MULTI-FIDELITY SURROGATE MODELING FOR TIME-SERIES OUTPUTS

Baptiste Kerleguer

MULTI-FIDELITY SURROGATE MODELING FOR TIME-SERIES OUTPUTS

BAPTISTE KERLEGUER [†‡]

Abstract. This paper considers the surrogate modeling of a complex numerical code in a multi-fidelity framework when the code output is a time series and two code levels are available: a high-fidelity and expensive code level and a low-fidelity and cheap code level. The goal is to emulate a fast-running approximation of the high-fidelity code level. An original Gaussian process regression method is proposed that uses an experimental design of the low- and high-fidelity code levels. The code output is expanded on a basis built from the experimental design. The first coefficients of the expansion of the code output are processed by a co-kriging approach. The last coefficients are processed by a kriging approach with covariance tensorization. The resulting surrogate model provides a predictive mean and a predictive variance of the output of the high-fidelity code level. It is shown to have better performance in terms of prediction errors and uncertainty quantification than standard dimension reduction techniques.

Key words. Gaussian processes, time-series outputs, tensorized covariance, dimension reduction

AMS subject classifications. 60G15, 62F15, 62G08

1. Introduction.

Advances in scientific modeling have led to the development of complex and computationally expensive codes. To solve the problem of computation time for tasks such as optimization or calibration of complex numerical codes, surrogate models are used. The surrogate modeling approach consists in building a surrogate model of a complex numerical code from a data set computed from an experimental design. A well-known method to build surrogate models is Gaussian process (GP) regression. This method, also called kriging, was originally proposed by [15] for geostatistics. This method has subsequently been used for computer experiments and in particular in the field of uncertainty quantification (UQ), see [31, 32].

It is common for complex codes to have different versions that are more or less accurate and more or less computationally expansive. The particular case that interests us is when codes are hierarchical, i.e. they are classified according to their computational cost and their accuracy. The more accurate the code, the more expensive it is. The autoregressive scheme presented by [13] is the first major result in the field of multi-fidelity Gaussian process regression. This technique has been amended by [20] in order to reduce the overall co-kriging problem to several independent kriging problems. The papers [10, 28] present different application cases and [9] is a synthesis of the use of multi-fidelity for surrogate modeling. In [21] the author introduces objective prior for the hyperparameters of the autoregressive model.

New methods for multi-fidelity surrogate modeling have been introduced. In particular, Deep Gaussian processes have been proposed to solve cases where the interactions between code levels are more complex [26]. This type of methods can deal with UQ [4] but thus are time-consuming and do not scale up the dimension of outputs. Neural networks have also been used to emulate multi-fidelity computer codes. In particular, the method proposed in [22] is a neural network method with an AR(1)-based model and is scalable to high-dimensional outputs. However, UQ is not taken into account in this method.

Among the codes with high-dimensional outputs, we are interested in those whose

[†]CEA,DAM,DIF, F-91297, Arpajon, France baptiste.kerleguer@cea.fr

[‡]Centre de Mathématiques Appliquées, Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau Cedex, France

outputs are functions of one variable. When they are sampled, such outputs are called time series. Previous work has solved the problem of functional outputs only in the single-fidelity case. Two methods have been considered to solve the single-fidelity problem: reduce the dimension of the outputs [24] or adapt the covariance kernel [29]. In the context of dimension reduction, surrogate models generally neglect the uncertainty quantification in the basis, which can be problematic for the quantification of prediction uncertainty. Moreover, large data sets (containing many low-fidelity data) lead to ill-conditioned covariance matrices that are difficult to invert. As proposed in [27], it is possible to strongly constrain the covariance kernel which makes it possible to improve the estimation compared to the dimension reduction method. However, this method implies that the covariance must be separable, which reduces the use cases. Knowing that the AR(1) multi-fidelity model for GP regression uses co-kriging, [25] presents an interesting approach for co-kriging in the context of functional outputs, which is based on dimension reduction. An approach to multi-fidelity with functional outputs is presented in [11] for multivariate Hilbert space valued random fields.

In this work, we introduce an original approach to the construction of a surrogate model in the framework of hierarchical multi-fidelity codes with time-series outputs. The main idea is to combine a reduction method of the output dimension, that fits well with the autoregressive model of multi-fidelity co-kriging, and a special single-fidelity method that allows to treat time series output by GP regression with covariance tensorization. We address the case of one high-fidelity code and one low-fidelity code.

In Section 2 we give the main elements of the GP regression theory that are needed in our paper.

In Section 3 we develop a Cross-Validation Based (CVB) method to determine an appropriate basis based on K-fold cross-validation which uses only low-fidelity data and allows us to study the first two moments of the basis vectors. Once projected onto the reduced space it is possible to characterize the models of the first coefficients of the expansion of the code output onto the basis by the multi-fidelity GP regression method.

The projection of the code output onto the orthogonal of the reduced space forms the orthogonal part. If we neglect the orthogonal part, then we obtain the results of Section 3. However, the orthogonal part may not be negligible. In Section 4 we treat by GP regression with covariance tensorization the orthogonal part. The latter approach collectively addresses the orthogonal part of the high-fidelity code, and it makes it possible to better predict the output of the high-fidelity code and to better quantify the prediction uncertainty.

The results presented in the numerical example in Section 5 confirm that the processing of the orthogonal part is important. In this example we test the different methods presented in the paper as well as NN state of the art multi-fideliy methods, and we assess their performance in terms of prediction errors and uncertainty quantification.

2. Gaussian Process regression.

The autoregressive multi-fidelity model has been introduced by [13]. The authors in [20] have simplified the computation. We present our AR model in subsection 2.1. In parallel, GP regression has been used with covariance tensorization in [29] and improved in [3, 27]. This method allows to extend GP regression to time-series outputs. We have exploited this method to build our own methodology for time-series regression and we present it in subsection 2.2

Let us consider a complex numerical code where the input is a point $\boldsymbol{x} \in Q$, with Q being a domain in $\mathbb{R}^d$ and the output is a function of a one-dimensional variable.

We are interested in hierarchical codes, which means that there are several code levels that can be classified according to their fidelity. In this work, we focus on only two code levels, a high-fidelity code and a low-fidelity code. In what follows, H represents the high-fidelity code and L the low-fidelity code, the generic notation is $\mathrm{F} \in \{\mathrm{H}, \mathrm{L}\}$. For any given input $\boldsymbol{x}$, we can run the F code and observe the data $z_{\mathrm{F}}(\boldsymbol{x}, t)$ for all t in a fixed regular grid $\{t_u, u = 1, \ldots, N_t\}$ in $[0, 1]$. However, the cost of the code allows only a limited number of code calls. This induces the use of the experimental design $D_{\mathrm{F}} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N_{\mathrm{F}})}\right\}$. N_{F} is the number of observations of the code F. The $N_t \times N_{\mathrm{F}}$ matrix containing the observations for $\boldsymbol{x} \in D_{\mathrm{F}}$ is $\boldsymbol{Z}^{\mathrm{F}}_{\mathrm{obs}}$. Our goal is to predict the values of $(z_{\mathrm{H}}(\boldsymbol{x}, t_u))_{u=1,\ldots,N_t}$ given $\left(\boldsymbol{Z}^{\mathrm{H}}_{\mathrm{obs}}, \boldsymbol{Z}^{\mathrm{L}}_{\mathrm{obs}}\right)$ for a point $\boldsymbol{x} \in Q$ with the quantification of the prediction uncertainty. We model the prior knowledge of the code output $(z_{\mathrm{L}}, z_{\mathrm{H}})$ as a Gaussian process $(Z_{\mathrm{L}}, Z_{\mathrm{H}})$. We denote by $\mathcal{Z}^{\mathrm{F}}$ the random vector containing the random variables $\left(Z_{\mathrm{F}}(\boldsymbol{x}^{(i)}, t_u)\right)_{\substack{u=1,\ldots,N_t \\ u=1,\ldots,N_t}}$. The combination of $\mathcal{Z}^{\mathrm{L}}$ and $\mathcal{Z}^{\mathrm{H}}$ is $\mathcal{Z}$.

2.1. Multi-fidelity Gaussian process regression. In this section we want to build a surrogate model of a code $\alpha_{\mathrm{H}}(\boldsymbol{x})$ whose input $\boldsymbol{x}$ is in $Q \subset \mathbb{R}^d$ and whose scalar output is in $\mathbb{R}$. The construction of a surrogate model for complex computer code is difficult because of the lack of available experimental outputs. We consider the situation in which a cheaper and approximate code $\alpha_{\mathrm{L}}(\boldsymbol{x})$ is available. In this section, we apply the regression method presented by [13], reviewed in [8] and improved in [20].

We model the prior knowledge of the code output $(\alpha_{\mathrm{L}}, \alpha_{\mathrm{H}})$ as a Gaussian process $(A_{\mathrm{L}}, A_{\mathrm{H}})$. The vector containing the values of $\alpha_{\mathrm{F}}(\boldsymbol{x})$ at the points of the experimental design D_{F} are denoted by α^{F} and $\mathcal{A}^{\mathrm{F}}$ is the Gaussian vector containing $A_{\mathrm{F}}(\boldsymbol{x})$, $\boldsymbol{x} \in D_{\mathrm{F}}$.The combination of $\mathcal{A}^{\mathrm{L}}$ and $\mathcal{A}^{\mathrm{H}}$ is $\mathcal{A}$. So is α, the combination of α^{L} and α^{H}. We present the recursive model of multi-fidelity introduced by [20]. The experimental design is constructed such that $D_{\mathrm{H}} \subset D_{\mathrm{L}}$. We assume the low-fidelity code is computationally cheap, and that we have access to a large experimental design for the low-fidelity code, i.e. $N_{\mathrm{L}} \gg N_{\mathrm{H}}$.

We consider the hierarchical model introduced by [20]:

$$\left\{\begin{array}{rcl} A_{\mathrm{H}}(\boldsymbol{x}) & = & \rho_{\mathrm{L}}(\boldsymbol{x})\tilde{A}_{\mathrm{L}}(\boldsymbol{x}) + \delta(\boldsymbol{x}) \\ \tilde{A}_{\mathrm{L}}(\boldsymbol{x}) & \perp & \delta(\boldsymbol{x}) \\ \rho_{\mathrm{L}}(\boldsymbol{x}) & = & g^T_{\mathrm{L}}(\boldsymbol{x})\beta_\rho \end{array}\right. , \tag{2.1}$$

where $\perp$ means independence, ^{T} stands for the transpose,

$$[\delta(\boldsymbol{x})|\beta_H, \sigma_H] \sim \mathcal{GP}\left(f^T_{\mathrm{H}}(\boldsymbol{x})\beta_{\mathrm{H}}, \sigma^2_{\mathrm{H}} r_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{x}')\right), \tag{2.2}$$

and $\tilde{A}_{\mathrm{L}}(\boldsymbol{x})$ is a Gaussian process conditioned by the values α^{L}. Its distribution is the one of $\left[A_{\mathrm{L}}(\boldsymbol{x})|\mathcal{A}^{\mathrm{L}} = \alpha^{\mathrm{L}}, \beta_{\mathrm{L}}, \sigma_{\mathrm{L}}\right]$ with

$$[A_{\mathrm{L}}(\boldsymbol{x})|\beta_L, \sigma_L] \sim \mathcal{GP}\left(f^T_{\mathrm{L}}(\boldsymbol{x})\beta_{\mathrm{L}}, \sigma^2_{\mathrm{L}} r_{\mathrm{L}}(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{2.3}$$

Therefore, the distribution of $\tilde{A}_{\mathrm{L}}(\boldsymbol{x})$ is Gaussian with mean $\mu_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x})$ and variance $\sigma^2_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x})$:

$$\mu_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x}) = f^T_{\mathrm{L}}(\boldsymbol{x})\beta_{\mathrm{L}} + r^T_{\mathrm{L}}(\boldsymbol{x})C^{-1}_{\mathrm{L}}\left(\alpha^{\mathrm{L}} - F_{\mathrm{L}}\beta_{\mathrm{L}}\right), \tag{2.4}$$

$$\sigma^2_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x}) = \sigma^2_{\mathrm{L}}\big(r_{\mathrm{L}}(\boldsymbol{x}, \boldsymbol{x}) - r^T_{\mathrm{L}}(\boldsymbol{x})C^{-1}_{\mathrm{L}} r_{\mathrm{L}}(\boldsymbol{x})\big). \tag{2.5}$$

Here:

- $\mathcal{GP}$ means Gaussian process,
- $g_{\mathrm{L}}(\boldsymbol{x})$ is a vector of q_{L} regression functions,
- $f_{\mathrm{F}}(\boldsymbol{x})$ are vectors of p_{F} regression functions,
- $r_{\mathrm{F}}(\boldsymbol{x}, \boldsymbol{x}')$ are correlation functions,
- β_{F} are p_{F}-dimensional vectors,
- σ_{F}^2 are positive real numbers,
- β_ρ is a q-dimensional vector of adjustment parameters,
- $C_{\mathrm{F}} = \left(r_{\mathrm{F}}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\right)_{i,j=1}^{N_{\mathrm{F}}}$ is the $N_{\mathrm{F}} \times N_{\mathrm{F}}$ correlation matrix of $\mathcal{A}^{\mathrm{F}}$,
- $r_{\mathrm{F}}(\boldsymbol{x}) = \left(r_{\mathrm{F}}(\boldsymbol{x}, \boldsymbol{x}^{(i)})\right)_{i=1}^{N_{\mathrm{F}}}$ is the N_{F}-dimensional vector of correlations between $A_{\mathrm{F}}(\boldsymbol{x})$ and $\mathcal{A}^{\mathrm{F}}$,
- F_{F} is the $N_{\mathrm{F}} \times p_{\mathrm{F}}$ matrix containing the values of $f_{\mathrm{F}}^T(\boldsymbol{x})$ for $\boldsymbol{x} \in D_{\mathrm{F}}$.

For $\boldsymbol{x} \in Q$, the conditional distribution of $A_{\mathrm{H}}(\boldsymbol{x})$ is:

$$\left[A_{\mathrm{H}}(\boldsymbol{x}) | \mathcal{A} = \alpha, \beta, \beta_\rho, \sigma^2\right] \sim \mathcal{N}\left(\mu_{A_{\mathrm{H}}}(\boldsymbol{x}), \sigma^2_{A_{\mathrm{H}}}(\boldsymbol{x})\right), \tag{2.6}$$

where $\beta = \left(\beta_{\mathrm{H}}^T, \beta_{\mathrm{L}}^T\right)^T$ is the $p_{\mathrm{H}} + p_{\mathrm{L}}$-dimensional vector of regression parameters, $\sigma^2 = \left(\sigma_{\mathrm{L}}^2, \sigma_{\mathrm{H}}^2\right)$ are the variance parameters,

$$\begin{aligned} \mu_{A_{\mathrm{H}}}(\boldsymbol{x}) =& g_{\mathrm{L}}^T(\boldsymbol{x})\beta_\rho \mu_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x}) + f_{\mathrm{H}}^T(\boldsymbol{x})\beta_{\mathrm{H}} \\ &+ r_{\mathrm{H}}^T(\boldsymbol{x}) C_{\mathrm{H}}^{-1}\left(\alpha^{\mathrm{H}} - \rho^{\mathrm{L}}(D_{\mathrm{H}}) \odot \alpha^{\mathrm{L}}(D_{\mathrm{H}}) - F_{\mathrm{H}}\beta_{\mathrm{H}}\right) \end{aligned} \tag{2.7}$$

and

$$\sigma^2_{A_{\mathrm{H}}}(\boldsymbol{x}) = \left(g_{\mathrm{L}}^T(\boldsymbol{x})\beta_\rho\right)^2 \sigma^2_{\tilde{A}_{\mathrm{L}}}(\boldsymbol{x}) + \sigma_{\mathrm{H}}^2\left(1 - r_{\mathrm{H}}^T(\boldsymbol{x}) C_{\mathrm{H}}^{-1} r_{\mathrm{H}}(\boldsymbol{x})\right). \tag{2.8}$$

The notation $\odot$ is the element by element matrix product. $\rho^{\mathrm{L}}(D_{\mathrm{H}})$ is the N_{H}-dimensional vector containing the values of $\rho_{\mathrm{L}}(\boldsymbol{x})$ for $\boldsymbol{x} \in D_{\mathrm{H}}$. $\alpha^{\mathrm{L}}(D_{\mathrm{H}})$ is the N_{H}-dimensional vector containing the values of $\alpha_{\mathrm{L}}(\boldsymbol{x})$ at the points of D_{H}.

The prior distributions of the parameters β and σ are given in Appendix A. The hyper-parameters of the covariance kernels r_{L} and r_{H} can be estimated by maximum likelihood or by leave-one-out cross validation [2]. The nested property of the experimental design sets $D_{\mathrm{H}} \subset D_{\mathrm{L}}$ is not necessary to build the model but it is simpler to estimate the parameters with this assumption [33]. Moreover, the ranking of codes and the low computer cost of the low-fidelity code allow for a nested design for practical applications.

2.2. Gaussian process regression for functional outputs. In this subsection, we address GP regression for a simple-fidelity code with time-series output. For the calculation of surrogate models with functional outputs, there are two different techniques. The simplest ones are dimension reduction techniques as presented in [1, 24] (see Section 3). An alternative is presented here, this method is GP regression with covariance tensorization. The method is presented in [29] and the estimation of the hyper-parameters is from [27].

In this section and the following ones we consider that the output is a time-dependent function observed on a fixed time grid $\{t_u\}_{u=1,\cdots,N_t}$, with $N_t \gg 1$, which is called a time series.

The experimental design in a times-series output case is very different from a scalar output case. In particular, for a value $\boldsymbol{x}$ in the experimental design D, all the

t of the time grid are in the experimental design. The $N_t \times N_x$ matrix containing the observations is $\boldsymbol{Z}_{\text{obs}} = \left(z(\boldsymbol{x}^{(i)}, t_u)\right)_{\substack{u=1,\dots,N_t \\ i=1,\dots,N_x}}$. In GP regression we model the prior knowledge of the code output as a Gaussian process $Z(\boldsymbol{x}, t_u)$ with $\boldsymbol{x} \in Q$ and $u = 1, \dots, N_t$ with a covariance function C given by (2.10) and a mean function μ given by (2.9). We focus our attention to the case $N_t > N_x$. We assume that the covariance structure can be decomposed into two different functions representing the correlation in $\boldsymbol{x}$ and the correlation in t. If we choose well both functions, the kriging calculation is possible [27, 29].

In the following we present a simplification of the method proposed in [27]. The a priori $\mathbb{R}^{N_t}$-valued mean function is assumed to be of the form:

$$\mu(\boldsymbol{x}) = Bf(\boldsymbol{x}) \tag{2.9}$$

where $f(\boldsymbol{x})$ is a given $\mathbb{R}^M$-valued function and $B \in \mathcal{M}_{N_t \times M}(\mathbb{R})$ is to be estimated. We define by F the $N_x \times M$ matrix $[f^T(\boldsymbol{x}^{(i)})]_{i=1,\dots,N_x}$.

The a priori covariance function $C(t_u, t_{u'}, \boldsymbol{x}, \boldsymbol{x}')$ can be expressed with the $N_t \times N_t$ matrix R_t and the correlation function $C_x : Q \times Q \to [0, 1]$ with $C_x(\boldsymbol{x}, \boldsymbol{x}) = 1$:

$$C(t_u, t_{u'}, \boldsymbol{x}, \boldsymbol{x}') = R_t(t_u, t_{u'}) C_x(\boldsymbol{x}, \boldsymbol{x}'). \tag{2.10}$$

The covariance in time is expressed as a matrix because the temporal grid is finite and fixed. The covariance "matrix" (here a tensor) of $\left(Z(\boldsymbol{x}^{(j)}, t_u)\right)_{\substack{u=1,\dots,N_t \\ j=1,\dots,N_x}}$ is

$$R = R_t \otimes R_x, \tag{2.11}$$

with $(R_x)_{k,l} = C_x(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(l)})$ $k, l = 1, \dots, N_x$.

If R_x and R_t are not singular, then the a posteriori distribution of the $\mathbb{R}^{N_t}$-valued process Z given the covariance functions and the observations $\boldsymbol{Z}_{\text{obs}}$ is Gaussian:

$$(Z(\boldsymbol{x}, t_u))_{u=1,\dots,N_t} | R_t, C_x, \boldsymbol{Z}_{\text{obs}} \sim \mathcal{GP}(\mu_\star(\boldsymbol{x}), R_\star(\boldsymbol{x}, \boldsymbol{x}') R_t), \tag{2.12}$$

with the N_t-dimensional posterior mean:

$$\mu_\star(\boldsymbol{x}) = \boldsymbol{Z}_{\text{obs}} R_x^{-1} r_x(\boldsymbol{x}) + B_\star u(\boldsymbol{x}) \tag{2.13}$$

where $r_x(\boldsymbol{x})$ is the N_x-dimensional vector $\left(C_x(\boldsymbol{x}, \boldsymbol{x}^{(j)})\right)_{j=1,\dots,N_x}$. The posterior covariance function $R_\star(\boldsymbol{x}, \boldsymbol{x}')$ is :

$$R_\star(\boldsymbol{x}, \boldsymbol{x}') = c_\star(\boldsymbol{x}, \boldsymbol{x}') \left(1 + v_\star(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{2.14}$$

The functions that are used in the regression are

$$\begin{cases} u(\boldsymbol{x}) = f(\boldsymbol{x}) - F^T R_x^{-1} r_x(\boldsymbol{x}) \\ c_\star(\boldsymbol{x}, \boldsymbol{x}') = C_x(\boldsymbol{x}, \boldsymbol{x}') - r_x(\boldsymbol{x})^T R_x^{-1} r_x(\boldsymbol{x}') \\ v_\star(\boldsymbol{x}, \boldsymbol{x}') = u(\boldsymbol{x})^T (F^T R_x^{-1} F)^{-1} u(\boldsymbol{x}') c_\star^{-1}(\boldsymbol{x}, \boldsymbol{x}') \end{cases}, \tag{2.15}$$

and

$$B_\star = \boldsymbol{Z}_{\text{obs}} R_x^{-1} F (F^T R_x^{-1} F)^{-1}. \tag{2.16}$$

The correlation function C_x is assumed to be a Matérn $\frac{5}{2}$ kernel with a tensorized form, see [32, Chapter 4]:

$$C_x(\boldsymbol{x}, \boldsymbol{x}') = \prod_{i=1}^{d} \left(1 + \frac{\sqrt{5}|x_i - x_i'|}{\ell_{x_i}} + \frac{5|x_i - x_i'|^2}{3\ell_{x_i}^2}\right) \exp\left(-\frac{\sqrt{5}|x_i - x_i'|}{\ell_{x_i}}\right), \tag{2.17}$$

with $\boldsymbol{\ell_x} = (\ell_{x_1}, \dots, \ell_{x_d})$ the vector of correlation lengths. Other choices are of course possible. R_t is estimated using R_x^{-1} and the observations $\boldsymbol{Z}_{\text{obs}}$ by maximum likelihood, as in [27]:

$$\widehat{R_t} = \frac{1}{N_x} \left(\boldsymbol{Z}_{\text{obs}} - \hat{\boldsymbol{Z}} \right) R_x^{-1} \left(\boldsymbol{Z}_{\text{obs}} - \hat{\boldsymbol{Z}} \right)^T, \tag{2.18}$$

with $\hat{\boldsymbol{Z}}$ is the $N_t \times N_x$ matrix of empirical means $\hat{Z}_{u,i} = \frac{1}{N_x} \sum_{j=1}^{N_x} (\boldsymbol{Z}_{\text{obs}})_{u,j}$, $\forall i = 1, \dots, N_x$ and $u = 1, \dots, N_t$.

It remains only to estimate the vector of correlation lengths $\boldsymbol{\ell_x} = (\ell_{x_1}, \dots, \ell_{x_d})$ to determine the function C_x. As presented in [27], the maximum likelihood estimation is not well defined for $\boldsymbol{\ell_x}$. Indeed the approximation of R_t by (2.18) is singular because $N_x < N_t$. In fact we do not need to invert R_t as seen in Equations (2.12)–(2.16). The method generally used to estimate the correlation lengths is cross-validation and, in our case, Leave-One-Out (LOO). The LOO mean square error that needs to be minimized is:

$$\varepsilon^2(\boldsymbol{\ell_x}) = \sum_{k=1}^{N_x} \| \mu_\star^{(-k)}(\boldsymbol{x}^{(k)} | \boldsymbol{Z}_{\text{obs}}^{(-k)}, \boldsymbol{l_x}) - \boldsymbol{Z}_{\text{obs}}(\boldsymbol{x}^{(k)}) \|^2, \tag{2.19}$$

where $\mu_\star^{(-k)}(\boldsymbol{x}^{(k)} | \boldsymbol{Z}_{\text{obs}}^{(-k)}, \boldsymbol{l_x})$ is the $\mathbb{R}^{N_t}$-valued prediction mean obtained with the correlation length vector $\boldsymbol{l_x}$, using all observations except the k-th, at the point $\boldsymbol{x}^{(k)}$ and $\| \cdot \|$ is the Euclidean norm in $\mathbb{R}^{Nt}$. We can use an expression of $\varepsilon^2(\boldsymbol{l_x})$ that does not require multiple regression, as in [2, 6]. For more detail see Appendix B.

3. AR(1) multi-fidelity model with projection. To carry out a single-fidelity regression for a code whose output is a time series we can use the method presented in Subsection 2.2 or use output dimension reduction. For dimension reduction, as in [24], a basis is chosen. The functional output is expanded onto this basis, the expansion is truncated and a surrogate model is built for each scalar-valued coefficient of the truncated expansion. In our case, we deal with both multi-fidelity and time-series outputs.

One solution could be to use covariance tensorization in the multi-fidelity framework. This method requires the inversion of large covariance matrices. However, the inversion methods that are possible in a single-fidelity framework become impossible in a multi-fidelity framework because the matrices are then too ill-conditioned to be inverted.

This leads us to introduce new methods. The most naive method consists in starting again from the dimension reduction technique and to carry out the projection of the outputs of the two codes onto the same appropriate basis. It is therefore possible to use the model Equation (2.1). The problem is that is not possible to define a basis that is optimal for both the high- and low-fidelity codes. A basis estimated from the low-fidelity data is preferred as it is more robust thanks to the larger number of data. Thus the loss of high-fidelity information is significant which leads us to introduce our new method in Section 4. The originality of our approach is to keep the dimension reduction technique but also to use the residual high-fidelity data to carry out a GP regression with covariance tensorization.

3.1. Model. We recall that $(Z_{\text{L}}, Z_{\text{H}})$ is a stochastic process. The temporal grid is $\{t_u\}_{u \in \{1,\dots,N_t\}}$. N_{F} observations are available for different values of $\boldsymbol{x}$ at the fidelity F.

Let $\mathbf{\Gamma}$ be an orthogonal $N_t \times N_t$ matrix. The columns of $\mathbf{\Gamma}$, Γ_i, form an orthonormal basis of $\mathbb{R}^{N_t}$. We assume that, given $\mathbf{\Gamma}$, $(Z_{\mathrm{L}}, Z_{\mathrm{H}})$ are Gaussian processes with covariance matrices that can be diagonalized on the basis formed by the columns of $\mathbf{\Gamma}$. The processes Z_{F} can then be expanded as:

$$Z_{\mathrm{L}}(\boldsymbol{x}, t_u) = \sum_{i=1}^{N_t} A_{i,\mathrm{L}}(\boldsymbol{x})\Gamma_i(t_u), \tag{3.1}$$

$$Z_{\mathrm{H}}(\boldsymbol{x}, t_u) = \sum_{i=1}^{N_t} A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u), \tag{3.2}$$

where $(A_{i,\mathrm{L}}(\boldsymbol{x}), A_{i,\mathrm{H}}(\boldsymbol{x}))$ are Gaussian processes which are independent with respect to i, given $\mathbf{\Gamma}$.

Let $(\alpha_{i,\mathrm{F}}(\boldsymbol{x}))_{i=1}^{N_t}$ be the $\mathbb{R}^{N_t}$-valued function:

$$\alpha_{i,\mathrm{F}}(\boldsymbol{x}) = \sum_{u=1}^{N_t} z_{\mathrm{F}}(\boldsymbol{x}, t_u)\Gamma_i(t_u). \tag{3.3}$$

We denote by α_i^{F} the $1 \times N_F$ row vector $\left(\alpha_{i,\mathrm{F}}(\boldsymbol{x}^{(j)})\right)_{j=1}^{N_F}$ that contains the available data. The full data set is $\alpha = \left(\alpha^{\mathrm{L}}, \alpha^{\mathrm{H}}\right)$.

Consequently we will use the method presented in Subsection 2.1 given $\mathbf{\Gamma}$. This leads us to the model presented in Equation (3.4). Given $\mathbf{\Gamma}$, $\forall i \in \{1, \ldots, N_t\}$,

$$\left\{ \begin{array}{rcl} A_{i,\mathrm{H}}(\boldsymbol{x}) & = & \rho_{i,\mathrm{L}}(\boldsymbol{x})\tilde{A}_{i,\mathrm{L}}(\boldsymbol{x}) + \delta_i(\boldsymbol{x}) \\ \tilde{A}_{i,\mathrm{L}}(\boldsymbol{x}) & \perp & \delta_i(\boldsymbol{x}) \\ \rho_{i,\mathrm{L}}(\boldsymbol{x}) & = & g_i^T(\boldsymbol{x})\beta_{\rho_{\mathrm{L}},i} \end{array} \right. , \tag{3.4}$$

where:

$$[\delta_i(\boldsymbol{x})|\ \mathbf{\Gamma}, \sigma_{i,H}, \beta_{i,H}] \sim \mathcal{GP}\left(f_{i,\mathrm{H}}^T(\boldsymbol{x})\beta_{i,\mathrm{H}}, \sigma_{i,\mathrm{H}}^2 r_{i,\mathrm{H}}(\boldsymbol{x}, \boldsymbol{x}')\right),$$

and $\tilde{A}_{i,\mathrm{L}}(\boldsymbol{x})$ a Gaussian process conditioned by the values α^{L}. The distribution of $\tilde{A}_{i,\mathrm{L}}(\boldsymbol{x})$ is the one of $\left[A_{i,\mathrm{L}}(\boldsymbol{x})|\mathbf{\Gamma}, \mathcal{A}^{\mathrm{L}} = \alpha^{\mathrm{L}}, \beta_{i,\mathrm{L}}, \sigma_{i,\mathrm{L}}\right]$ where:

$$[A_{i,\mathrm{L}}(\boldsymbol{x})|\mathbf{\Gamma}, \sigma_{i,L}, \beta_{i,L}] \sim \mathcal{GP}\left(f_{i,\mathrm{L}}^T(\boldsymbol{x})\beta_{i,\mathrm{L}}, \sigma_{i,\mathrm{L}}^2 r_{i,\mathrm{L}}(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{3.5}$$

$g_i(\boldsymbol{x})$ are vectors of q regression functions, $f_{i,\mathrm{F}}(\boldsymbol{x})$ are vectors of p_{F} regression functions, $r_{i,\mathrm{F}}(\boldsymbol{x}, \boldsymbol{x}')$ are correlation functions, $\beta_{i,\mathrm{F}}$ are p_{F}-dimensional vectors, $\beta_{\rho_{\mathrm{L}},i}$ are q-dimensional vectors and $\sigma_{i,\mathrm{F}}^2$ are positive real numbers. For simplicity the regression functions g_i and $f_{i,\mathrm{F}}$ do not depend on i.

The model depends on $\mathbf{\Gamma}$, which is why we discuss in Subsection 3.2 the choice of the basis.

3.2. Basis. In this section we present different models for the random orthogonal matrix $\mathbf{\Gamma}$. Its law depends on the available information. If we have access to a lot of information based on the output of our code, we can use a Dirac distribution concentrated on one orthogonal matrix $\boldsymbol{\gamma}$ (it is a form of plug-in method). In contrast, the least informative law is the Uniform Law, i.e. the Haar measure over the group of orthogonal matrices. In order to make the best use of the available information, i.e. the known results of the code, an empirical law can be used.

3.2.1. Dirac distribution. We can choose the distribution of the random matrix $\mathbf{\Gamma}$ as a Dirac distribution concentrated on a well chosen orthogonal matrix $\boldsymbol{\gamma}$. This matrix is chosen when the basis is known a priori or if the basis can be efficiently estimated from the observed code outputs. Motivated by the remark below Equation (3.2), the matrix $\boldsymbol{\gamma}$ can be computed using the singular value decomposition (SVD) of the code outputs.

The general idea is to choose subsets $\tilde{D}_{\mathrm{F}} \subset D_{\mathrm{F}}$ of size $\tilde{N}_{\mathrm{F}}$ and to apply a SVD on the $N_t \times \left(\tilde{N}_{\mathrm{H}} + \tilde{N}_{\mathrm{L}}\right)$ matrix $\tilde{\boldsymbol{Z}}_{\mathrm{obs}}$ that contains the observed values $(z_{\mathrm{H}}(\boldsymbol{x}, t_u))_{\substack{u=1,\dots,N_t \\ \boldsymbol{x}\in\tilde{D}_{\mathrm{H}}}}$ and $(z_{\mathrm{L}}(\boldsymbol{x}, t_u))_{\substack{u=1,\dots,N_t \\ \boldsymbol{x}\in\tilde{D}_{\mathrm{L}}}}$. The SVD gives:

$$\tilde{\boldsymbol{Z}}_{\mathrm{obs}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{V}}^T. \tag{3.6}$$

The choice of $\boldsymbol{\gamma}$ is $\tilde{\boldsymbol{U}}$.

The first idea is to mix all available data, high- and low-fidelity: $\tilde{D}_{\mathrm{H}} = D_{\mathrm{H}}$ and $\tilde{D}_{\mathrm{L}} = D_{\mathrm{L}}$. However, we typically have that $N_{\mathrm{L}} \gg N_{\mathrm{H}}$, so the basis is mainly built from the low-fidelity data. In addition, the small differences in the data between high- and low-fidelity code outputs that would be useful to build the basis have negligible impact because they are overwhelmed by the low-fidelity data. This method is not appropriate in our framework.

We have to choose between high- and low- fidelity. High-fidelity has the advantage of being closer to the desired result. However, it is also almost impossible to validate the chosen $\boldsymbol{\gamma}$ because the high-fidelity data size N_{H} is small. The low-fidelity data set is larger, hence the estimation of $\boldsymbol{\gamma}$ is more robust. In order to choose $\boldsymbol{\gamma}$, we therefore suggest to use the low-fidelity data and to calculate the SVD with $\tilde{D}_{\mathrm{H}} = \emptyset$ and $\tilde{D}_{\mathrm{L}} = D_{\mathrm{L}}$.

3.2.2. Uniform distribution. We can choose a random matrix $\mathbf{\Gamma}$ using the Haar measure on the orthogonal group O_{N_t}, the group of $N_t \times N_t$ orthogonal matrices. This is the Uniform Orthogonal Matrix Law.

To generate a random matrix from the Haar measure over O_{N_t}, one can first generate a $N_t \times N_t$ matrix with independent and identically distributed coefficients with the reduced normal distribution, then, apply the Gram-Schmidt process onto the matrix. As shown in [5], this generator produces a random orthogonal matrix with the uniform orthogonal matrix law. This method completely ignores the available data and is not appropriate in our framework.

3.2.3. Cross-Validation Based distribution. The downside of the Dirac distribution is that the uncertainties on the basis estimation are not taken into account. A cross-validation based (CVB) method to assess the uncertainty estimation is therefore considered.

The proposed method uses only the low-fidelity data because it is assumed that there are too few high-fidelity data to implement this method, so $\tilde{D}_{\mathrm{H}} = \emptyset$. For the construction of the basis we try to have different sets to evaluate the basis in order to have empirical estimates of the moments of the basis vectors. Let k be a fixed integer in $\{1, \dots, N_{\mathrm{L}}\}$. Let $I = \{J_1, \dots, J_k\}$ be a random set of k elements in $\{1, \dots, N_{\mathrm{L}}\}$, with uniform distribution over the subsets of k elements in $\{1, \dots, N_{\mathrm{L}}\}$. The empirical distribution for the matrix $\mathbf{\Gamma}$ is defined as follows: for any bounded function $f : O_{N_t} \to \mathbb{R}$,

$$\mathbb{E}\left[f(\mathbf{\Gamma}_I)\right] = \frac{1}{\binom{N_{\mathrm{L}}}{k}} \sum_{\{j_1,\dots,j_k\}\subset\{1,\dots,N_{\mathrm{L}}\}} f(\tilde{\boldsymbol{U}}_{[J_1,\dots,J_k]}), \tag{3.7}$$

where $\tilde{\boldsymbol{U}}_{[J_1,\dots,J_k]}$ is the matrix of the left singular vectors of the SVD of $\left(z_{\mathrm{L}}(\boldsymbol{x}^{(i)},t_u)\right)_{\substack{u\in\{1,\dots,N_t\}\\ i\in\{1,\dots,N_{\mathrm{L}}\}\setminus\{J_1,\dots,J_k\}}}$. This distribution depends on the choice of k, that will be discussed in Section 5.

3.3. Predictive mean and variance. The goal of this section is to calculate the posterior distribution of $Z_{\mathrm{H}}(\boldsymbol{x},t_u)$. The problem can be split into two parts: the multi-fidelity regression of the basis coefficients knowing $\boldsymbol{\Gamma}$ and the integration with respect to the distribution of $\boldsymbol{\Gamma}$. The Dirac and CVB distributions described in subsection 3.2 can be used to define the law of $\boldsymbol{\Gamma}$.

Multi-fidelity surrogate modeling of the coefficients. By applying the model proposed in Subsection 2.1 we can therefore deduce the prediction mean and variance. Their expressions are given in Appendix D.1

3.3.1. Dirac law of $\boldsymbol{\Gamma}$. Here we assume that the law of $\boldsymbol{\Gamma}$ is Dirac at $\boldsymbol{\gamma}$. Consequently, the posterior distribution of $Z_{\mathrm{H}}(\boldsymbol{x},t)$ is Gaussian. In order to characterize the law of $Z_{\mathrm{H}}(\boldsymbol{x},t_u)$ it is necessary and sufficient to compute its mean and variance.

Mean:. The posterior mean is:

$$\mathbb{E}\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|\mathcal{A}=\alpha\right]=\sum_{i=1}^{N_t}\gamma_i(t_u)\mathbb{E}\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right], \tag{3.8}$$

where the expectation $\mathbb{E}\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$ is given by (D.1).

Variance:. The posterior variance:

$$\mathbb{V}\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|\mathcal{A}=\alpha\right]=\sum_{i=1}^{N_t}\gamma_i^2(t_u)\mathbb{V}\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right], \tag{3.9}$$

where the variance $\mathbb{V}\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$ is given by (D.2).

3.3.2. CVB law of $\boldsymbol{\Gamma}$. Because the law is different from Dirac the posterior distribution of $Z_{\mathrm{H}}(\boldsymbol{x},t)$ is not Gaussian anymore. However, we can characterize the posterior mean and the variance of $Z_{\mathrm{H}}(\boldsymbol{x},t_u)$.

We denote $\mathbb{E}_\alpha\left[\cdot\right]=\mathbb{E}\left[\cdot|\mathcal{A}=\alpha\right]$, $\mathbb{V}_\alpha\left[\cdot\right]=\mathbb{V}\left[\cdot|\mathcal{A}=\alpha\right]$, $\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\cdot\right]=\mathbb{E}\left[\cdot|\mathcal{Z}=\boldsymbol{Z}_{\mathrm{obs}}\right]$ and $\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\cdot\right]=\mathbb{V}\left[\cdot|\mathcal{Z}=\boldsymbol{Z}_{\mathrm{obs}}\right]$.

Mean. The linearity of the expectation and the law of total expectation give:

$$\mathbb{E}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)\right]=\sum_{i=1}^{N_t}\mathbb{E}_\alpha[\Gamma_i(t_u)\mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]], \tag{3.10}$$

where the expectation $\mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]$ is given by Equation (D.1).

Variance. The law of total variance gives :

$$\mathbb{V}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)\right]=\mathbb{V}_\alpha\left[\mathbb{E}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma}\right]\right]+\mathbb{E}_\alpha\left[\mathbb{V}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma}\right]\right] \tag{3.11}$$

By Appendix D.2 we get:

$$\begin{array}{c}\mathbb{V}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)\right]=\sum_{i=1}^{N_t}\mathbb{V}_\alpha\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\ +\sum_{i,j=1,i\neq j}^{N_t}\mathrm{Cov}_\alpha(\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right],\Gamma_j(t_u)\mathbb{E}_\alpha\left[A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right])\\ +\sum_{i=1}^{N_t}\mathbb{E}_\alpha\left[\Gamma_i^2(t_u)\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\end{array}. \tag{3.12}$$

Equations (3.10) and (3.12) are combinations of expectations of explicit functions of $\boldsymbol{\Gamma}$. We can compute the result using our knowledge on the law of $\boldsymbol{\Gamma}$. The expectation of a function of $\boldsymbol{\Gamma}$ is given by Equation (3.7).

3.4. Truncation. There is a problem with the surrogate modeling of the coefficients of the decomposition with indices larger than N_{L}. Indeed, we typically have $N_{\mathrm{L}} < N_t$ so the vectors Γ_i with indices larger than N_{L} of the basis are randomly constructed, which is not suitable for building surrogate models. To solve this problem, it is possible to truncate the sum. Only the first N coefficients, with $N \leq N_{\mathrm{L}}$ are calculated. This would be reasonable if the contributions of the terms $A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u)$ for $i > N$ were negligible. However it turns out that these terms are often not collectively negligible (see Section 5) and the truncation method does not achieve a good bias-variance trade-off even when optimizing with respect to N (by a cross validation procedure for instance). The high- and low-fidelity outputs do not necessarily have the same forms. Thus it is possible that an important part of the high-fidelity code is neglected because it is not taken into account by the sub-space spanned by $\{\Gamma_i\}_{i\leq N}$. We will therefore propose in the next section an original method to tackle this problem.

4. AR(1) multi-fidelity model with tensorized covariance and projection.

The naive method presented in Section 3 has many flaws. It leaves part of the output untreated by regression and the variance is underestimated. The major problem with the solution we propose in Section 3, is that we typically have $N_{\mathrm{L}} < N_t$. Consequently for $i > N_{\mathrm{L}}$ the vectors Γ_i of the basis do not represent the typical variations of Z_{H}. We should find an appropriate way to predict the law of the projection of Z_{H} on $\operatorname{span}\{\Gamma_i, i > N_{\mathrm{L}}\}$.

One interesting approach is to apply the covariance tensorization method to the orthogonal part. This idea is to address the last terms of the expression collectively through a GP model with tensorized covariance structure. This allows us to split the problem into two parts. The first part is to compute (as presented in the previous section) the first N terms of the expansion of Z_{H} onto the basis by using a co-kriging approach. The second part is to compute the projection of Z_{H} onto the orthogonal space by using a kriging approach with tensorized covariance. The choice of the optimal N will be carried out by a K-fold cross-validation method.

4.1. Decomposition. The proposed method is based on the decomposition of the outputs, as presented in Equation (3.1).

Projection. Let N be an integer, smaller than the time dimension N_t. Let $\boldsymbol{\Gamma}$ be an orthogonal matrix, as presented in Subsection 3.2, the columns of $\boldsymbol{\Gamma}$ are $\{\Gamma_i\}_{i=1,\cdots,N_t}$. As discussed in Subsection 3.4 the full computation of all N_t surrogate models may not give good results. This leads to the idea of the introduction of the orthogonal subspaces $S_N^{\parallel}$ and $S_N^{\perp}$, where $S_N^{\parallel} = \operatorname{span}\{\Gamma_1, \ldots, \Gamma_N\}$ and $S_N^{\perp} = \operatorname{span}\{\Gamma_{N+1}, \ldots, \Gamma_{N_t}\}$.

With a given basis $\boldsymbol{\Gamma}$ it is possible to decompose the code outputs. The decomposition over the basis $\boldsymbol{\Gamma}$ gives us coefficients. We decompose Z_{H} and Z_{L} over the subspace $S_N^{\parallel}$. The rests are denoted $Z_{\mathrm{H}}^{\perp}$ and $Z_{\mathrm{L}}^{\perp}$. Consequently, we get:

$$Z_{\mathrm{L}}(\boldsymbol{x}, t_u) = Z_{\mathrm{L}}^{\parallel}(\boldsymbol{x}, t_u) + Z_{\mathrm{L}}^{\perp}(\boldsymbol{x}, t_u) = \sum_{i=1}^{N} A_{i,\mathrm{L}}(\boldsymbol{x})\Gamma_i(t_u) + Z_{\mathrm{L}}^{\perp}(\boldsymbol{x}, t_u) \tag{4.1}$$

and

$$Z_{\mathrm{H}}(\boldsymbol{x}, t_u) = Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x}, t_u) + Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u) = \sum_{i=1}^{N} A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u) + Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u). \tag{4.2}$$

We are able to describe the code outputs with the basis $\mathbf{\Gamma}_N = \{\Gamma_i\}_{i=1,\dots,N}$ of S_N, the coefficients $A_{i,\mathrm{H}}$ and $A_{i,\mathrm{L}}$, and the orthogonal parts $Z_{\mathrm{L}}^{\perp}$ and $Z_{\mathrm{H}}^{\perp}$. We denote by $\alpha_{i,\mathrm{H}}$ and $\alpha_{i,\mathrm{L}}$ the available data sets, the full set is called α. The expression of $\alpha_{i,\mathrm{F}}$ is given by Equation (3.3).

We will use the method presented in Subsection 2.1 for all $i \leq N$. Given $\mathbf{\Gamma}$,

$$\left\{ \begin{array}{rcl} A_{i,\mathrm{H}}(\boldsymbol{x}) & = & \rho_{i,\mathrm{L}}(\boldsymbol{x})\tilde{A}_{i,\mathrm{L}}(\boldsymbol{x}) + \delta_i(\boldsymbol{x}) \\ \tilde{A}_{i,\mathrm{L}}(\boldsymbol{x}) & \perp & \delta_i(\boldsymbol{x}) \\ \rho_{i,\mathrm{L}}(\boldsymbol{x}) & = & g_i^T(\boldsymbol{x})\beta_{i,\rho_{\mathrm{L}}} \end{array} \right. , \tag{4.3}$$

where:

$$[\delta_i(\boldsymbol{x})|\mathbf{\Gamma}, \beta_{i,H}, \sigma_{i,H}] \sim \mathcal{GP}\left(f_{i,\mathrm{H}}^T(\boldsymbol{x})\beta_{i,\mathrm{H}}, \sigma_{i,\mathrm{H}}^2 r_{i,\mathrm{H}}(\boldsymbol{x}, \boldsymbol{x}')\right),$$

and $\tilde{A}_{i,\mathrm{L}}(\boldsymbol{x})$ is a Gaussian process conditioned by α^{L}. Its distribution is the one of $\left[A_{i,\mathrm{L}}(\boldsymbol{x})|\mathbf{\Gamma}, \mathcal{A}^{\mathrm{L}} = \alpha^{\mathrm{L}}, \beta_{\mathrm{L}}, \sigma_{\mathrm{L}}\right]$ where the law of $[A_{i,\mathrm{L}}(\boldsymbol{x})|\mathbf{\Gamma}, \beta_{i,L}, \sigma_{i,L}]$ is of the form (3.5).

$$[A_{i,\mathrm{L}}(\boldsymbol{x})|\mathbf{\Gamma}, \beta_{i,L}, \sigma_{i,L}] \sim \mathcal{GP}\left(f_{i,\mathrm{L}}^T(\boldsymbol{x})\beta_{i,\mathrm{L}}, \sigma_{i,\mathrm{L}}^2 r_{i,\mathrm{L}}(\boldsymbol{x}, \boldsymbol{x}')\right).$$

g_i are vectors of q regression functions, $f_{i,\mathrm{F}}(\boldsymbol{x})$ are vectors of p_{F} regression functions, $r_{i,\mathrm{F}}(\boldsymbol{x}, \boldsymbol{x}')$ are correlation functions, $\beta_{i,\mathrm{F}}$ are p_{F}-dimensional vectors, $\beta_{i,\rho_{\mathrm{L}}}$ are q-dimensional vectors and $\sigma_{i,\mathrm{F}}^2$ are positive real numbers.

For the orthogonal part projected onto $S_N^{\perp}$ the method is different. The hypothesis is that the projection $Z_{\mathrm{L}}^{\perp}(\boldsymbol{x}, t_u)$ of $Z_{\mathrm{L}}(\boldsymbol{x}, t_u)$ has a negligible influence on the projection $Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u)$ of $Z_{\mathrm{H}}(\boldsymbol{x}, t_u)$. Our assumption is that $Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u)$ is a Gaussian process with a tensorized covariance. The method we will use on $Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u)$ is described in Subsection 2.2.

Note that the value $N = 0$ corresponds to full single-fidelity, in this case we use only GP regression with covariance tensorization as in subsection 2.2. For $N = N_{\mathrm{L}}$ the dimension reduction is minimal and co-kriging is applied to all pairs $(A_{i,\mathrm{L}}, A_{i,\mathrm{H}})$ for $i \leq N_{\mathrm{L}}$. We will see in section 5 that the optimal N is in fact positive but smaller than N_{L}.

4.2. Predictive mean and covariance. In this section we first make a quick reminder of the methods presented in Section 2. Moreover, with different assumptions about the law of $\mathbf{\Gamma}$ we present the regression using the model and the data.

Multi-fidelity of coefficients. As in subsection 3.3 we compute the N multi-fidelity models of the first N coefficients of the expansion of the code output given $\mathbf{\Gamma}$. If we apply the method proposed in subsection 2.1 we can therefore deduce the prediction mean and variance, as in subsection 3.3.

Tensorized covariance regression. The orthogonal part of the regression is computed using the method presented in subsection 2.2. The adaptation is that the regression must be carried out in subspace $S_N^{\perp}$ given $\mathbf{\Gamma}$,

$$\boldsymbol{Z}_{\mathrm{obs}}^{\perp} = \boldsymbol{Z}_{\mathrm{obs}} - \left(\sum_{i=1}^{N} \alpha_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u)\right)_{\substack{u=1,\cdots,N_t \\ \boldsymbol{x}\in D_{\mathrm{H}}}}, \tag{4.4}$$

where $\alpha_{i,\mathrm{H}}$ is given by Equation (3.3).

This does not have any consequence on the $\boldsymbol{x}$ part but only on the t part. Contrarily to $Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x}, t_u)$ only one surrogate model is needed for $Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u)$. The detail of how we can deal with $Z_{\mathrm{H}}^{\perp}(\boldsymbol{x}, t_u)$ and $Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x}, t_u)$ is explained in Appendix C.

4.2.1. Dirac law of $\boldsymbol{\Gamma}$. Here we assume that $\boldsymbol{\Gamma}$ is known and its distribution is Dirac at $\boldsymbol{\gamma}$. Consequently, as in Section 3, $Z_{\mathrm{H}}(\boldsymbol{x},t)$ is a Gaussian process by linear combination of independent Gaussian processes. Its posterior distribution is completely determined if we can evaluate its mean and covariance.

Mean. The $\Gamma_i(t_u)$' s are constant and equal to $\gamma_i(t_u)$. Consequently:

$$\mathbb{E}_\alpha\left[Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x},t_u)|N\right] = \sum_{i=1}^{N} \mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})]\gamma_i(t_u), \tag{4.5}$$

and

$$\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}] = \textstyle\sum_{i=1}^{N} \mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})]\Gamma_i(t_u) + \mathbb{E}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] , \tag{4.6}$$

where $\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\right]$ is given by (D.1) and $\mathbb{E}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]$ by (2.13).

Variance. The formula of the variance is:

$$\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t)\right] = \mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\right]\gamma_i(t)^2. \tag{4.7}$$

The uncorrelation of the coefficients $A_{i,\mathrm{F}}(\boldsymbol{x})$ gives $\mathrm{Cov}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x}), A_{j,\mathrm{H}}(\boldsymbol{x})\right] = 0$, for $i \neq j$ and $\mathrm{Cov}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\gamma_i(t_u), Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)\right] = 0$. The expression of the variance becomes simple:

$$\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = \textstyle\sum_{i=1}^{N} \mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\right]\gamma_i(t_u)^2 + \mathbb{V}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right], \tag{4.8}$$

where $\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\right]$ is given by (D.2) and $\mathbb{V}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]$ is given by (2.12).

4.2.2. CVB law of $\boldsymbol{\Gamma}$. The posterior distribution of $Z_{\mathrm{H}}(\boldsymbol{x},t)$ is not Gaussian anymore. However to predict the output of the high-fidelity code and to quantify the prediction uncertainty we are able to compute the posterior mean and variance of $Z_{\mathrm{H}}(\boldsymbol{x},t)$.

Mean. We can decompose the process into two parts:

$$Z_{\mathrm{H}}(\boldsymbol{x},t_u) = Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x},t_u) + Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u). \tag{4.9}$$

The linearity of the expectation gives us:

$$\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}] = \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}[A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u)] + \mathbb{E}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]. \tag{4.10}$$

The theorem of total expectation gives us:

$$\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\parallel}(\boldsymbol{x},t_u)|N\right] = \sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]], \tag{4.11}$$

and therefore,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = &\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}[\Gamma_i(t_u)\mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]] \\ &+ \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}^{\perp}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|N,\boldsymbol{\ell}_{\boldsymbol{x}},\boldsymbol{\Gamma}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right].\end{aligned} \tag{4.12}$$

where $\mathbb{E}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]$ is given by Equation (D.1) and $\mathbb{V}_\alpha\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]$ is given by Equation (2.13). Equation (4.12) is a combination of expectations of explicit functions of $\boldsymbol{\Gamma}$, which can be computed by Equation (3.7).

Variance. The theorem of the total variance gives us:

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[Z_{\text{H}}\left(\boldsymbol{x},t_u\right)|N,\boldsymbol{\ell_x}\right]=\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\text{obs}}}\left[Z_{\text{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x}\right]|N,\boldsymbol{\ell_x}\right]\\+\mathbb{E}_{\boldsymbol{Z}_{\text{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[Z_{\text{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x}\right]|N,\boldsymbol{\ell_x}\right]\end{array}. \tag{4.13}$$

By Appendix D.3 we get:

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[Z_{\text{H}}\left(\boldsymbol{x},t_u\right)|N,\boldsymbol{\ell_x}\right]=\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[\mathbb{E}^{\perp}_{\boldsymbol{Z}_{\text{obs}}}\left[Z^{\perp}_{\text{H}}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x}\right]|N,\boldsymbol{\ell_x}\right]\\+\mathbb{E}_{\boldsymbol{Z}_{\text{obs}}}\left[\mathbb{V}_{\alpha}\left[Z^{\perp}_{\text{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x}\right]|N,\boldsymbol{\ell_x}\right]\\+\sum_{i=1}^{N}\mathbb{V}_{\boldsymbol{Z}_{\text{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_{\alpha}\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\+\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{Z}_{\text{obs}}}\left[\Gamma_i(t_u)^2\mathbb{V}_{\alpha}\left[A_i\left(\boldsymbol{x}\right)|\boldsymbol{\Gamma}\right]\right]\\+\sum_{i,j=1;i\neq j}^{N}\text{Cov}_{\boldsymbol{Z}_{\text{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_{\alpha}\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right],\Gamma_j(t_u)\mathbb{E}_{\alpha}\left[A_{j,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\+2\sum_{i=1}^{N}\text{Cov}_{\boldsymbol{Z}_{\text{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_{\alpha}\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right],\mathbb{E}^{\perp}_{\boldsymbol{Z}_{\text{obs}}}\left[Z^{\perp}_{\text{H}}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x}\right]|N,\boldsymbol{\ell_x}\right]\end{array}. \tag{4.14}$$

where $\mathbb{V}_{\alpha}[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]$ is given by Equation (D.2) and $\mathbb{V}_{\alpha}\left[Z^{\perp}_{\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]$ is given by Equation (2.14). Equation (D.12) is a combination of expectations and variances of explicit functions of $\boldsymbol{\Gamma}$, which can be computed by Equation (3.7).

4.3. Effective dimension. For the formulas in Subsection 4.2 to be valid, N must be fixed. We may choose N by a knowledge on the physical system or on the code but it is impossible in most cases due to the high/low-fidelity differences. The best solution is generally to determine N by a K-fold cross validation procedure.

The criterium that we choose to maximize is:

$$Q^2_N(t_u)=1-\frac{\sum_{k=1}^{N_{\text{H}}}\left(z_{\text{H}}(\boldsymbol{x}^{(k)},t_u)-\mathbb{E}\left[Z_{\text{H}}(\boldsymbol{x}^{(k)},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell_x},\boldsymbol{Z}^{(-k)}_{\text{obs}}\right]\right)^2}{N_{\text{H}}\mathbb{V}\left[z_{\text{H}}(D_{\text{H}},t_u)\right]}, \tag{4.15}$$

where $\mathbb{V}\left[z_{\text{H}}(D_{\text{H}},t_u)\right]$ is the empirical variance of the observed values:

$$\mathbb{V}\left[z_{\text{H}}(D_{\text{H}},t_u)\right]=\frac{1}{N_{\text{H}}}\sum_{k=1}^{N_{\text{H}}}z_{\text{H}}(\boldsymbol{x}^{(k)},t_u)^2-\left(\frac{1}{N_{\text{H}}}\sum_{k=1}^{N_{\text{H}}}z_{\text{H}}(\boldsymbol{x}^{(k)},t_u)\right)^2.$$

The procedure we propose starts with the dimension 0. For the case $N=0$ the surrogate model depends only on high-fidelity regression.

- We compute the surrogate model for all $N=0,\ldots,N_{\text{L}}$
- We calculate the mean in t_u of $Q^2_N(t_u)$:

$$\hat{Q}^2_N=\frac{1}{N_t}\sum_{u=1}^{N_t}Q^2_N(t_u)$$

We compare the $\hat{Q}^2_N$ values and the value N with the largest $\hat{Q}^2_N$ is chosen. In order to evaluate the surrogate model in the next section, we compute $\hat{Q}^2=\max_N\hat{Q}^2_N$.

5. Illustration: double pendulum simulator. The purpose of this section is to apply the methods proposed in the previous sections to a mechanical example. The example is based on a simulator of a pendulum attached to a spring-mass system. We have two codes: the high-fidelity code numerically solves Newton's equation. The low-fidelity code simplifies the equation, by linearisation for small angles of the pendulum motion, and solves the system.

5.1. Characteristics of the outputs.

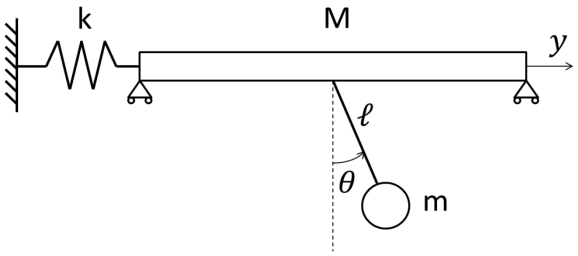

FIGURE 1. *The double pendulum system with its parameters.*

The physical system. The system can be seen as a dual-oscillator cluster. The first oscillator is a spring-mass system whose axis is perpendicular to the gravitational axis. The parameters of this system are the mass of the system M_S and the spring stiffness k. The initial position of the mass is denoted y_0, its initial velocity is 0. The second oscillator is a pendulum. A schematic representation of the system is presented in Figure 1. The parameters are the mass m and the length of the pendulum ℓ, which are fixed. The initial value of the angle is θ_0 and its derivative is $\dot{\theta_0}$. By Newton's law of motion, the dynamics is governed by a system of two coupled ordinary differential equations (ODEs). However, we do not have a closed form expression that gives the solution of the system. This forces us to use computer codes. The output signal is the position of the mass m at time $t \in \{t_1, \ldots, t_{N_t}\}$ with $N_t = 101$. The input vector is $\boldsymbol{x} = \left\{M_S, k, y_0, \theta_0, \dot{\theta_0}\right\}$. The input variables are assumed to be independent and identically distributed with uniform distributions as described in Table 1.

The two different code levels. We propose two codes. The high-fidelity code numerically solves the coupled system of ODEs by an Euler's derivation of the position y and the angle θ for each t_u. This gives functions $\theta(t_u)$ and $y(t_u)$. The low-fidelity code assumes that the angle of the θ pendulum is small so that the linearisation of $\sin(\theta)$ makes it possible to get a simpler form of the expression of the two coupled ODEs and a faster resolution.

TABLE 1
Distributions of the input variables.

M_S	k	θ_0	$\dot{\theta_0}$	y_0
$\mathcal{U}(10, 12)$	$\mathcal{U}(1; 1.4)$	$\mathcal{U}(\frac{\pi}{4}; \frac{\pi}{3})$	$\mathcal{U}(0; \frac{1}{10})$	$\mathcal{U}(0; 0.2)$

Code Analysis. A sensitivity analysis is carried out for information purposes, but it is not used in the forthcoming surrogate modeling. The sensitivity analysis makes it possible to determine the effective dimension of our problem. We compare outputs of the high- and low-fidelity codes and the associated Sobol indices on Figure 2. We estimate Sobol indices by the method described in [30] and implemented in the R library [12] by using a Monte Carlo sample of size 10^5 for each code. No surrogate model was used to estimate the indices in Figure 2. The main result is that the two codes depend on the same input variables. The four most important input variables are y_0, k, M and θ_0.

5.2. Comparison between methods. The experimental designs used to compare the methods are presented in [18]. They are constructed from two independent

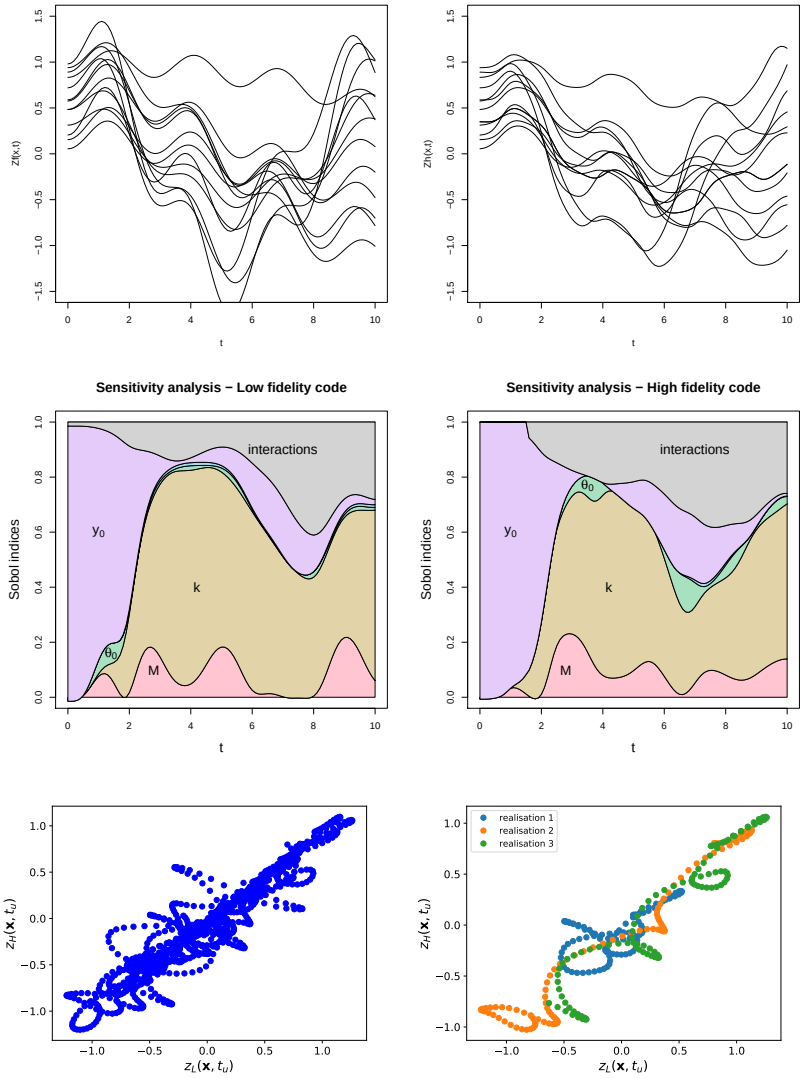

FIGURE 2. *Comparison between low-fidelity (top left) and high-fidelity (top right) code outputs. Sobol indices for high- and low-fidelity codes (center left: low-fidelity, center right: high-fidelity). For each time t in the time grid, we report the first-order Sobol indices and "interactions" stands for the sum of the Sobol indices of order larger than 2. Finally, the bottom plots represent the interactions between codes (plots of $(z_L(x,t_u), z_H(x,t_u))_{u=1}^{N_t}$ for different x). The bottom left graph is for 10 and the bottom right graph is for 3 values of x.*

maximum LHS designs with $N_{\mathrm{H}} = 10$ and $N_{\mathrm{L}} = 100$ points. The low-fidelity design is then modified so that the designs are nested. Only the points of the low-fidelity design closest to the points of the high fidelity design are moved. To generate these designs the R packages [7, 16] are used. A random uniform nested design can also be used, but we choose a more effective design for GP regression. The test design is composed of 4000 points randomly chosen in the hypercube determined by the supports of the uniform distributions described in Table 1.

In this section, we want to demonstrate the interest of the method presented in Section 4. For this we will compare several methods:

- the multi-fidelity method presented in Section 3 with a Dirac distribution of $\mathbf{\Gamma}$, called the SVD method.
- the multi-fidelity method that uses GP regression of the orthogonal part with covariance-tensorization and the distribution of $\mathbf{\Gamma}$ is Dirac at $\boldsymbol{\gamma}$ the matrix of the SVD of the observed low-fidelity code outputs. It prediction is computed as is Section 4 and called Dirac method.
- the multi-fidelity method presented in Section 4 with the CVB distribution, called CVB method.
- the neural network (NN) method presented in [22]. We extend this method for time-series outputs by considering N_t-dimensional outputs for the low- and high-fidelity neural networks and by removing the physical inspired NN part. We used the parameters proposed in the article, i.e. 2 hidden layers for each network with 20 neurons per layer. We also tested the NN method up to 100 neurons per layer but the best results were obtained with 20 neurons approximately.

The method we would like to highlight is the CVB method.

CVB basis. The law of $\mathbf{\Gamma}$ needs to be determined in order to compute or estimate the moments (3.10),(3.12), (4.12), and (D.12). The distribution of $\mathbf{\Gamma}$ is the CVB distribution described in Subsection 3.2.3. As shown by (3.7) it depends on the size k of the random subset I. Here we choose $k = 4$. Because it is too expensive to compute the sum over all $\binom{N_\mathrm{L}}{k}$ different subsets $\{j_1, \dots, j_k\}$, we estimate the expectation (3.7) by an empirical average over $n = 64$ realizations I_j of the random subset I:

$$\mathbb{E}\left[f(\mathbf{\Gamma})\right] \simeq \frac{1}{n}\sum_{j=1}^{n} f(\tilde{\boldsymbol{U}}_{I_j}), \tag{5.1}$$

where $\tilde{\boldsymbol{U}}_{I_j}$ is the matrix of the left singular vectors of the SVD of $\left(z_\mathrm{L}(\boldsymbol{x}^{(i)}, t_u)\right)_{\substack{u\in\{1,\dots,N_t\}\\ i\in\{1,\dots,N_\mathrm{L}\}\setminus I_j}}$. We have checked that the stability with respect to k is conserved if $1 < k < N_\mathrm{L} - N_\mathrm{H}$ and the stability with regard to n is valid if $n > \max(k, 50)$. We have tested the construction of the CVB basis for all k values in this range and found that changes in k do not influence the basis significantly.

The computational cost of calculating the basis is very important in particular because it is impossible for us to calculate it for all subsets. A method to compute the basis with only a cost of $O(N_t^2)$ is given in [23] whereas we compute it with $O(N_\mathrm{L}^2 N_t)$ by our method. The gain is however very small especially if $N_\mathrm{L} \ll N_t$ which is our case. We have therefore not implemented this method in the results presented in this paper.

Prediction of the orthogonal part. A simple model for the a priori mean function is chosen $M = 1$ and $f(\boldsymbol{x}) = 1$. Consequently, $F^T R_x^{-1} F = \sum_{i,j} \left\{R_x^{-1}\right\}_{i,j}$.

Multi-fidelity regression of the coefficients. Our implementation of the multi-fidelity regression is based on [16]. We use an informative prior for the regression of the coefficients. For more information refer to [18, Section 3.4.4]. In this example the size of the priors are $q = p_\mathrm{L} = p_\mathrm{H} = 1$. Considering the relation between the two codes we choose $b^\rho = 1$. The trend is supposed to be null consequently, $b_\mathrm{H}^\beta = b_\mathrm{L} = 0$. The variances are $\sigma_\mathrm{L} = 0.5$ and $\sigma_\mathrm{H} = 0.5$ with $V_\mathrm{H}^\beta = 2$ and $V_\mathrm{L} = 2$. The parameters for the inverse Gamma distribution $m_\mathrm{L} = m_\mathrm{H} = 0.2$ and $\varsigma_\mathrm{L} = \varsigma_\mathrm{H} = 1.5$. We have checked the robustness of the results with respect to the hyper-parameters of the

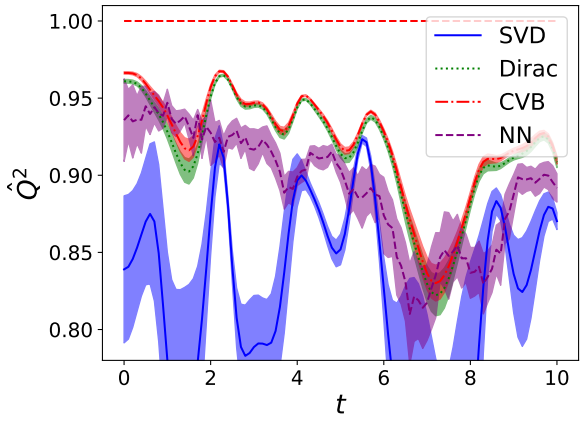

FIGURE 3. *Comparison between the methods in terms of time-dependent* $\hat{Q}^2$. *Averages over* 40 *random experimental designs are computed. The colored fields represent the confidence intervals determined by* ± 1.96 *empirical standard deviation. Here* $N_H = 10$, $N_L = 100$ *and* $N_t = 101$.

prior distributions. Alternatively, the article [21] presents non-informative priors for the autoregressive co-kriging.

Prediction. In order to estimate the errors of the surrogate models, we calculate their $\hat{Q}^2$'s and report them in Figure 3. To compute $\hat{Q}^2$ for a model, we calculate the difference between the validation set of size 4000 and the predictions of the model. We have averaged the estimates of the $\hat{Q}^2$ over 40 different experimental designs.

The SVD method gives a very interesting result because the $\hat{Q}^2$ is almost always higher than 0.8. However, in Figure 4 we can see that it does not capture the form of the times series. The $\hat{Q}^2$ of the Dirac and CVB methods are larger than the ones of the other methods. The error is also less variable as a function of t. And the variance is much lower for both methods. However, even if there is a difference between the Dirac and CVB methods, it is not possible to say that the CVB method is better in this application. The difference between the Dirac and CVB methods is small, in our example.

The variance of the prediction is very important for the quantification of prediction uncertainty. All formulas are given in the previous sections and we illustrate the results in Figure 4. We can see that the variance of the projection method is not accurate and overestimates the quality of the prediction. This method is not acceptable for prediction. The Dirac method and the CVB method have almost the same variance. If we compare to the variance of the SVD method, it means that most of the uncertainty relates to the orthogonal part. This leads to the conclusion that this part is important in the regression.

In order to understand the interest of the method with covariance tensorization for the orthogonal part we study in more detail the orthogonal part. First we study the role of the value of N. Here $N_{\mathrm{L}} = 100$, so the possible values of N are between 0 and 100. We find that the optimal value of N for 40 learning sets is between 8 and 10. Even when the value of N_{H} is increased, N remains constant in the 8 to 10 range. This means that the low-fidelity code can give reliable information on the high-fidelity code output projection into a 8-dimensional space. The high-fidelity code output is, however, higher dimensional and it is important to predict the orthogonal part with a dedicated method, namely the proposed covariance tensorization method.

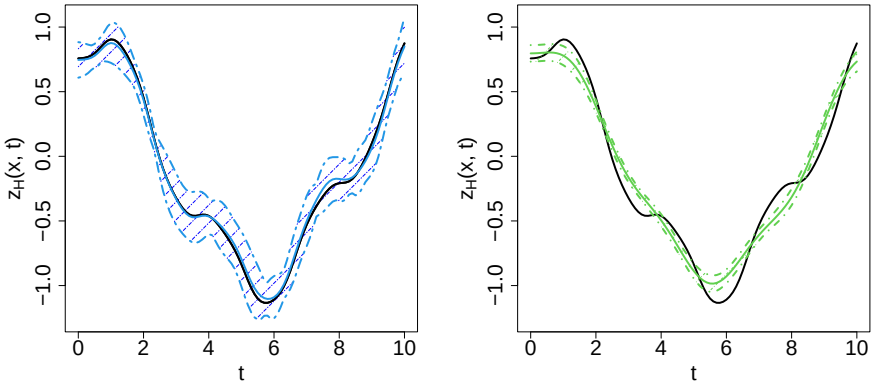

FIGURE 4. *Comparison between the predictions of the CVB method (left) and the SVD method (right). The black solid line is the exact high-fidelity time series, the colored solid line is the prediction mean and the dashed lines are the confidence intervals. In this example the value of N obtained by cross validation is* 8.

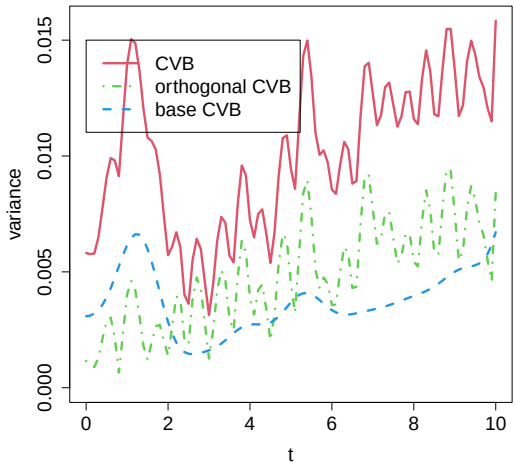

FIGURE 5. *Estimation of the different time-dependent prediction variance terms for the empirical method.*

We have carried out extensive numerical simulations with values of N_H in the range $[5:20]$ and values of N_L in the range $[50:1000]$. If only very small data sets are available ($5 \leq N_H \leq 7$) the prediction is not satisfactory whatever the method. Moreover, for values of N_L greater than 200, there is no significant change except for the NN method which improves to the level of the CVB method for the largest data sets. There is also a decrease in prediction uncertainty with the increase of the N_H number as can be expected. At the same time as the prediction uncertainty increases when N_H is decreased, there is also a decrease in prediction performance, but independently of the method. The code we used is available in [14].

6. Discussion. The objective of this work is to propose a method that generates a surrogate model in the context of multi-fidelity and time series outputs and that quantifices the prediction uncertainty. The method we propose is based on three main ingredients: dimension reduction, co-kriging, and covariance tensorization. The model we present is based on multi-fidelity (co-kriging) regression. By reducing the output dimension, multi-fidelity regression becomes possible. To take into account all the information contained in the data sets, the part that cannot be treated with the previous method is predicted by Gaussian process regression with covariance tensorization.

First, we have presented different ways to build the basis that allows to represent the high- and low-fidelity code outputs. Second, we have presented a model that allows to estimate the high-fidelity code outputs from data collected from the high- and low-fidelity codes. The combination of a multi-fidelity part and a single-fidelity part with tensorized covariance is the central point of the proposed method. The performance of our model has been tested on a mechanical example. We have been able to use multi-fidelity in a very convincing way to build a robust surrogate model better than any other method presented so far.

There are several ways to extend the method presented in this article. Sequential experimental designs in a multi-fidelity context have already been dealt with by [19]. However, they deserve to be extended to the case of time-series outputs. We can consider regression problems for more than two levels of code. It is conceivable in this case to build several levels of bases which from code to code would improve the basis and thus reduce the orthogonal part. In addition, high-dimensional outputs are not different from time-series outputs as considered in this paper. It is therefore conceivable to adapt this method to more general functional outputs.

Appendix A. Multi-fidelity priors for AR(1) model. In the following we define the priors needed to use the AR(1) model defined in subsection 2.1.

The goal of a Bayesian prediction is to integrate the uncertainty of the parameter estimation into the predictive distribution as in [17]. Here the parameters are σ, β and β_ρ. As explained in [20] the result is not Gaussian but we can obtain expressions of the posterior mean $\mathbb{E}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$ and variance $\mathbb{V}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$. It is possible to consider informative or non informative priors for the parameters [20, 21]. Here we consider informative conjugate priors:

$$\left[\sigma_{\mathrm{L}}^2\right] \sim \mathcal{IG}\left(m_{\mathrm{L}}, \varsigma_{\mathrm{L}}\right), \tag{A.1}$$

$$\left[\beta_{\mathrm{L}}|\sigma_{\mathrm{L}}^2\right] \sim \mathcal{N}_{p_{\mathrm{L}}}\left(b_{\mathrm{L}}, \sigma_{\mathrm{L}}^2 V_{\mathrm{L}}\right), \tag{A.2}$$

$$\left[\sigma_{\mathrm{H}}^2|\mathcal{A}^{\mathrm{L}}=\alpha^{\mathrm{L}}, \beta_{\mathrm{L}}, \sigma_{\mathrm{L}}\right] \sim \mathcal{IG}\left(m_{\mathrm{H}}, \varsigma_{\mathrm{H}}\right), \tag{A.3}$$

$$\left[\beta_\rho, \beta_{\mathrm{H}}|\mathcal{A}^{\mathrm{L}}=\alpha^{\mathrm{L}}, \sigma_{\mathrm{H}}^2, \beta_{\mathrm{L}}, \sigma_{\mathrm{L}}\right] \sim \mathcal{N}_{q+p_{\mathrm{H}}}\left(b_{\mathrm{H}}=\begin{pmatrix} b^\rho \\ b_{\mathrm{H}}^\beta \end{pmatrix}, \sigma_{\mathrm{H}}^2 V_{\mathrm{H}}=\sigma_{\mathrm{H}}^2 \begin{pmatrix} V^\rho & 0 \\ 0 & V_{\mathrm{H}}^\beta \end{pmatrix}\right), \tag{A.4}$$

with

- b_{L} a vector of size p_{L},
- b^ρ a vector of size q,
- b_{H}^β a vector of size p_{H},
- V_{H}^β a $p_{\mathrm{H}} \times p_{\mathrm{H}}$ matrix,
- V^ρ a $q \times q$ matrix,
- V_{L} a $p_{\mathrm{L}} \times p_{\mathrm{L}}$ matrix,

- m_{F} and ς_{F} are positive real numbers and $\mathcal{IG}$ stands for the inverse Gamma distribution.

By using these informative conjugate priors we obtain the following a posteriori distributions as in [20] :

$$\left[\sigma_{\mathrm{L}}^2|\mathcal{A}^{\mathrm{L}}=\alpha^{\mathrm{L}}\right] \sim \mathcal{IG}\left(d_{\mathrm{L}},Q_{\mathrm{L}}\right), \tag{A.5}$$

$$\left[\beta_{\mathrm{L}}|\mathcal{A}^{\mathrm{L}}=\alpha^{\mathrm{L}},\sigma_{\mathrm{L}}^2\right] \sim \mathcal{N}_{p_{\mathrm{L}}}\left(\Sigma_{\mathrm{L}}\nu_{\mathrm{L}},\Sigma_{\mathrm{L}}\right), \tag{A.6}$$

$$\left[\sigma_{\mathrm{H}}^2|\mathcal{A}=\alpha\right] \sim \mathcal{IG}\left(d_{\mathrm{H}},Q_{\mathrm{H}}\right), \tag{A.7}$$

$$\left[\beta_{\mathrm{H}},\beta_{\rho}|\mathcal{A}=\alpha,\sigma_{\mathrm{H}}^2\right] \sim \mathcal{N}_{p_{\mathrm{H}}+q}\left(\Sigma_{\mathrm{H}}\nu_{\mathrm{H}},\Sigma_{\mathrm{H}}\right), \tag{A.8}$$

with:

- $d_{\mathrm{F}}=\frac{n_{\mathrm{F}}}{2}+m_{\mathrm{F}}$,
- $\tilde{Q}_{\mathrm{F}}=\left(\alpha^{\mathrm{F}}-H_{\mathrm{F}}\hat{\lambda}_{\mathrm{F}}\right)^T C_{\mathrm{F}}^{-1}\left(\alpha^{\mathrm{F}}-H_{\mathrm{F}}\hat{\lambda}_{\mathrm{F}}\right)$,
- $Q_{\mathrm{F}}=\tilde{Q}_{\mathrm{F}}+\varsigma_{\mathrm{F}}+\left(b_{\mathrm{F}}-\hat{\lambda}_{\mathrm{F}}\right)^T\left(V_{\mathrm{F}}+\left(H_{\mathrm{F}}^T C_{\mathrm{F}}^{-1}H_{\mathrm{F}}\right)\right)^{-1}\left(b_{\mathrm{F}}-\hat{\lambda}_{\mathrm{F}}\right)$,
- $\Sigma_{\mathrm{F}}=\left[H_{\mathrm{F}}^T\frac{C_{\mathrm{F}}^{-1}}{\sigma_{\mathrm{F}}^2}H_{\mathrm{F}}+\frac{V_{\mathrm{F}}^{-1}}{\sigma_{\mathrm{F}}^2}\right]^{-1}$,
- $\nu_{\mathrm{F}}=\left[H_{\mathrm{F}}^T\frac{C_{\mathrm{F}}^{-1}}{\sigma_{\mathrm{F}}^2}\alpha^{\mathrm{F}}+\frac{V_{\mathrm{F}}^{-1}}{\sigma_{\mathrm{F}}^2}b_{\mathrm{F}}\right]$,
- H_{F} is defined by $H_{\mathrm{L}}=F_{\mathrm{L}}$ and $H_{\mathrm{H}}=\left[G^{\mathrm{L}}\odot\left(\alpha^{\mathrm{H}}\mathbf{1}_{q_{\mathrm{L}}}^T\right)\ F_{\mathrm{H}}\right]$,
- G^{L} is the $N_{\mathrm{H}}\times q$ matrix containing the values of $g_{\mathrm{L}}^T(\boldsymbol{x})$ for $\boldsymbol{x}\in D_{\mathrm{H}}$,
- $\mathbf{1}_{q_{\mathrm{L}}}$ is a q-dimensional vector containing 1,
- $\hat{\lambda}_{\mathrm{F}}=\left(H_{\mathrm{F}}^T C_{\mathrm{F}}^{-1}H_{\mathrm{F}}\right)^{-1}H_{\mathrm{F}}^T C_{\mathrm{F}}^{-1}\alpha^{\mathrm{F}}$.

Consequently, the posterior distribution of $A_{\mathrm{H}}(\boldsymbol{x})$ has the following mean and variance:

$$\begin{aligned}\mathbb{E}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right] =& h_{\mathrm{H}}^T(\boldsymbol{x})\Sigma_{\mathrm{H}}\nu_{\mathrm{H}}+r_{\mathrm{H}}^T(\boldsymbol{x})C_{\mathrm{H}}^{-1}\left(\alpha^{\mathrm{H}}-H_{\mathrm{H}}\Sigma_{\mathrm{H}}\nu_{\mathrm{H}}\right),\\ \mathbb{V}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right] =& \left(\hat{\rho}_{\mathrm{L}}^2(\boldsymbol{x})+\varepsilon_{\rho}(\boldsymbol{x})\right)\sigma_{\tilde{A}_{\mathrm{L}}}^2(\boldsymbol{x})+\frac{Q_{\mathrm{H}}}{2(d_{\mathrm{H}}-1)}(1-r_{\mathrm{H}}^T(\boldsymbol{x})C_{\mathrm{H}}^{-1}r_{\mathrm{H}}(\boldsymbol{x}))\\ &+\left(h_{\mathrm{H}}^T-r_{\mathrm{H}}^T(\boldsymbol{x})C_{\mathrm{H}}^{-1}H_{\mathrm{H}}\right)\Sigma_{\mathrm{H}}\left(h_{\mathrm{H}}^T-r_{\mathrm{H}}^T(\boldsymbol{x})C_{\mathrm{H}}^{-1}H_{\mathrm{H}}\right)^T,\end{aligned} \tag{A.9}$$

$$\tag{A.10}$$

with $\hat{\rho}_{\mathrm{L}}(\boldsymbol{x})=g_{\mathrm{L}}^T(\boldsymbol{x})\hat{\beta}_{\rho}$, $\hat{\beta}_{\rho}=[\Sigma_{\mathrm{H}}\nu_{\mathrm{H}}]_{i=p_{\mathrm{H}}+1,\ldots,p_{\mathrm{H}}+q}$ and $\varepsilon_{\rho}(\boldsymbol{x})=g_{\mathrm{L}}^T(\boldsymbol{x})\tilde{\Sigma}_{\mathrm{H}}g_{\mathrm{L}}(\boldsymbol{x})$ with $\tilde{\Sigma}_{\mathrm{H}}=[\Sigma_{\mathrm{H}}]_{i,j=p_{\mathrm{H}}+1,\ldots,p_{\mathrm{H}}+q}$.

The posterior mean $\mathbb{E}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$ is the predictive model of the high fidelity response and the posterior variance $\mathbb{V}\left[A_{\mathrm{H}}(\boldsymbol{x})|\mathcal{A}=\alpha\right]$ represents the predictive variance of the model.

Appendix B. LOO formula and discussion.

LOO without loop. In order to quickly minimize the LOO error with respect to the vector of correlation lengths $\boldsymbol{\ell_x}$ there exist formulas to evaluate $\varepsilon^2(\boldsymbol{\ell_x})$ with matrix products [2],[6]. The LOO optimization problem is equivalent to minimize a function $f_{\mathbf{CV}}(\boldsymbol{\ell_x})$ given by:

$$f_{\mathbf{CV}}(\boldsymbol{\ell_x})=z^T R_{\boldsymbol{\ell_x}}^{-1}\,\mathrm{diag}\left(R_{\boldsymbol{\ell_x}}^{-1}\right)^{-2}R_{\boldsymbol{\ell_x}}^{-1}z, \tag{B.1}$$

where z is a vector that collect all the data.

Considering Equation (2.11) and the mixed-product property, the inverse of a Kronecker product and the formula $\operatorname{diag}(A \otimes B) = \operatorname{diag} A \otimes \operatorname{diag} B$ the cost function can be expressed as:

$$f_{\mathbf{CV}}(\boldsymbol{\ell_x}) = z^T \left(R_t^{-1} \operatorname{diag}\left(R_t^{-1}\right)^{-2} R_t^{-1} \right) \otimes \left(R_x^{-1} \operatorname{diag}\left(R_x^{-1}\right)^{-2} R_x^{-1} \right) z. \tag{B.2}$$

However, the term in R_t is impossible to calculate because R_t is not invertible. $R_t^{-1} \operatorname{diag}\left(R_t^{-1}\right)^{-2} R_t^{-1}$ can be approximated by I_{N_t} in order to have a tractable problem. This assertion is equivalent to the hypothesis:

$$R_t^2 = \operatorname{diag}(R_t^{-1})^{-2}. \tag{B.3}$$

This assumption can be seen as the fact that the error is estimated by taking into account only the spatial distribution of the covariance. Indeed, to calculate the error only the matrix R_x is used, even if the value of R_t is calculated by maximum likelihood later in the GP regression method.

Thus, the minimization described in Equation (2.19) makes it possible to calculate the correlation lengths by minimizing:

$$f_{\mathbf{CV}}(\boldsymbol{\ell_x}) \simeq z^T I_{N_t} \otimes \left(R_x^{-1} \operatorname{diag}\left(R_x^{-1}\right)^{-2} R_x^{-1} \right) z, \tag{B.4}$$

where I_{N_t} is the $N_t \times N_t$ identity matrix. The main interest of this method is to give an approximate value of the error and to make the optimization much faster.

Optimization with hypothesis (B.3). Efficient minimization algorithms require to have the derivative of the function so that it does not have to be calculated by finite differences. Thanks to the simplification (B.4) it is possible to calculate the derivative of the LOO error [2]:

$$\begin{aligned} \frac{\partial}{\partial \ell_{x_k}} f_{\mathbf{CV}}(\boldsymbol{\ell_x}) = 2 z^T I_{N_t} \otimes R_x^{-1} \operatorname{diag}\left(R_x^{-1}\right)^{-2} \left(R_x^{-1} \frac{\partial R_x}{\partial \ell_{x_k}} R_x^{-1} \right) \operatorname{diag}\left(R_x^{-1}\right)^{-1} R_x^{-1} z \\ - 2 z^T I_{N_t} \otimes R_x^{-1} \operatorname{diag}\left(R_x^{-1}\right)^{-2} R_x^{-1} \frac{\partial R_x}{\partial \ell_{x_k}} R_x^{-1} z \end{aligned}, \tag{B.5}$$

with

$$\left(\frac{\partial R_{x,l}}{\partial \ell_{x_k}} \right)_{i,j} = \frac{\ell_{x_k} (x_{k,j} - x_{k,i})^2}{|x_{k,j} - x_{k,i}|} h'_{\frac{5}{2}} \left(\frac{|x_{k,j} - x_{k,i}|}{\ell_{x_k}} \right) \tag{B.6}$$

and

$$h'_{\frac{5}{2}}(x) = -\frac{5}{3} x \left(1 + \sqrt{5}x\right) \exp\left(-\sqrt{5}x\right). \tag{B.7}$$

The method used to calculate the value of $\boldsymbol{\ell_x}$ is the Nelder-Mead method with only one starting point, because starting from more points will be more costly and the function $f_{\mathbf{CV}}$ is close to quadratic consequently does not need multiple starting points.

Optimization without hypothesis (B.3):. When hypothesis (B.3) does not hold, a way must be found to calculate $\boldsymbol{\ell_x}$ without this assumption. By a regularization of the matrix R_t it is possible to calculate $f_{\mathbf{CV}}(\boldsymbol{\ell_x})$ and its derivative by Equations (2.19),

	Loop LOO	Full LOO	Simplified LOO
$\varepsilon_{\mathcal{Q}^\in}$	$7.06\ 10^{-4}$	$8.72\ 10^{-4}$	$8.14\ 10^{-4}$
time	18.41 s	3.47 min	0.17 s

TABLE 2

Benchmark of the different LOO optimization techniques for estimating the $f(\boldsymbol{x}, t) = \cos(4\pi(x_2+1)t)\sin(3\pi x_1 t)$ *function using the separable covariance method. Loop LOO processes the error by computing the approximation, Full LOO processes the regularized analytic expression and Simplified LOO processes the simplified one given by Equation (B.4).*

(B.2), and (B.8). However, the solution will be a regularized solution and not an exact solution.

There are different types of regularization that allow matrices to be inverted. Two methods have been investigated here. The first one is standard (Tilkonov regularization):

$$\widehat{R_t^{-1}} = \left(R_t^T R_t + \varepsilon^2 I_{N_t}\right)^{-1} R_t^T. \tag{B.8}$$

The second one is:

$$\widehat{R_t^{-1}} = \left(R_t + \varepsilon I_{N_t}\right)^{-1}. \tag{B.9}$$

It has the disadvantage of being more sensitive to ε than the first one, which is why it will not be used.

However, in the calculation of the determinant, this adjustment may have advantages. Denoting by $\widehat{R_t^{-1}} = V\Sigma^{-1}U^T$ the SVD of R_t, with $\Sigma^{-1} = \operatorname{diag}\frac{1}{\sigma_i+\varepsilon}$ whereas the same decomposition gives for Equation (B.8) $\Sigma^{-1} = \operatorname{diag}\frac{\sigma_i}{\sigma_i^2+\varepsilon^2}$. This is the reason why the two adjustments presented are not used in the same case. Indeed $\frac{1}{\sigma_i+\varepsilon}$ is less efficient for the calculation of a determinant but more efficient for the calculation of the inverse of $\widehat{R_t}$.

$$\begin{aligned}\frac{\partial}{\partial \ell_{x_k}} f_{\mathbf{CV}}(\boldsymbol{\ell_x}) = 2z^T &\left(\widehat{R_t^{-1}}\operatorname{diag}\left(\widehat{R_t^{-1}}\right)^{-2}\widehat{R_t^{-1}}\right)\\ \otimes R_x^{-1}\operatorname{diag}\left(R_x^{-1}\right)^{-2}&\left(R_x^{-1}\frac{\partial R_x}{\partial \ell_{x_k}}R_x^{-1}\right)\operatorname{diag}\left(R_x^{-1}\right)^{-1}R_x^{-1}z.\\ -2z^T\left(\widehat{R_t^{-1}}\operatorname{diag}\left(\widehat{R_t^{-1}}\right)^{-2}\widehat{R_t^{-1}}\right)&\otimes R_x^{-1}\operatorname{diag}\left(R_x^{-1}\right)^{-2}R_x^{-1}\frac{\partial R_x}{\partial \ell_{x_k}}R_x^{-1}z\end{aligned} \tag{B.10}$$

Equation (B.6) and Equation (B.7) are still valid.

This complete approach was compared to the LOO calculation using a loop. However, the calculation time of Kronecker products is too long compared to the calculation of the simple error with a loop. Moreover, the differences in the errors of the different methods are negligible. Thus this solution is only recommended when the calculation of Equations (B.2) and (B.10) is optimized.

Table 2 shows that the gain in calculation time by the simplified method is significant even though the error difference is very small. The extremely long time for the complete LOO is mainly due to an implementation of the Kronecker product that is not very effective in our implementation.

Appendix C. Tensorized covariance of the orthogonal part. For R_x we assume that C_x is chosen in the Matérn-5/2 class of functions. The function only

depends on the correlation length vector $\boldsymbol{\ell_x}$. The matrix R_t is estimated as described in Subsection 2.2 by the matrix $\widehat{R_t}$ given by:

$$\widehat{R_t} = \frac{1}{N_x}\left(\boldsymbol{Z}^{\perp}_{\text{obs}} - \hat{\boldsymbol{Z}}^{\perp}\right) R_x^{-1} \left(\boldsymbol{Z}^{\perp}_{\text{obs}} - \hat{\boldsymbol{Z}}^{\perp}\right)^T,$$

where $\hat{\boldsymbol{Z}}^{\perp}$ is the $N_t \times N_x$ matrix of empirical means $\hat{Z}^{\perp}_{u,i} = \frac{1}{N_x}\sum_{j=1}^{N_x}\left(\boldsymbol{Z}^{\perp}_{\text{obs}}\right)_{u,j}$, $\forall i = 1,\dots,N_x$ and $u = 1,\dots,N_t$. Its range is indeed in $S_N^{\perp}$.

The prediction mean is the sum of two terms, $\boldsymbol{Z}^{\perp}_{\text{obs}} R_x^{-1} r_x(\boldsymbol{x})$ which is $S_N^{\perp}$-valued and $B_\star u(\boldsymbol{x})$ also $S_N^{\perp}$-valued because:

$$B_\star = \boldsymbol{Z}^{\perp}_{\text{obs}} R_x^{-1} F \left(F^T R_x^{-1} F\right)^{-1}, \tag{C.1}$$

with F the $N_{\text{F}} \times M$ matrix $\left[f^T(\boldsymbol{x}^{(i)})\right]_{i=1,\dots,N_{\text{F}}}$. Consequently, we have:

$$Z^{\perp}_{\text{H}}(\boldsymbol{x}, t_u)|\boldsymbol{\ell_x}, \boldsymbol{\Gamma}, N, \boldsymbol{Z}^{\perp}_{\text{obs}} \sim \mathcal{GP}(\mu_\star(\boldsymbol{x}), R_\star(\boldsymbol{x}, \boldsymbol{x}')) \tag{C.2}$$

where the mean is given by (2.13) and the covariance by (2.14) with $\boldsymbol{Z}^{\perp}_{\text{obs}}$ as the observed inputs. LOO estimation of the vector of correlation lengths $\boldsymbol{\ell_x}$ given $\boldsymbol{\Gamma}$ and N is carried out by the method presented in Appendix B.

Appendix D. Expressions of some expectations and variances.

D.1. Computation for uncorrolated Gaussian Processes. Given $\boldsymbol{\Gamma}$, $\left(A_{i,\text{H}}(\boldsymbol{x}, t_u), A_{i,\text{L}}(\boldsymbol{x}, t_u)\right)_{\substack{\boldsymbol{x}\in Q \\ u=1,\dots,N_t}}$ are independent with respect to i. This independence makes it possible to generate N_t independent surrogate models, with mean and variance given by (A.9) and (A.10):

$$\begin{aligned}
\mathbb{E}\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}, \mathcal{A} = \alpha\right] =& h^T_{i,\text{H}}(\boldsymbol{x})\Sigma_{i,\text{H}}\nu_{i,\text{H}} + r^T_{i,\text{H}}(\boldsymbol{x}) C^{-1}_{i,\text{H}}\left(\alpha^{\text{H}}_i - H_{i,\text{H}}\Sigma_{i,\text{H}}\nu_{i,\text{H}}\right), \qquad \text{(D.1)}\\
\mathbb{V}\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}, \mathcal{A} = \alpha\right] =& \left(\hat{\rho}^2_{i,\text{L}}(\boldsymbol{x}) + \varepsilon_{i,\rho}(\boldsymbol{x})\right)\sigma^2_{\tilde{A}_{\text{L},i}}(\boldsymbol{x})\\
&+ \frac{Q_{i,\text{H}}}{2(d_{i,\text{H}} - 1)}(1 - r^T_{i,\text{H}}(\boldsymbol{x}) C^{-1}_{i,\text{H}} r_{i,\text{H}}(\boldsymbol{x}))\\
&+ \left(h^T_{i,\text{H}}(\boldsymbol{x}) - r^T_{i,\text{H}}(\boldsymbol{x}) C^{-1}_{i,\text{H}} H_{i,\text{H}}\right)\Sigma_{i,\text{H}}\left(h^T_{i,\text{H}}(\boldsymbol{x}) - r^T_{i,\text{H}}(\boldsymbol{x}) C^{-1}_{i,\text{H}} H_{i,\text{H}}\right)^T. \qquad \text{(D.2)}
\end{aligned}$$

D.2. Variance for projection. The law of total variance gives :

$$\mathbb{V}_\alpha\left[Z_{\text{H}}(\boldsymbol{x}, t_u)\right] = \mathbb{V}_\alpha\left[\mathbb{E}_\alpha\left[Z_{\text{H}}(\boldsymbol{x}, t_u)|\boldsymbol{\Gamma}\right]\right] + \mathbb{E}_\alpha\left[\mathbb{V}_\alpha\left[Z_{\text{H}}(\boldsymbol{x}, t_u)|\boldsymbol{\Gamma}\right]\right] \tag{D.3}$$

The variance term can be expressed as follows :

$$\begin{aligned}
&\mathbb{V}_\alpha\left[\mathbb{E}_\alpha\left[Z_{\text{H}}(\boldsymbol{x}, t_u)|\boldsymbol{\Gamma}\right]\right] = \mathbb{V}_\alpha\left[\textstyle\sum_{i=1}^{N_t}\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\
&= \textstyle\sum_{i=1}^{N_t}\mathbb{V}_\alpha\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\
&+ \textstyle\sum_{i,j=1,i\neq j}^{N_t}\text{Cov}_\alpha\left(\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \Gamma_j(t_u)\mathbb{E}_\alpha\left[A_{j,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right)
\end{aligned}, \tag{D.4}$$

where $\mathbb{E}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]$ is given by Equation (D.1). The expectation term can be expressed as :

(D.5)

$$\begin{aligned}
&\mathbb{E}_\alpha\left[\mathbb{V}_\alpha\left[Z_{\text{H}}(\boldsymbol{x}, t_u)|\boldsymbol{\Gamma}\right]\right] =\\
&\mathbb{E}_\alpha\left[\textstyle\sum_{i=1}^{N_t}\mathbb{V}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})\Gamma_i(t_u)|\boldsymbol{\Gamma}\right] + \sum_{i,j=1,i\neq j}^{N_t}\text{Cov}_\alpha\left(A_{i,\text{H}}(\boldsymbol{x})\Gamma_i(t_u), A_{j,\text{H}}(\boldsymbol{x})\Gamma_j(t_u)|\boldsymbol{\Gamma}\right)\right]\\
&= \textstyle\sum_{i=1}^{N_t}\mathbb{E}_\alpha\left[\Gamma_i^2(t_u)\mathbb{V}_\alpha\left[A_{i,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\\
&+ \textstyle\sum_{i,j=1,i\neq j}^{N_t}\mathbb{E}_\alpha\left[\Gamma_i(t_u)\Gamma_j(t_u)\text{Cov}_\alpha\left(A_{i,\text{H}}(\boldsymbol{x}), A_{j,\text{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right)\right]
\end{aligned},$$

where $\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u)|\boldsymbol{\Gamma}\right]$ is given in Equation (D.2) and $\mathrm{Cov}_\alpha(A_{i,\mathrm{H}}(\boldsymbol{x}), A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}) = 0$ if $i \neq j$. Consequently:

$$\begin{array}{c}\mathbb{V}_\alpha\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)\right] = \sum_{i=1}^{N_t}\mathbb{V}_\alpha\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right] \\ +\sum_{i,j=1,i\neq j}^{N_t}\mathrm{Cov}_\alpha(\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \Gamma_j(t_u)\mathbb{E}_\alpha\left[A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]) \\ +\sum_{i=1}^{N_t}\mathbb{E}_\alpha\left[\Gamma_i^2(t_u)\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right]\end{array} . \tag{D.6}$$

D.3. Variance for tensorisation of covariance and projection. The theorem of the total variance gives us:

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = \mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]\end{array} . \tag{D.7}$$

The two terms of (4.13) are:

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = \\ \mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] + \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\parallel}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N\right]|N\right]\end{array} , \tag{D.8}$$

and

$$\begin{array}{c}\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = \\ \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[2\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\parallel}\left(\boldsymbol{x},t_u\right), Z_{\mathrm{H}}^{\perp}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_\alpha\left[Z_{\mathrm{H}}^{\parallel}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N\right]|N\right].\end{array} \tag{D.9}$$

The uncorrelation of the $A_{i,\mathrm{H}}(\boldsymbol{x},t_u)$ coefficients given $\boldsymbol{\Gamma}$ gives $\mathrm{Cov}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x}), A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right] = 0$ for $i \neq j$ and $\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[A_{i,\mathrm{H}}(\boldsymbol{x})\Gamma_i(t_u), Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma}\right] = 0$. This leads us to simplify Equations (D.8) and (D.9) into:

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]\right] = \mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_\alpha\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\sum_{i=1}^{N}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right] \\ +\sum_{i,j=1;i\neq j}^{N}\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \Gamma_j(t_u)\mathbb{E}_\alpha\left[A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right] \\ +2\sum_{i=1}^{N}\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{l}_{\boldsymbol{x}}\right]\end{array} , \tag{D.10}$$

and

$$\begin{array}{c}\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]\right] = \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)^2\mathbb{V}_\alpha\left[A_{i,\mathrm{H}}\left(\boldsymbol{x}\right)|\boldsymbol{\Gamma}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]\end{array} . \tag{D.11}$$

The full formula of the variance can be expressed as :

$$\begin{array}{c}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}\left(\boldsymbol{x},t_u\right)|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] = \mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\mathbb{V}_\alpha\left[Z_{\mathrm{H}}^{\perp}\left(\boldsymbol{x},t_u\right)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right] \\ +\sum_{i=1}^{N}\mathbb{V}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right] \\ +\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)^2\mathbb{V}_\alpha\left[A_i\left(\boldsymbol{x}\right)|\boldsymbol{\Gamma}\right]\right] \\ +\sum_{i,j=1;i\neq j}^{N}\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \Gamma_j(t_u)\mathbb{E}_\alpha\left[A_{j,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]\right] \\ +2\sum_{i=1}^{N}\mathrm{Cov}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[\Gamma_i(t_u)\mathbb{E}_\alpha\left[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}\right], \mathbb{E}_{\boldsymbol{Z}_{\mathrm{obs}}}\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x},t_u)|\boldsymbol{\Gamma},N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]|N,\boldsymbol{\ell}_{\boldsymbol{x}}\right]\end{array} . \tag{D.12}$$

where $\mathbb{V}_\alpha[A_{i,\mathrm{H}}(\boldsymbol{x})|\boldsymbol{\Gamma}]$ is given by Equation (D.2) and $\mathbb{V}_\alpha\left[Z_{\mathrm{H}}^{\perp}(\boldsymbol{x})|\boldsymbol{\Gamma}\right]$ is given by Equation (2.14).

Acknowledgments. The author thanks Pr. Josselin Garnier and Dr. Claire Cannamela for their guidance and advice.

REFERENCES

[1] G. Aversano, A. Bellemans, Z. Li, A. Coussement, O. Gicquel, and A. Parente, *Application of reduced-order models based on PCA & kriging for the development of digital twins of reacting flow applications*, Computers & Chemical Engineering, 121 (2019), pp. 422–441.

[2] F. Bachoc, *Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification*, Computational Statistics & Data Analysis, 66 (2013), pp. 55–69.

[3] S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan, *Gaussian process emulation of dynamic computer codes*, Biometrika, 96 (2009), pp. 663–676.

[4] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, *Deep Gaussian processes for multi-fidelity modeling*, arXiv preprint arXiv:1903.07320v1, (2019).

[5] P. Diaconis, *What is a random matrix*, Notices of the AMS, 52 (2005), pp. 1348–1349.

[6] O. Dubrule, *Cross validation of kriging in a unique neighborhood*, Journal of the International Association for Mathematical Geology, 15 (1983), pp. 687–699.

[7] D. Dupuy, C. Helbert, and J. Franco, *DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments*, Journal of Statistical Software, 65 (2015), pp. 1–38, http://www.jstatsoft.org/v65/i11/.

[8] A. I. Forrester, A. Sóbester, and A. J. Keane, *Multi-fidelity optimization via surrogate modelling*, Proceedings of the royal society a: mathematical, physical and engineering sciences, 463 (2007), pp. 3251–3269.

[9] M. Giselle Fernández-Godino, C. Park, N. H. Kim, and R. T. Haftka, *Issues in deciding whether to use multifidelity surrogates*, AIAA Journal, 57 (2019), pp. 2039–2054.

[10] J. Goh, D. Bingham, J. P. Holloway, M. J. Grosskopf, C. C. Kuranz, and E. Rutter, *Prediction and computer model calibration using outputs from multifidelity simulators*, Technometrics, 55 (2013), pp. 501–512.

[11] O. Grujic, A. Menafoglio, G. Yang, and J. Caers, *Cokriging for multivariate Hilbert space valued random fields: application to multi-fidelity computer code emulation*, Stochastic Environmental Research and Risk Assessment, 32 (2018), pp. 1955–1971.

[12] B. Iooss, S. D. Veiga, A. Janon, G. Pujol, with contributions from Baptiste Broto, K. Boumhaout, T. Delage, R. E. Amri, J. Fruth, L. Gilquin, J. Guillaume, L. Le Gratiet, P. Lemaitre, A. Marrel, A. Meynaoui, B. L. Nelson, F. Monari, R. Oomen, O. Rakovec, B. Ramos, O. Roustant, E. Song, J. Staum, R. Sueur, T. Touati, and F. Weber, *sensitivity: Global Sensitivity Analysis of Model Outputs*, 2020, https://CRAN.R-project.org/package=sensitivity. R package version 1.22.1.

[13] M. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, Biometrika, 87 (2000), pp. 1–13.

[14] B. Kerleguer, *MultiFi Time-Series*, 2022, https://github.com/ehbihenscoding/MultiFiTimeSeries.

[15] D. G. Krige, *A statistical approach to some basic mine valuation problems on the witwatersrand*, Journal of the Southern African Institute of Mining and Metallurgy, 52 (1951), pp. 119–139.

[16] L. Le Gratiet, *MuFiCokriging: Multi-Fidelity Cokriging models*, 2012, https://CRAN.R-project.org/package=MuFiCokriging. R package version 1.2.

[17] L. Le Gratiet, *Bayesian analysis of hierarchical multifidelity codes*, SIAM/ASA Journal on Uncertainty Quantification, 1 (2013), pp. 244–269.

[18] L. Le Gratiet, *Multi-fidelity Gaussian process regression for computer experiments*, phdthesis, Université Paris-Diderot - Paris VII, Oct. 2013, https://tel.archives-ouvertes.fr/tel-00866770.

[19] L. Le Gratiet and C. Cannamela, *Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes*, Technometrics, 57 (2015), pp. 418–427.

[20] L. Le Gratiet and J. Garnier, *Recursive co-kriging model for design of computer experiments with multiple levels of fidelity*, International Journal for Uncertainty Quantification, 4 (2014), pp. 364–386.

[21] P. Ma, *Objective bayesian analysis of a cokriging model for hierarchical multifidelity codes*, SIAM/ASA Journal on Uncertainty Quantification, 8 (2020), pp. 1358–1382.

[22] X. Meng and G. E. Karniadakis, *A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems*, Journal of Computational Physics, 401 (2020), p. 109020.

[23] B. Mertens, T. Fearn, and M. Thompson, *The efficient cross-validation of principal com-*

ponents applied to principal component regression, Statistics and Computing, 5 (1995), pp. 227–235.

[24] S. Nanty, C. Helbert, A. Marrel, N. Pérot, and C. Prieur, *Uncertainty quantification for functional dependent random variables*, Computational Statistics, 32 (2017), pp. 559–583.

[25] D. Nerini, P. Monestiez, and C. Manté, *Cokriging for spatial functional data*, Journal of Multivariate Analysis, 101 (2010), pp. 409–418.

[26] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis, *Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473 (2017), p. 20160751.

[27] G. Perrin, *Adaptive calibration of a computer code with time-series output*, Reliability Engineering and System Safety, 196 (2020), p. 106728.

[28] G. Pilania, J. E. Gubernatis, and T. Lookman, *Multi-fidelity machine learning models for accurate bandgap predictions of solids*, Computational Materials Science, 129 (2017), pp. 156–163.

[29] J. Rougier, *Efficient emulators for multivariate deterministic functions*, Journal of Computational and Graphical Statistics, 17 (2008), pp. 827–843.

[30] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, *Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index*, Computer physics communications, 181 (2010), pp. 259–270.

[31] T. J. Santner, B. J. Williams, W. Notz, and B. J. Williams, *The design and analysis of computer experiments*, Springer, New York, NY, 2003.

[32] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, MIT press Cambridge, MA, 2006.

[33] Q. Zhou, Y. Wu, Z. Guo, J. Hu, and P. Jin, *A generalized hierarchical co-kriging model for multi-fidelity data fusion*, Structural and Multidisciplinary Optimization, 62 (2020), pp. 1885–1904.