}$ (8)

Root-mean-square error (RMSE): $\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{n\times m}(M_j(K_{si1},\cdots,K_{sid})-Z_{ij})^2}{N\times n\times m}}$ (9)

Where:

- M is the metamodel
- Z is the value of the initial hydraulic model
- K is the Strickler coefficient

D. Optimization

In this part, we compare metamodel predictions with observations and minimize the error between them. The cost function is a RMSE:

$$\sqrt{\frac{\sum_{j=1}^{n\times m}(M_j(K_{s1},\cdots,K_{sd})-O_j)^2}{n\times m}} \quad (10)$$

O is the corresponding observation.

We use a common optimization algorithm named particle swarm optimization (PSO). it belongs to the metaheuristic optimization methods. It can converge fast, has short computational time and has high probability and efficiency in finding the global optimum [11].

III. HYDRAULIC MODELS AND STUDY AREAS

There are totally three study cases or hydraulic models: Chautagne (CE) 1D, Bourg-lès-Valance (BV) 1D and BV 2D. The 3 models are in two different parts of the Rhône river: in the upper Rhône for Chautagne and lower Rhône for Bourg-lès-Valence (lower bottom slope, larger cross-sections).

A 1D computation code named Crue10 (property of CNR) is used for the sake of methodology investigations, as it computes fast. But the main test cases are in the region of Bourg-lès-Valence and involves 2 different computations codes: Crue and TELEMAC-2D. The 2 hydraulic models are using a different discretization of shallow water equations. They can both be used to calculate the water level at different sections under different boundary conditions.

In our study cases, we use minor-river bed and major-river bed Strickler coefficient over different parts of the hydraulic models as input variables. The models do not include any storage area.

A. *Chautagne (1D) [CE]*

Chautagne's model covers about 59 km^2 around a run-of-river dam. The hydraulic model (1D) used in this paper contains hundred sections and covers 7 km long. 2 water level profiles correspond to different boundary conditions (see discharge values in Table IV), the water level is measured at 15 locations for each water line. There are 12 input variables (6 minor-river bed and 6 major-river bed Strickler coefficient) and 30 (15*2) output variables.

TABLE I. CHAUTAGNE MODEL MAIN CHARACTERISTICS

Number of sections	100
River length	6.7 km
Number of water lines	2

B. *Bourg-lès-Valence (1D and 2D) [BV]*

The model of BV is located along the Rhône river, upstream the dam of La Roche-de-Glun. It covers nearly 16 km and has been separated into multiples zones, as presented on figure 2. The 2 hydraulic models (Crue and TELEMAC-2D) are covering the same domain and have a similar bathymetry.

4 water lines are available and the water level are measured at 32 locations. Overall, there are 14 input variables and 128 (32*4) output variables.

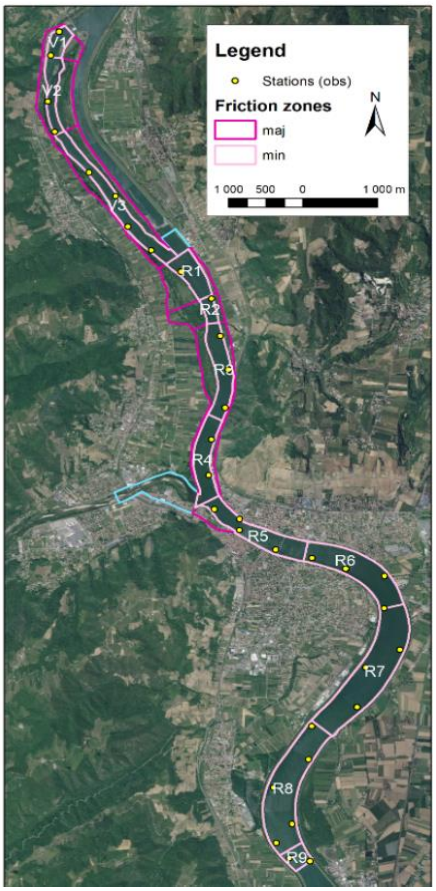


Figure 2. Overview of BV domain and zoning

TABLE II. BOURG-LÈS-VALENCE MODELS MAIN CHARACTERISTICS

	1D	2D
Number of sections/nodes	248	32 000
River length	15.6 km	
Number of water lines	4	

IV. APPLICATIONS

Normally, XGBoost parameters need to be fine-tuned for each case. But in practice, XGBoost method is compared with kriging. If the validation RMSE of XGBoost method is much better than kriging method (about a half), there is no need to fine-tune XGBoost parameters. When we deal with similar problems (response surfaces are similar) or data structures, the hyperparameters do not need to be adapted.

And in our study cases, the hyperparameters of XGBoost are fine tuned in the BV 1D model, and they work for two another cases. Therefore, the same hyperparameters of XGBoost were used for all subsequent tests.

A. *Chautagne 1D model*

(1) *Benchmark of DOE – validation step*

For the Chautagne 1D, 10 DOEs were repeated for each size and each method (4 sizes * 4 methods = 16 combinations). The size tested are (3**d*, 5**d*, 7**d*, 10**d*) where *d* (*d*=12) is the number of input variables. There are four methods in total: non-optimized LHS, LHS + Space filling C2, LHS + MinDist, LHS + PhiP. Finally, two metamodel methods are compared: kriging and XGBoost, with a 10-fold cross-validation RMSE for all DOEs (160 in total). This benchmark corresponds to a convergence study.

Figure. 3 is a box plot and each box consist of 10 DOEs.

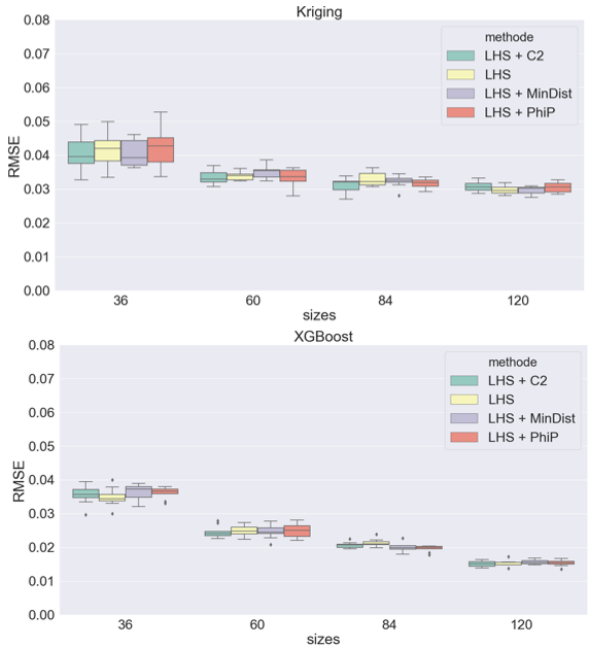


Figure 3. The 10-fold CV RMSE of XGBoost and Kriging in different designs of experiment with different sizes and methods

We find:

- As the *N* (size of DOE) increases, both RMSE and variance (the height of box) are decreasing.
- When the size of DOE reaches 10**d*, there is almost no difference between the four methods.
- XGBoost method is slightly better than kriging with lower RMSE and lower variance. Additionally, the RMSE of XGBoost method decrease more rapidly than using kriging with *N* increasing.

In this study case, the relationship between one output and inputs is not too complicated. Generally, one output is only related to three or four inputs and the relation is like a combination of second or third order polynomials. A DOE with size 10**d* is enough.

There is no obvious difference between the four methods. So, we prefer to use LHS optimized by MinDist which is very simple and can optimize the property of space filling.

Because XGBoost method is not guaranteed that it is always performing better than kriging and it takes only a few seconds to construct a metamodel, both metamodels are used. Then the recommendation is to use directly the one giving the best RMSE.

(2) Benchmark of metamodels for optimization

For the optimization process a metaheuristic method, called PSO, is used. Hence the process is not reproducible and the solution not fixed. In this comparison, the PSO process was applied 1**d* (*d*=12 input variables) times for each metamodel. The optimization seeks to minimize the RMSE between hydraulic model and observations. The DOE in this part has a size of 10**d* created by LHS+MinDist.

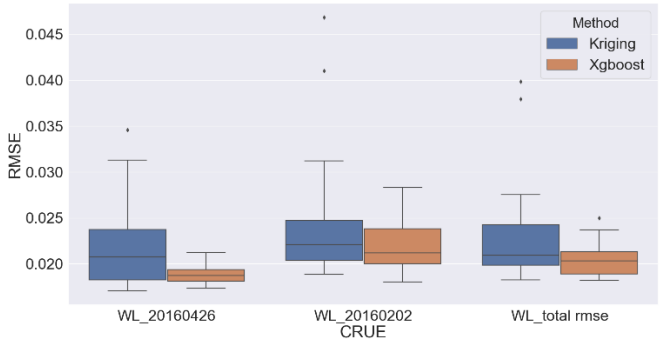


Figure 4. RMSE between hydraulic model and observations

Figure 4 presents box plots and each box contain 1**d* experiments. It shows the RMSE between hydraulic model simulations and observations for two water lines and the total RMSE of them.

In terms of optimization, XGBoost method is slightly better than kriging method, with lower variance and lower average of total RMSE. For kriging method, the average of total RMSE is 0.023 m and the standard deviation is 0.006 m. For XGBoost method, the average is 0.021 m and the standard deviation is 0.002 m.

The minimum RMSE for kriging and XGBoost methods are very close, values are respectively 0.019 m and 0.018 m. Therefore, their performances are similar in terms of calibration but results with the XGBoost method have a better repeatability.

Based on the result of optimization step, we prefer to do 1**d* times optimization processes by using XGBoost method and choose the best RMSE. This configuration is investigated in depth in the following part.

(3) Results for the best configuration

The selected configuration is based on:

- Metamodel: XGBoost method,
- Size of DOE: 10**d*,
- Type of DOE: LHS + MinDist.

The validation criteria are presented in Table III. The metamodel is not biased (MSE close to 0) and has an absolute error (MAE) less than 1cm. Furthermore, the total validation RMSE (0.014m) is close to the optimisation RMSE (0.018m).

TABLE III. VALIDATION STEP FOR CE 1D

	RMSE	MAE	MSE
Value (m)	0.014	0.009	-0.0005

Fig. 5 plots the water level of hydraulic model, metamodel prediction and observation at different sections (axis label "sect_id" corresponds to the section number) and different water lines. We get 0.019 m RMSE for water line (1), 0.018 m RMSE for water line (2). This result is basically equivalent to the traditional calibration method.

Fig. 6 plots the value of input variables (Strickler coefficients) for the optimal point, the upper and lower boundary for each input. Here, *Maj* means major-river bed, *Min* means minor-river bed and *R1* corresponds to a friction zone. The optimal values correspond to the typical values that the modeller could reach with the traditional method.

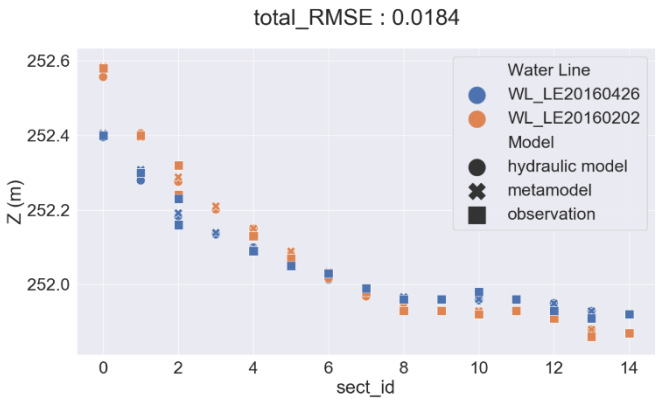


Figure 5. Water lines for Chautagne 1D

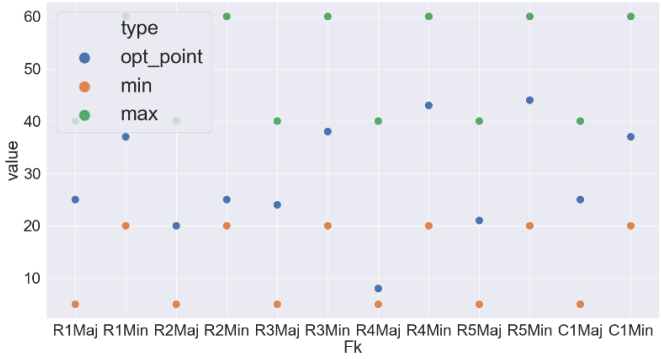


Figure 6. Input values (Strickler coefficients) range (min/max) and optimal configuration (opt_point) for Chautagne

TABLE IV. WATER LINES REFERENCES

Waterline	2016-02-02 (1)	2016-04-26 (2)
Discharge (m^3/s)	900	646
RMSE Optimization (m)	0.0186	0.0182

(4) Sensitivity Analysis

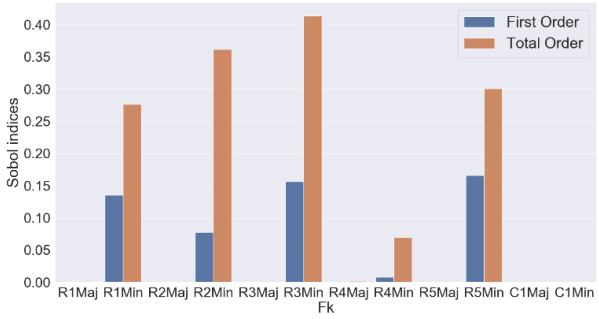


Figure 7. Sobol' indices (total and first order) of RMSE between metamodel predictions and observations for Chautagne.

Fig. 7 shows only 5 parameters are significant and they are all Strickler coefficients located in the minor riverbed. This observation results of the selected events since discharges are not high enough to flood the major bed.

B. Bourg-lès-Valence 1D model

The methodology defined on Chautagne test case has been conducted on the more complex model, named BV. In a similar manner, a DOE of size 10*d (d=14 inputs variables) created by LHS+MinDist is used.

(1) Validation

In the validation step, we create two metamodel by using respectively kriging and XGBoost methods.

TABLE V. KRIGING'S VALIDATION STEP FOR BV 1D

	RMSE (m)	MAE (m)	MSE (m)
Kriging	0.025	0.014	-0.0002
XGBoost	0.022	0.013	-0.0002

For the BV 1D, in the validation step, a fine-tuned XGBoost method gives a 0.022 m RMSE that is still better than 0.025 m given by kriging method. This result corresponds to what was obtained with Chautagne 1D.

(2) Optimization

Like the study case Chautagne 1D, we use PSO method to find 1*d (d=14 inputs variables) optimal Strickler coefficient by minimizing the RMSE between hydraulic model and observations. The value of the minimum RMSE was used as criterion to compare the performance of the two metamodels.

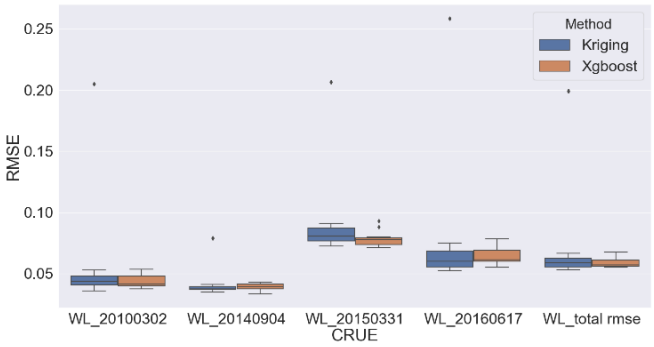


Figure 8. Optimization step for BV 1D

In terms of optimization, XGBoost method is better than kriging method, with lower variance and lower average of total RMSE. For kriging method, the average of total RMSE is 0.079 m and the standard deviation is 0.040 m. For XGBoost method, the average is 0.059 m and the standard deviation is 0.004 m.

The minimum RMSE of kriging is 0.053 m and the minimum RMSE of XGBoost method is 0.055 m. The RMSE of optimization step is about twice as much as the RMSE of validation step.

BV 1D model gives similar results compared with CE 1D model. In the validation step, XGBoost method is always better than kriging method. In the optimization step, their performances are similar for calibration and XGBoost method has better repeatability.

C. Bourg-lès-Valence 2D model

As for BV 1D model, a DOE of size 10*d (d=14 inputs variables) created by LHS+MinDist is used in this part.

(1) Validation

In the validation step, we create two metamodels by using respectively kriging and XGBoost methods.

TABLE VI. KRIGING'S VALIDATION STEP FOR BV 2D

	RMSE (m)	MAE (m)	MSE (m)
Kriging	0.115	0.071	-0.0008
XGBoost	0.066	0.040	0.0002

For the BV 2D model, in the validation step, a fine-tuned XGBoost method gives a 0.066 m RMSE which is much better than 0.115 m given by Kriging method. We get a RMSE much larger than Chautagne 1D and BV 1D models. Nevertheless,

R-square in the three cases are very close. Therefore, the variance in the water level in BV 2D model is larger than in the two another cases.

(2) Optimization

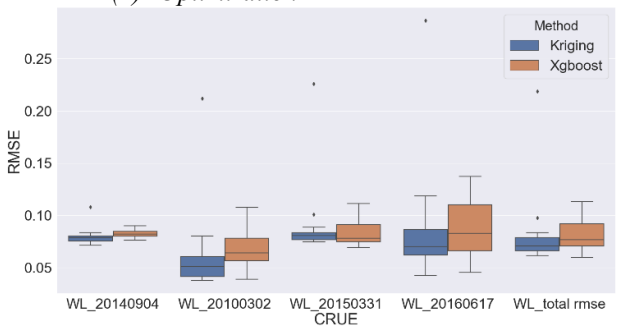


Figure 9. Optimization step for BV 2D

In terms of optimization, for Kriging method, the average of total RMSE is 0.083 m and the standard deviation is 0.040 m. For XGBoost method, the average is 0.082 m and the standard deviation is 0.016 m. The minimum RMSE of Kriging method is 0.061 m and the minimum RMSE of XGBoost method is 0.060 m.

We find all these three study cases give a similar result. XGBoost method is always better than Kriging method in the validation step. In the optimization step, XGBoost method is more stable (lower variance), especially in 1D. But the minimum RMSE of both methods is very close.

So, only one optimal point is considered, XGBoost method is better. But if 1*d optimal points are computed and the minimum of them used, the performance of both methods is close.

D. BV 1D vs BV 2D models

All results are based on the following configuration:

- Metamodel: XGBoost method,
- Size of DOE: 10*d,
- Type of DOE: LHS + MinDist,
- 10-fold cross-validation for validation step.

(1) Water lines

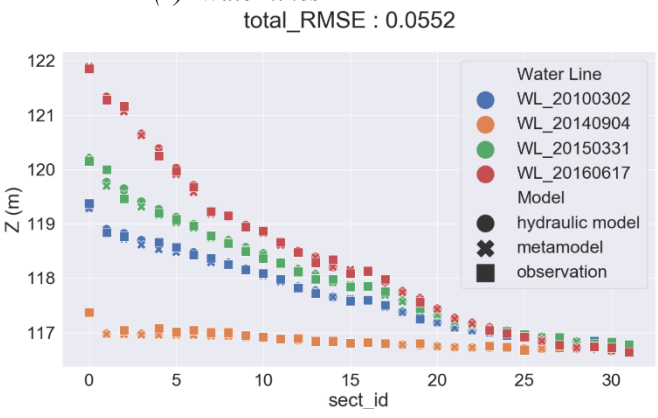


Figure 10. Water lines of BV 1D

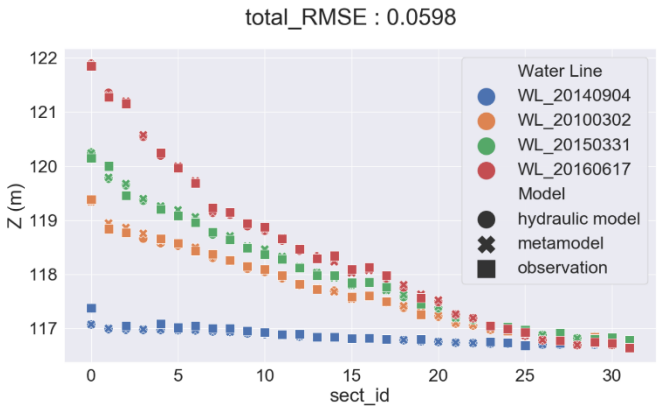


Figure 11. Water lines of BV 2D

In this section, we use the minimum RMSE of 1* d optimal points for each model (either 1D or 2D). In both cases, the Strickler coefficients used in the model are those calibrated with the metamodel built thanks to XGBoost method with the method LHS + MinDist. For BV 1D model, the RMSE between observations and simulations is 0.055 m. For BV 2D model, the RMSE is 0.060m. We find also that the points with large error are concentrated in the upper part (section id 0,1,2,4).

(2) Strickler coefficients

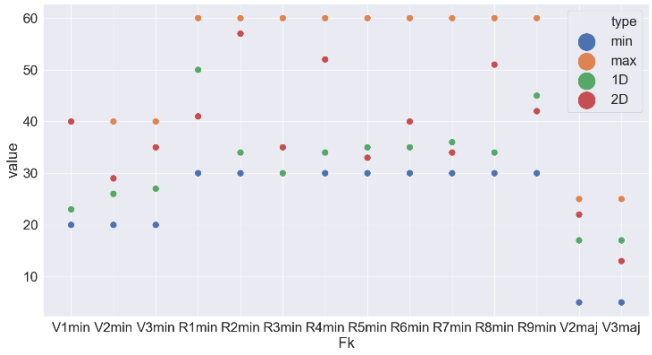


Figure 12. Strickler coefficients: BV 1D vs BV 2D

Figure 12 shows the value of Strickler coefficient for BV 1D and BV 2D models at the optimal point. *Min* and *Max* correspond to the domain of Strickler coefficients. Strickler coefficients are overall higher in 2D than in 1D, which is what is expected. In the details, the Strickler coefficients for 1D and 2D in V1min, R2min, R4min and R8min are much different. Whereas the values are closer for the other.

(3) Sobol' indices

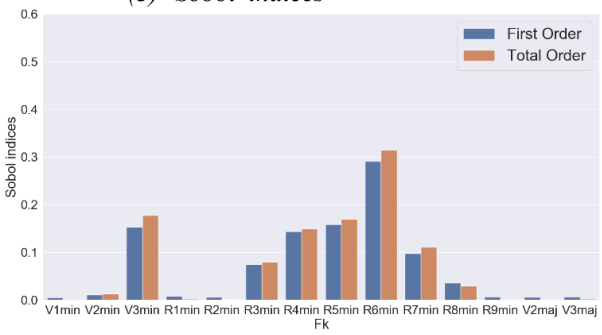


Figure 13. Sobol' indices (total and first order) of BV 1D

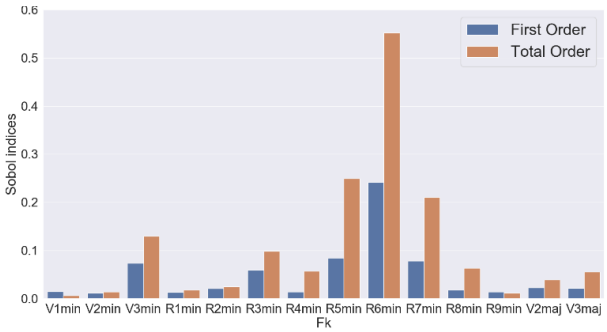


Figure 14. Sobol' indices (total and first order) of BV 2D

Figure 13 shows the Sobol' indices of BV 1D model, the first order is close to the total order. It means there is no much interaction effects between inputs. These inputs mainly affect outputs independently. But in BV 2D model, the total order is larger than first order. It means the interaction effects are much more important. The outlines of 1D and 2D are basically similar, except that V3min and R4min in the BV 1D are more important. In the Figure 12, the K-value in V1min, R1min, R2min and R8min is much different between 1D and 2D models. And we also find that in the Sobol' indices of these Strickler coefficients have no effect to the output neither in 1D model nor in 2D model. So, the value of these coefficients is not important. This also the case for V2min, R9min and major-river bed zones. But for 1D model, values of Strickler coefficient in parts V3min and R4min play a significant role in the outputs. Therefore, the dispersion around the mean K-value is expected to be lower in those groups of sections than in other parts of the river. But for 2D model, the role played by those two coefficients is not so important. So, in the process of optimization, the value of these two coefficients will be more random and the discrepancy is expected to be larger.

V. CONCLUSION

This paper shows it's relevant to build an intermediate metamodel and it has been applied to various test cases (1D and 2D). Based on three study cases, in the validation step, XGBoost method always performs better than kriging method, with lower RMSE.

In the optimization step, because of the use of a meta-heuristic and the equifinality problem, it is recommended to repeat the optimization $1 * d$ times and choose the best solution. We find the best RMSE of XGBoost method is close to the best RMSE of kriging method in this step.

Finally, when we apply this approach to other study cases, we might need to adjust the hyperparameters of XGBoost method. This choice of the hyperparameters should be investigated and automated.

The perspectives include some more advanced applications on the calibrated metamodel, such as uncertainty quantification. In addition, when the hydraulic model or the uncertainty definition slightly changes, updating old metamodel, without reconstructing a new one, is also an interesting problem.

REFERENCES

[1] Fabrice Zaoui, Cédric Goeury and and Yoann Audouin. "A Metamodel of theTelemac Errors", XXVIth TELEMAC-MASCARET User Conference, October 2019, Toulouse.

[2] Mohammed Reza Kianifar and Felician Campean. "Performance evaluation of metamodelling methods for engineering problems: towards a practitioner guide", Structural and Multidisciplinary Optimization, July 2019.

[3] Mona Abtini. "Plans prédictifs à taille fixe et séquentiels pour le krigeage". Thèse de doct. August 2018.

[4] Jessica Franco. "Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes". Thèse de doct. 2008.

[5] Loic Le Gratietet and Claire Cannamela. "Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes." 2012. arXiv :1210.6187 [math.ST].

[6] Razi Sheikholeslamiet Saman Razavi. "Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environ-mental models". In: Environmental Modelling Software93 (2017), p. 109-126.issn: 1364-8152.

[7] Airbus-EDF-IMACS-Phimeca, "Reference Guide - OpenTurns 1.15," 2020.

[8] F. Pedregosaet al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), p. 2825-2830.

[9] Carlos Guestrin Tianqi Chen. "XGBoost: A Scalable Tree Boosting System". Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).

[10] Jack P.C. Kleijnen, Ehsan Mehdad "Multivariate versus univariate Kriging metamodels for multi-response simulation models" Volume 236, Issue 2, 2014, Pages 573-582.

[11] Abdmouleh, Zeineb & Gastli, Adel & Ben-Brahim, L. & Haouari, Mohamed & Al-Emadi, Nasser. (2017). "Review of optimization techniques applied for the integration of distributed generation from renewable energy sources." Renewable Energy.

[12] Meta-heuristic Optimization Method for the Calibration of Friction Coefficients in 1-D Open Surface Channel Modeling, Pierre-Loïk Rothé et al., Advances in Hydroinformatics, 2018

[13] P.O. Malaterre, J.P. Baume, N. Jean-Baptiste, J. Sau. Calibration of open channel flow models: a system analysis and control engineering approach. SimHydro 2010: Hydraulic modeling and uncertainty, Jun 2010, Sophia-Antipolis, France. 10 p., 2010. <hal-00783013>

[14] Mouradi, Rem-Sophia & Audouin, Yoann & Goeury, Cédric & Claude, Nicolas & Tassi, Pablo & el Kadi Abderrezzak, Kamal. (2016). Sensitivity analysis and uncertainty quantification in 2D morphodynamic models using a newly implemented API for TELEMAC2D/SISYPHE.