Computational Statistics manuscript No.
(will be inserted by the editor)

An efficient dimension reduction for the Gaussian process emulation of two nested codes with functional outputs

Sophie Marque-Pucheu [1,2] · **Guillaume Perrin** [1] · **Josselin Garnier** [3]

Received: date / Accepted: date

Abstract In this paper, we first propose an efficient method for the dimension reduction of the functional input of a code with functional output. It is based on the approximation of the output by a model which is linear with respect to the functional input. This approximation has a sparse structure, whose parameters can be accurately estimated from a small set of observations of the code. The Gaussian predictor based on this projection basis is significantly more accurate than the one based on a projection obtained with Partial Least Squares. Secondly, the surrogate modeling of two nested codes with functional outputs is considered. In such a case, the functional output of the first code is one of the inputs of the second code. The Gaussian process regression of the second code is performed using the proposed dimension reduction. A Gaussian predictor of the nested code is obtained by composing the predictors of the two codes and linearizing this composition. Moreover, two sequential design criteria are proposed. Since we aim at performing a sensitivity analysis, these criteria are based on a minimization of the prediction variance. Moreover, one of the criteria enables to choose, if it is possible, which of the two codes to run. Thus, the computational budget is optimally allocated between the two codes and the prediction error is substantially reduced.

Keywords nested computer codes, Gaussian process regression, uncertainty quantification, dimension reduction, sequential designs.

1 Introduction

The role of simulation for the design and the certification of complex systems is increasing. The design and the certification of complex systems involve methods like optimization, risk analysis or sensitivity analysis (Sobol, 2001). However such methods require the evaluation of the output of the studied system at a large number of input points. Therefore, if the computer codes are costly, the use of surrogate models becomes necessary.
In this paper, we focus on a chain of two codes with functional outputs, which are time-varying functions. By functional output, we mean high dimensional vectorial output and not infinite dimensional output (Pinski et al., 2015). The functional output of the first code is one of the inputs of the second code.
Several challenges are posed by such a system:

- there are two codes,
- the two codes are computationally expensive, with different computational costs,
- the second code has a functional input and a functional output,

Sophie Marque-Pucheu
E-mail: sophie.marque-pucheu@cea.fr

Guillaume Perrin
E-mail: guillaume.perrin2@cea.fr

Josselin Garnier
E-mail: josselin.garnier@polytechnique.edu

[1] CEA/DAM/DIF, F-91297, Arpajon, France
[2] Laboratoire de Probabilités, Statistique et Modélisation, Université Paris Diderot, 75205 Paris Cedex 13, France
[3] Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France

– the two codes are coupled by a functional variable.

We want to perform a sensitivity analysis of the nested code output with respect to its inputs. Given the high computational cost of the codes, our first objective is to emulate the nested code output.

Among the surrogate modeling methods, the Gaussian process regression is a widely used technique (Sacks et al., 1989; Santner et al., 2003; Rasmussen and Williams, 2006). In the Gaussian process regression framework, the output of a code is treated as being the realization of a Gaussian process. However, the literature on Gaussian process regression generally deals with a single code or regard a chain or a network of codes as a single code. Besides, Marque-Pucheu et al. (2018) only treats the case of two nested codes which are linked by a scalar intermediary variable.

Besides, the literature for the Gaussian process regression of a code with functional output generally deals with the case of low dimensional vectorial inputs. Two approaches exist for the Gaussian process regression of such a code. The first one is based on a dimension reduction of the functional output (through a Principal Components Analysis for example) and the independent surrogate modeling of the projected components (Fricker et al., 2013; Higdon et al., 2008). The second one is based on a tensorized (Kronecker) structure of the covariance of the Gaussian process modeling the code, which means a separation between the time (or the indices of the functional output) and the other inputs (Hung et al., 2015; Rougier, 2008; Williams et al., 2006; Conti et al., 2009; Conti and O'Hagan, 2010). Given that the Principal Components can be difficult to estimate from a small number of observations, and following Perrin (2018), the second approach will be chosen in this paper.

Concerning the Gaussian process regression of a code with a functional input, the existing approaches generally rely on a projection of the functional input on a basis and consider a scalar output. There exist numerous projection methods of a functional variable. Among these methods, two approaches provide a projection basis of the functional input which is based on the relationship between the input and the output. These methods are based on Partial Least Squares (PLS) (Nanty et al., 2017) or Active Subspaces (Russi, 2010; Constantine et al., 2014). Moreover, Zahm et al. (2018) studies the projection based on the Active Subspace in the case of a vectorial output and a ridge approximation (Pinkus, 2015) of the code. However, the first approach cannot take into account additional information about the code, like causality. Indeed, in this paper, the output of the second code at a given time depends only on the functional input at previous times. The second approach requires the knowledge of the derivatives of the code's output, but this information is not always available.

The contributions of this paper are the following. First, we propose a method for the dimension reduction of the functional input of a code which is adapted to the output of the code. This method has two steps. First, the output is approximated by a model which is causal and linear with respect to the functional input in a first stage. Secondly, the functional input is reduced using a projection basis based on the linear approximation. This new dimension reduction method does not require the knowledge of the derivatives of the code and can take into account additional information about the code, like causality. By causality, we mean that the output at a given time depends only on the inputs at previous times. Moreover, the Gaussian predictor constructed using this dimension reduction is significantly more accurate than the one constructed using the projection based on Partial Least Squares.

In a second phase, we provide an extension of the work presented in Marque-Pucheu et al. (2018) to the case of two codes linked by a functional intermediary variable. The proposed method enables to obtain a predictor of the nested code which can take into account observations of the functional intermediary variable or observations of one of the two codes only. Moreover, this predictor of the nested code is Gaussian with fast to compute conditioned mean and variance. Finally, we propose sequential design criteria (Jones et al., 1998; Picheny et al., 2010; Bect et al., 2012) which are suited for the improvement of the accuracy of the predictor of two nested codes with a functional intermediary variable. In particular, the nested structure of the codes is exploited by adding observations of one or the other code in order to improve the prediction accuracy. This enables to optimally allocate the computational budget between the two codes.

The proposed dimension reduction method is presented in Section 2. Section 3 details the chosen approach for the surrogate modeling of a code with low dimensional vectorial inputs and a functional output. Section 4 contains the description of the Gaussian process predictor of the nested code and the associated sequential designs. Finally the methods are applied to two numerical examples in Sections 5 and 6.

Note that the proofs of the results presented in the remainder of this paper are given in Section 8.

2 Dimension reduction of the functional input of a code

In this section, we provide a technique for the dimension reduction of the functional input of a code with functional input and output.

2.1 Notations

In this paper, the following notations will be used:

- $\overset{d}{=}$ denotes the equality in distribution.
- x, y correspond to scalars.
- $\boldsymbol{x}, \boldsymbol{y}$ correspond to vectors.
- $\boldsymbol{X}, \boldsymbol{Y}$ correspond to matrices.
- $(\boldsymbol{x})_i$ denotes the entries of the vector $\boldsymbol{x}$ and $(\boldsymbol{X})_{ij}$ the entries of the matrix $\boldsymbol{X}$.
- $\boldsymbol{X}^T$ denotes the transpose of the matrix $\boldsymbol{X}$.
- $\mathrm{diag}\,(\boldsymbol{x})$ corresponds to a diagonal matrix with diagonal $\boldsymbol{x}$.
- $\mathrm{Tr}\,(\boldsymbol{X})$ denotes the trace of the matrix $\boldsymbol{X}$.
- $\mathcal{N}(\boldsymbol{x}, \boldsymbol{X})$ corresponds to the multivariate normal distribution with mean vector $\boldsymbol{x}$ and covariance matrix $\boldsymbol{X}$.
- $\mathrm{GP}(m, k)$ corresponds to a Gaussian process with mean function m and covariance function k.
- $\mathbb{E}\,[\cdot]$, $\mathbb{V}\,[\cdot]$ and $\mathrm{cov}\,(\cdot)$ denote the mathematical expectation, the variance and the covariance.
- $\boldsymbol{X}^c = \boldsymbol{I}_{N_t} - \boldsymbol{X}$ where $\boldsymbol{I}_{N_t}$ denotes the identity matrix and $\boldsymbol{X}$ is a $(N_t \times N_t)$-dimensional matrix.
- $\|\cdot\|_F$ denotes the Frobenius norm.

2.2 Formalism

In this paper, a system characterized by a low dimensional vectorial input $\boldsymbol{x}_{\mathrm{nest}} \in \mathbb{X}_{\mathrm{nest}}$, is considered. The output of the system is a time varying function. The computer codes associated with this system are discretized and their outputs are very high dimensional vectors. The number of discretization steps is denoted N_t. The output of the system can thus be associated with the deterministic function $\boldsymbol{y}_{\mathrm{nest}} : \mathbb{X}_{\mathrm{nest}} \to \mathbb{R}^{N_t}$.
Moreover, we focus on the case where the deterministic function $\boldsymbol{y}_{\mathrm{nest}}$ can be split into two nested computer codes with functional outputs. The two codes are characterized by the deterministic functions $\boldsymbol{y}_1 : \mathbb{X}_1 \to \mathbb{R}^{N_t}$ and $\boldsymbol{y}_2 : \mathbb{R}^{N_t} \times \mathbb{X}_2 \to \mathbb{R}^{N_t}$. The nested code can therefore be defined as follows:

$$\begin{array}{rcl} & & \boldsymbol{x}_2 \\ & & \searrow \\ \boldsymbol{x}_1 & \to & \boldsymbol{y}_1(\boldsymbol{x}_1) \\ & & \nearrow \end{array} \quad \boldsymbol{y}_{\mathrm{nest}}(\boldsymbol{x}_{\mathrm{nest}}) := \boldsymbol{y}_2(\boldsymbol{y}_1(\boldsymbol{x}_1), \boldsymbol{x}_2), \tag{1}$$

where $\boldsymbol{x}_{\mathrm{nest}} := (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_{\mathrm{nest}} = \mathbb{X}_1 \times \mathbb{X}_2$, $\mathbb{X}_1$ and $\mathbb{X}_2$ are subsets of $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, and d_1 and d_2 are two non-negative integers. The inputs are uncertain and are modeled by random variables with known distributions. We denote by $\mu_{\mathbb{X}_1}$ the probability measure associated with $\mathbb{X}_1$ and $\mu_{\mathbb{X}_2}$ the probability measure associated with $\mathbb{X}_2$.
The two codes are computationally expensive. Since we want to perform a sensitivity analysis of the nested code output with respect to its inputs, we aim therefore at constructing an emulator of the output of the nested code, which has to be accurate on the most likely regions of the input domain.
Furthermore, a set of n observations of the nested code is available. The observations of the inputs are independently drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$. The sets of observations of the inputs and output of the first code are denoted by:

$$\boldsymbol{X}_1^{\mathrm{init}} := \left(\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_1^{(n)}\right), \quad \boldsymbol{Y}_1^{\mathrm{init}} := \left(\boldsymbol{y}_1^{(1)} = \boldsymbol{y}_1\left(\boldsymbol{x}_1^{(1)}\right); \ldots; \boldsymbol{y}_1^{(n)} = \boldsymbol{y}_1\left(\boldsymbol{x}_1^{(n)}\right)\right), \tag{2}$$

and those of the second code are denoted by:

$$\boldsymbol{X}_2^{\mathrm{init}} := \left(\boldsymbol{x}_2^{(1)}, \ldots, \boldsymbol{x}_2^{(n)}\right), \quad \boldsymbol{Y}_2^{\mathrm{init}} := \left(\boldsymbol{y}_2^{(1)} = \boldsymbol{y}_2\left(\boldsymbol{y}_1^{(1)}, \boldsymbol{x}_2^{(1)}\right); \ldots; \boldsymbol{y}_2^{(n)} = \boldsymbol{y}_2\left(\boldsymbol{y}_1^{(n)}, \boldsymbol{x}_2^{(n)}\right)\right), \tag{3}$$

where $\boldsymbol{X}_i^{\mathrm{init}}$ is a $(n \times d_i)$-dimensional matrix and $\boldsymbol{Y}_i^{\mathrm{init}}$ is a $(N_t \times n)$-dimensional matrix, $i \in \{1, 2\}$.

As mentioned in Section 1, several challenges are raised by the objective of emulating the studied system. One of these challenges is the Gaussian process regression of the second code, which has a functional input

and a functional output. Moreover, few observations of the second code are available and these observations do not include the derivatives of the second code's output. The number of observations is small compared to the dimension of the functional variables. Besides, the existing methods for the Gaussian process regression of a code with a functional output are generally suited for the case of low dimensional vectorial inputs. Our first objective is therefore to reduce the dimension of the functional input of the second code.
Then, thanks to the dimension reduction of the functional input of the second code, the surrogate modeling of this code can be performed using an existing framework for the surrogate modeling of a code with low dimensional vectorial inputs and a functional output (see Conti et al. (2009)).

As explained in Section 1, two methods of dimension reduction coupled with a Gaussian process regression are commonly used. The first one is based on Partial Least Squares (see Nanty et al. (2017)) and the second one is based on Active Subspaces (see Constantine et al. (2014)). The first one cannot easily take into account additional information about the code, like causality. The second one requires the knowledge of the first derivatives of the output of the code, which are, in our case, not available.
In this section, we propose an innovative method for the dimension reduction of a functional input which is adapted to the output.
The proposed method is based on the remark that the estimation of the m-dimensional projection basis of $\boldsymbol{y}_1$ which is adapted to $\boldsymbol{y}_2$ can be associated with the following minimization problem:

$$\underset{\overline{\boldsymbol{y}}_1 \in \mathbb{R}^{N_t}, \boldsymbol{Z} \text{ of rank } \leq m}{\operatorname{argmin}} \int_{\mathbb{X}_1 \times \mathbb{X}_2} \left\| \boldsymbol{y}_2\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) - \boldsymbol{y}_2\left(\overline{\boldsymbol{y}}_1 + \boldsymbol{Z}\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right) - \overline{\boldsymbol{y}}_1\right), \boldsymbol{x}_2\right) \right\|^2 d\mu_{\mathbb{X}_1}\left(\boldsymbol{x}_1\right) d\mu_{\mathbb{X}_2}\left(\boldsymbol{x}_2\right). \tag{4}$$

We propose to split this minimization problem into two parts. First a linear model of $\boldsymbol{y}_2$ is obtained, by estimating a matrix $\boldsymbol{A}$ which minimizes:

$$\int_{\mathbb{X}_1 \times \mathbb{X}_2} \left\| \boldsymbol{y}_2\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) - \boldsymbol{A}\boldsymbol{y}_1\left(\boldsymbol{x}_1\right) \right\|^2 d\mu_{\mathbb{X}_1}\left(\boldsymbol{x}_1\right) d\mu_{\mathbb{X}_2}\left(\boldsymbol{x}_2\right). \tag{5}$$

Furthermore, the estimation of the linear model can take into account the causality of the system. Second, we compute the $N_t \times N_t$-dimensional matrix $\boldsymbol{Z}$ of rank $\leq m$ and the N_t-dimensional vector $\overline{\boldsymbol{y}}_1$, which minimize:

$$\int_{\mathbb{X}_1} \left\| \boldsymbol{A}\boldsymbol{y}_1\left(\boldsymbol{x}_1\right) - \boldsymbol{A}\left(\overline{\boldsymbol{y}}_1 + \boldsymbol{Z}\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right) - \overline{\boldsymbol{y}}_1\right)\right) \right\|^2 d\mu_{\mathbb{X}_1}\left(\boldsymbol{x}_1\right). \tag{6}$$

Thanks to this two-stage approach, we can provide a method for the dimension reduction of the functional input of a code which:

1. is adapted to the surrogate modeling of the output of the code,
2. does not require the knowledge of the derivatives of the output,
3. can take into account additional information about the code, like causality.

The following sections detail the two steps of the proposed method, as well as the details regarding the estimation of the projection basis from a reduced number of observations.

2.3 A two-stage dimension reduction based on a linear approximation of the output

2.3.1 A linear approximation with a sparse structure

In this section, we describe the approximation of the functional output of the code by a causal filter. This causal filter has the interesting property of being characterized by few parameters. The matrix $\boldsymbol{A}$ characterizing this causal filter is of the form $L\left(\boldsymbol{a}\right)$, with:

$$L\left(\boldsymbol{a}\right) = \begin{bmatrix} a_1 & & & \\ a_2 & a_1 & \boldsymbol{0} & \\ \vdots & \ddots & \ddots & \\ a_{N_t} & \dots & a_2 & a_1 \end{bmatrix}, \tag{7}$$

and $\boldsymbol{a} := (a_1, \dots, a_{N_t})$. Such a form corresponds to a linear causal model or filter. This linear causal model is characterized by a N_t-dimensional vector only, instead of N_t^2 in the case of a classical linear model. By noting that $L\left(\boldsymbol{a}\right)\boldsymbol{y}_1 = L\left(\boldsymbol{y}_1\right)\boldsymbol{a}$, an estimate of $\boldsymbol{a}$ can be obtained from the observations:

$$\boldsymbol{a}^* = \underset{\boldsymbol{a} \in \mathbb{R}^{N_t}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \boldsymbol{y}_2^{(i)} - L\left(\boldsymbol{y}_1^{(i)}\right)\boldsymbol{a} \right\|^2, \tag{8}$$

where the observations $\boldsymbol{y}_1^{(i)}$ are introduced in Eq. (2) and the observations $\boldsymbol{y}_2^{(i)}$ are introduced in Eq. (3). This leads to the following estimate:

$$\boldsymbol{a}^* = \left(\sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T L\left(\boldsymbol{y}_1^{(i)}\right)\right)^+ \sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T \boldsymbol{y}_2^{(i)}, \tag{9}$$

where $\left(\sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T L\left(\boldsymbol{y}_1^{(i)}\right)\right)^+$ is the pseudo-inverse of $\sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T L\left(\boldsymbol{y}_1^{(i)}\right)$.

2.3.2 A linear model based dimension reduction of the functional input

In this section, we propose an output based dimension reduction of the functional input of a linear model. First, the following Proposition provides a method of dimension reduction which is adapted to a linear model.

Proposition 1 *If one considers:*

1. *a linear model of the form $\boldsymbol{A}\boldsymbol{x}_t$ with $\boldsymbol{A}$ a $(N_t \times N_t)$-dimensional matrix and $\boldsymbol{x}_t$ a zero-mean N_t-dimensional vector,*
2. *a set of N_x observations of $\boldsymbol{x}_t$ $\{\boldsymbol{x}_t^{(i)},\ 1 \leq i \leq N_x\}$, which are gathered in the $(N_t \times N_x)$-dimensional matrix $\boldsymbol{X}_t$,*

then the m-rank matrix $\boldsymbol{Z}^*$ which minimizes $\sum_{i=1}^{N_x} \left\| \boldsymbol{A}\boldsymbol{x}_t^{(i)} - \boldsymbol{A}\boldsymbol{Z}\boldsymbol{x}_t^{(i)} \right\|^2$ with respect to $\boldsymbol{Z}$, is given by:

$$\boldsymbol{Z}^* = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-\frac{1}{2}} \boldsymbol{U}_m \boldsymbol{D}_m \boldsymbol{V}_m^T cov(\boldsymbol{X}_t)^{-\frac{1}{2}}, \tag{10}$$

where:

- *$cov(\boldsymbol{X}_t)^{-\frac{1}{2}}$ is the pseudo-inverse of $\left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}}$,*
- *$\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-\frac{1}{2}}$ is the pseudo-inverse of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}}$,*
- *$\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ is the singular-value decomposition of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}}$, $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ gather the m first columns of matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and $\boldsymbol{D}_m$ is a diagonal matrix, whose diagonal gathers the m highest singular values.*

The proof of this Proposition is given in Section 8.

Corollary 1 *If one considers:*

1. *a linear model of the form $\boldsymbol{A}\boldsymbol{x}_t$ with $\boldsymbol{A}$ a $(N_t \times N_t)$-dimensional matrix and $\boldsymbol{x}_t$ a zero-mean N_t-dimensional vector,*
2. *a set of N_x observations of $\boldsymbol{x}_t$ $\{\boldsymbol{x}_t^{(i)},\ 1 \leq i \leq N_x\}$, which are gathered in the $(N_t \times N_x)$-dimensional matrix $\boldsymbol{X}_t$,*

then a m-dimensional projection of $\boldsymbol{x}_t$, which is adapted to this linear model, is defined by the following application:

$$\boldsymbol{x}_t \mapsto \boldsymbol{V}_m^T cov(\boldsymbol{X}_t)^{-\frac{1}{2}} \boldsymbol{x}_t, \tag{11}$$

with $\boldsymbol{V}_m$ gathering the m right-singular vectors associated with the m highest singular values of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} cov(\boldsymbol{X}_t)^{\frac{1}{2}}$.

Corollary 2 *The pair $(\overline{\boldsymbol{y}}_1, \boldsymbol{Z})$ which minimizes:*

$$\int_{\mathbb{X}_1} \left\| \boldsymbol{A}\boldsymbol{y}_1(\boldsymbol{x}_1) - \boldsymbol{A}\left(\overline{\boldsymbol{y}}_1 + \boldsymbol{Z}\left(\boldsymbol{y}_1(\boldsymbol{x}_1) - \overline{\boldsymbol{y}}_1\right)\right)\right\|^2 d\mu_{\mathbb{X}_1}(\boldsymbol{x}_1), \tag{12}$$

with $\overline{\boldsymbol{y}}_1$ a N_t-dimensional vector and $\boldsymbol{Z}$ a $(N_t \times N_t)$-dimensional matrix of rank $\leq m$, is given by:

$$\overline{\boldsymbol{y}}_1 = \mathbb{E}\left[\boldsymbol{y}_1\right], \tag{13}$$

and:

$$\boldsymbol{Z} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-\frac{1}{2}} \boldsymbol{U}_m \boldsymbol{D}_m \boldsymbol{V}_m^T cov(\boldsymbol{y}_1)^{-\frac{1}{2}}, \tag{14}$$

with $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ gathering respectively the left and right-singular vectors associated with the m highest singular values of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} cov(\boldsymbol{y}_1)^{\frac{1}{2}}$, which are gathered in the diagonal of the diagonal matrix $\boldsymbol{D}_m$.

2.3.3 An efficient estimation of the covariance matrix of the functional input

If a predictor of the first code output is available, then, following the Kriging believer approach of Ginsbourger et al. (2010), the covariance matrix can be estimated from the predicted output at a large number N_1 of input points independently drawn according to $\mu_{\mathbb{X}_1}$. The predictor can be constructed from the set of n observations of the input and output of the first code $\boldsymbol{X}_1^{\text{init}}$ and $\boldsymbol{Y}_1^{\text{init}}$. This estimate of the covariance matrix can be written:

$$\boldsymbol{R}_{\boldsymbol{\mu}_1} = \frac{1}{N_1 - 1} \sum_{i=1}^{N_1} \left(\boldsymbol{\mu}_1^{(i)} - \overline{\boldsymbol{\mu}_1} \right) \left(\boldsymbol{\mu}_1^{(i)} - \overline{\boldsymbol{\mu}_1} \right)^T, \tag{15}$$

where $\boldsymbol{\mu}_1^{(i)} = \boldsymbol{\mu}_1 \left(\tilde{\boldsymbol{x}}_1^{(i)} \right)$ is the mean of a Gaussian process emulator of $\boldsymbol{y}_1$ at input $\tilde{\boldsymbol{x}}_1^{(i)}$. The $\{\tilde{\boldsymbol{x}}_1^{(i)}, 1 \leq i \leq N_1\}$ are independently drawn according to $\mu_{\mathbb{X}_1}$ and

$$\overline{\boldsymbol{\mu}_1} = \frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{\mu}_1^{(i)} \tag{16}$$

is the empirical average of the predicted trajectories.
The estimation of the covariance matrix from an increased number of paths compared to the initial set of observations enables to increase the rank of the estimator of the covariance matrix. We observed on a series of test cases (see Section 5.2) that this procedure leads to a more accurate estimation of the covariance matrix of the first output.

2.3.4 Combination between the linear approximation and the linear model based dimension reduction

In what follows, we combine the approximation by a linear model with a sparse structure and the linear model based dimension reduction of Proposition 1. Such an approach is similar to the PLS regression, because the dimension reduction of the input is adapted to the output. However, unlike PLS regression, the proposed approach can take into account the sparse structure of the linear model of Eq. (7).
Thus, if $\boldsymbol{A}^* = L(\boldsymbol{a}^*)$ denotes the matrix characterizing the approximated sparse linear model, then, based on Corollary 1, we can define the projection matrix as follows:

$$\widehat{\boldsymbol{\Pi}}_m = \boldsymbol{V}_m^T \boldsymbol{R}_{\boldsymbol{\mu}_1}^{-\frac{1}{2}}, \tag{17}$$

where $\boldsymbol{R}_{\boldsymbol{\mu}_1}$ is defined by Eq. (15), $\boldsymbol{R}_{\boldsymbol{\mu}_1}^{-\frac{1}{2}}$ is the pseudo-inverse of $\boldsymbol{R}_{\boldsymbol{\mu}_1}^{\frac{1}{2}}$ and $\boldsymbol{V}_m$ gathers the m right-singular vectors corresponding to the m highest singular values of $\left((\boldsymbol{A}^*)^T \boldsymbol{A}^* \right)^{\frac{1}{2}} \boldsymbol{R}_{\boldsymbol{\mu}_1}^{\frac{1}{2}}$.
Based on Corollary 2, we can therefore define the dimension reduction function r_m:

$$r_m(\boldsymbol{y}_1) = \widehat{\boldsymbol{\Pi}}_m \left(\boldsymbol{y}_1 - \overline{\boldsymbol{\mu}_1} \right), \tag{18}$$

with $\widehat{\boldsymbol{\Pi}}_m$ the $m \times N_t$-dimensional matrix defined by Eq. (17).
By comparison, in the case of the PLS method (see Wold (1966) and Höskuldsson (1988)), this function would have been:

$$r_m(\boldsymbol{y}_1) = \boldsymbol{L}_m^T \left(\boldsymbol{y}_1 - \overline{\boldsymbol{y}_1^{\text{init}}} \right). \tag{19}$$

The $(N_t \times m)$-dimensional matrix $\boldsymbol{L}_m$ gathers the left-singular vectors associated with the m highest singular values of the covariance matrix $\boldsymbol{R}_{\boldsymbol{Y}_1^{\text{init}} \boldsymbol{Y}_2^{\text{init}}}$, which is defined by:

$$\boldsymbol{R}_{\boldsymbol{Y}_1^{\text{init}} \boldsymbol{Y}_2^{\text{init}}} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\boldsymbol{y}_1^{(i)} - \overline{\boldsymbol{y}_1^{\text{init}}} \right) \left(\boldsymbol{y}_2^{(i)} - \overline{\boldsymbol{y}_2^{\text{init}}} \right)^T, \tag{20}$$

where $\boldsymbol{y}_1^{(i)}$ are introduced in Eq. (2), $\boldsymbol{y}_2^{(i)}$ are introduced in Eq. (3) and

$$\overline{\boldsymbol{y}_j^{\text{init}}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_j^{(i)}, \ j \in \{1, 2\}. \tag{21}$$

The optimal size m of the projection basis can be estimated using a stepwise forward selection criterion (Efroymson, 1960) when constructing the predictor of the second code.

Besides, it is worth noticing that, when computing the singular value decomposition of matrices of rank below n (the number of observations), only the n first singular values (or eigenvalues for the symmetric matrices) are non-zero.

In this paper, we adopt a Gaussian process regression framework. Thanks to the dimension reduction of the functional input of the second code, the second code can be associated with a Gaussian process which has a functional output and a low dimensional vectorial input comprising the projection of the functional input and the inputs $\boldsymbol{x}_2$. The next section describes the properties of the Gaussian process regression of a code with a low dimensional vectorial input and a functional output.

3 Gaussian process regression with low dimensional inputs and a functional output

As mentioned previously, the surrogate modeling will be performed using the Gaussian process regression framework proposed in Conti et al. (2009). The Gaussian processes modeling the codes have a tensorized structure, which means that there is a separation between the time (or the indices of the functional output) and the other inputs. This tensorized structure is obtained through a Kronecker structure of the covariance function and a specific structure of the mean function.

As mentioned in the introduction of this paper, we consider the system defined by Eq. (1). In order to construct a predictor of the nested code output, we propose the following system of Gaussian processes:

$$\begin{array}{ccccc} & & & \overset{\boldsymbol{x}_2}{\searrow} & \overset{\boldsymbol{Y}_2}{} \\ \boldsymbol{x}_1 & \xrightarrow{\boldsymbol{Y}_3} & \boldsymbol{\rho} := r_m\left(\boldsymbol{y}_1\right) & & \boldsymbol{y}_2, \\ & & & \underset{\nearrow}{} & \end{array} \tag{22}$$

and

$$\boldsymbol{x}_1 \xrightarrow{\boldsymbol{Y}_1} \boldsymbol{y}_1 \tag{23}$$

where $r_m : \mathbb{R}^{N_t} \mapsto \mathbb{R}^m$ denotes the projection function of the functional input of the second code on its reduced basis of size m, and $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$ and $\boldsymbol{Y}_3$ are Gaussian processes.

The Gaussian process $\boldsymbol{Y}_1 : \mathbb{X}_1 \to \mathbb{R}^{N_t}$ is used for the estimation of the projection basis (more precisely, for the estimation of the covariance matrix of $\boldsymbol{y}_1$), and for one of the sequential designs. For further details, the interested reader can refer to the definition of the sequential design criteria in Section 4.2. The Gaussian process $\boldsymbol{Y}_3 : \mathbb{X}_1 \to \mathbb{R}^m$ is used for the emulation of the function $\boldsymbol{x}_1 \mapsto r_m\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right)\right)$. The output of the second code is emulated by the Gaussian process $\boldsymbol{Y}_2 : \mathbb{R}^m \times \mathbb{X}_2 \to \mathbb{R}^{N_t}$, which is indexed by $(\boldsymbol{\rho}, \boldsymbol{x}_2) \in \mathbb{R}^m \times \mathbb{X}_2$.
Notice that if two outputs of the first code $\boldsymbol{y}_1\left(\boldsymbol{x}_1\right)$ and $\boldsymbol{y}_1\left(\boldsymbol{x}_1'\right)$ have the same image under r_m, then their predicted values under $\boldsymbol{Y}_2$ are the same, but they can be different under $\boldsymbol{y}_2$.
Finally, the composition of the Gaussian predictors based on the Gaussian processes $\boldsymbol{Y}_3$ and $\boldsymbol{Y}_2$ will be useful for the prediction of the function $(\boldsymbol{x}_1, \boldsymbol{x}_2) \mapsto \boldsymbol{y}_2\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)$, where $\boldsymbol{Y}_2\left(\boldsymbol{Y}_3\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)$ approximates $\boldsymbol{y}_2\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)$.

To harmonize the notations, we define:

$$\{\bar{\boldsymbol{x}}_i; \overline{\mathbb{X}}_i\} = \begin{cases} \{\boldsymbol{x}_1; \mathbb{X}_1\} \text{ if } i = 1 \text{ or } i = 3, \\ \{(\boldsymbol{\rho}, \boldsymbol{x}_2); \mathbb{R}^m \times \mathbb{X}_2\}, \text{ if } i = 2, \end{cases} \tag{24}$$

and:

$$\boldsymbol{y}_3 = r_m\left(\boldsymbol{y}_1\right). \tag{25}$$

Moreover, a set of observations of the inputs and outputs of the two codes is available. In this paper, we consider the case where an initial design is drawn according to the distribution of the inputs of the nested code, i.e. $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$. The design can be enriched at a later stage. However, this initial design will be useful for the estimation of the hyperparameters of the covariance functions of the Gaussian processes, as well as the estimation of the projection basis of the functional intermediary variable. Then this initial design is sequentially enriched. The criteria used for the sequential design will be defined in Section 3.2.
The sets of n_1 observations of the inputs and the functional output of the first code are denoted by:

$$\begin{aligned} \boldsymbol{X}_1^{\text{obs}} &:= \left(\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_1^{(n_1)}\right), \\ \boldsymbol{Y}_1^{\text{obs}} &:= \left(\boldsymbol{y}_1^{(1)} = \boldsymbol{y}_1\left(\boldsymbol{x}_1^{(1)}\right); \ldots; \boldsymbol{y}_1^{(n_1)} = \boldsymbol{y}_1\left(\boldsymbol{x}_1^{(n_1)}\right)\right), \end{aligned} \tag{26}$$

where $\overline{\boldsymbol{X}}_1^{\text{obs}}$ is a $(n_1 \times d_1)$-dimensional matrix, and $\boldsymbol{Y}_1^{\text{obs}}$ a $(N_t \times n_1)$-dimensional matrix.
In the same way, the sets of n_2 observations of the inputs and the output of the second code are denoted by:

$$\begin{aligned}
\boldsymbol{\Phi}_1^{\text{obs}} &:= \left(\boldsymbol{\varphi}_1^{(1)}; \ldots; \boldsymbol{\varphi}_1^{(n_2)}\right), \\
\boldsymbol{P}^{\text{obs}} &:= \left(\boldsymbol{\rho}^{(1)} = r_m\left(\boldsymbol{\varphi}_1^{(1)}\right), \ldots, \boldsymbol{\rho}^{(n_2)} = r_m\left(\boldsymbol{\varphi}_1^{(n_2)}\right)\right), \\
\boldsymbol{X}_2^{\text{obs}} &:= \left(\boldsymbol{x}_2^{(1)}, \ldots, \boldsymbol{x}_2^{(n_2)}\right), \\
\boldsymbol{Y}_2^{\text{obs}} &:= \left(\boldsymbol{y}_2^{(1)} = \boldsymbol{y}_2\left(\boldsymbol{\varphi}_1^{(1)}, \boldsymbol{x}_2^{(1)}\right); \ldots; \boldsymbol{y}_2^{(n_2)} = \boldsymbol{y}_2\left(\boldsymbol{\varphi}_1^{(n_2)}, \boldsymbol{x}_2^{(n_2)}\right)\right),
\end{aligned} \tag{27}$$

where $\boldsymbol{X}_2^{\text{obs}}$ is a $(n_2 \times d_2)$-dimensional matrix, $\boldsymbol{P}^{\text{obs}}$ is a $(n_2 \times m)$-dimensional matrix and $\boldsymbol{\Phi}_1^{\text{obs}}$ and $\boldsymbol{Y}_2^{\text{obs}}$ are $(N_t \times n_2)$-dimensional matrices. Note that $\boldsymbol{\varphi}_1$ can be the output of the first code or a prediction of the output of the first code.
Finally, at the initial stage, we have:

$$\boldsymbol{X}_1^{\text{obs}} = \boldsymbol{X}_1^{\text{init}},\ \boldsymbol{Y}_1^{\text{obs}} = \boldsymbol{Y}_1^{\text{init}}, \tag{28}$$

and

$$\boldsymbol{\Phi}_1^{\text{obs}} = \boldsymbol{Y}_1^{\text{init}},\ \boldsymbol{X}_2^{\text{obs}} = \boldsymbol{X}_2^{\text{init}},\ \boldsymbol{Y}_2^{\text{obs}} = \boldsymbol{Y}_2^{\text{init}}, \tag{29}$$

where $\boldsymbol{X}_1^{\text{init}}$, $\boldsymbol{Y}_1^{\text{init}}$ are defined by Eq. (2) and $\boldsymbol{X}_2^{\text{init}}$, $\boldsymbol{Y}_2^{\text{init}}$ are defined by Eq. (3).
Notice that, at the initial stage, the number of observations of the two codes is the same. However, in the following, we provide a predictor of the nested code which can take into account a different number of observations of the two codes.
We can therefore introduce the following notations for the observations of the inputs of the three studied functions:

$$\overline{\boldsymbol{X}}_i^{\text{obs}} := \begin{cases} \boldsymbol{X}_1^{\text{obs}}, \text{ if } i = 1 \text{ or } i = 3, \\ \left(\left(\boldsymbol{\rho}^{(1)}, \boldsymbol{x}_2^{(1)}\right), \ldots, \left(\boldsymbol{\rho}^{(n_2)}, \boldsymbol{x}_2^{(n_2)}\right)\right), \text{ if } i = 2. \end{cases} \tag{30}$$

Furthermore, the observations of $\boldsymbol{y}_3$ can be defined as:

$$\boldsymbol{Y}_3^{\text{obs}} := \left(r_m\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1^{(1)}\right)\right), \ldots, r_m\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1^{(n_1)}\right)\right)\right), \tag{31}$$

where $\boldsymbol{Y}_3^{\text{obs}}$ is a $(m \times n_1)$-dimensional matrix.

3.1 Prediction for a given set of observations

Following Conti et al. (2009) and its generalization of Universal Kriging to the multi-output case, a functional output $\boldsymbol{y}_i$, $i \in \{1, 2, 3\}$, can be modeled by a realization of a Gaussian process $\boldsymbol{Y}_i$, whose *a priori* distribution is:

$$\boldsymbol{Y}_i(\cdot)\,|\boldsymbol{M}_i, \boldsymbol{R}_{t_i}, C_i \sim \text{GP}\left(\boldsymbol{M}_i \boldsymbol{h}_i(\cdot), \boldsymbol{R}_{t_i} \otimes C_i(\cdot,\cdot)\right), \tag{32}$$

where $\boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i) := (\boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i, t_1), \ldots, \boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i, t_{N_t}))$, $\boldsymbol{M}_i$ is a $(N_t \times p_i)$-dimensional matrix to be determined, $\boldsymbol{h}_i$ is a vector of p_i basis functions, $\boldsymbol{R}_{t_i}$ is a $(N_t \times N_t)$-dimensional symmetric non negative definite matrix, to be determined, and C_i is a covariance function on $\overline{\mathbb{X}}_i \times \overline{\mathbb{X}}_i$.
Moreover, in the Universal Kriging framework, if the prior distribution of $\boldsymbol{M}_i$ is an improper uniform distribution on the space of the $(N_t \times p_i)$-dimensional real-valued matrices, the conditional distribution of $\boldsymbol{Y}_i$ given the observations is:

$$\boldsymbol{Y}_i^c := \boldsymbol{Y}_i | \boldsymbol{Y}_i^{\text{obs}}, C_i \sim \text{GP}\left(\boldsymbol{\mu}_i^c, \widehat{\boldsymbol{R}}_{t_i} \otimes C_i^c\right), \tag{33}$$

with:

$$\begin{aligned}
\boldsymbol{\mu}_i^c(\bar{\boldsymbol{x}}_i) &= \widehat{\boldsymbol{M}}_i \boldsymbol{h}_i(\bar{\boldsymbol{x}}_i) + \left[\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i \boldsymbol{H}_i^{\text{obs}}\right] \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i\right), \\
C_i^c\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i'\right) &= C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i'\right) - C_i\left(\bar{\boldsymbol{x}}_i, \overline{\boldsymbol{X}}_i^{\text{obs}}\right) \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C_i\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i'\right) \\
&\quad + \boldsymbol{u}_i(\bar{\boldsymbol{x}}_i)^T \left(\boldsymbol{H}_i^{\text{obs}} \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} \left(\boldsymbol{H}_i^{\text{obs}}\right)^T\right)^{-1} \boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i'\right), \\
\boldsymbol{u}_i(\bar{\boldsymbol{x}}_i) &= \boldsymbol{h}_i(\bar{\boldsymbol{x}}_i) - \boldsymbol{H}_i^{\text{obs}} \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C_i\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i\right).
\end{aligned} \tag{34}$$

$\widehat{\boldsymbol{M}}_i$ denotes the posterior mean of $\boldsymbol{M}_i$:

$$\begin{aligned}\widehat{\boldsymbol{M}}_i &= \mathbb{E}\left[\boldsymbol{M}_i|\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{R}_{t_i}, C_i\right] \\ &= \boldsymbol{Y}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T\right)^{-1},\end{aligned} \tag{35}$$

$\widehat{\boldsymbol{R}}_{t_i}$ is the maximum likelihood estimator of $\boldsymbol{R}_{t_i}$ (see (Perrin, 2018) for further details):

$$\widehat{\boldsymbol{R}}_{t_i} = \boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right) = \frac{1}{n_i}\left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)^T, \tag{36}$$

and $\boldsymbol{R}_i^{\text{obs}}$ is the $(n_i \times n_i)$-dimensional matrix such that:

$$\left(\boldsymbol{R}_i^{\text{obs}}\right)_{kl} = C_i\left(\bar{\boldsymbol{x}}_i^{(k)}, \bar{\boldsymbol{x}}_i^{(l)}\right). \tag{37}$$

Here $\bar{\boldsymbol{x}}_i^{(k)}$ denotes the k-th observation of $\overline{\boldsymbol{X}}_i^{\text{obs}}$ and $\boldsymbol{H}_i^{\text{obs}}$ is a $(n_i \times p_i)$-dimensional matrix whose j-th line is given by $\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{(j)}\right)$.
The prediction variance of the functional output at input $\bar{\boldsymbol{x}}_i$ is thus defined by:

$$\left(\boldsymbol{\sigma}_i^c\right)^2(\bar{\boldsymbol{x}}_i) := \text{diag}\left(\mathbb{V}\left[\boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i)|\boldsymbol{Y}_i^{\text{obs}}\right]\right), \tag{38}$$

This prediction variance can also be written:

$$\left(\boldsymbol{\sigma}_i^c\right)^2(\bar{\boldsymbol{x}}_i) = \text{diag}\left(\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right)\right) v_i\left(\bar{\boldsymbol{x}}_i; \overline{\boldsymbol{X}}_i^{\text{obs}}\right), \tag{39}$$

where $v_i\left(\bar{\boldsymbol{x}}_i; \overline{\boldsymbol{X}}_i^{\text{obs}}\right) = C_i^c(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i)$.
In our case, the regularity of the correlation functions C_i is unknown. Following Cornford et al. (2002), we use Matérn $\frac{5}{2}$ correlation functions for the functions C_i, $i \in \{1,2,3\}$. However, other correlation functions can be chosen. For a further discussion on the choice of the regularity of the Matérn correlation function, the interested reader is referred to Rasmussen and Williams (2006).

Since our objective is to obtain an accurate prediction mean, the correlation lengths of the covariance function are estimated by minimizing a Cross Validation criterion (Dubrule, 1983; Bachoc, 2013):

$$\begin{aligned}\widehat{\boldsymbol{\ell}}_i &= \underset{\boldsymbol{\ell}_i \in \mathbb{R}^{d_i}}{\text{argmin}}\ \text{Tr}\left(\boldsymbol{Y}_i^{\text{obs}}\boldsymbol{R}_i^-(\boldsymbol{\ell}_i)\,\text{diag}\left(\boldsymbol{R}_i^-(\boldsymbol{\ell}_i)\right)^{-2}\boldsymbol{R}_i^-(\boldsymbol{\ell}_i)\left(\boldsymbol{Y}_i^{\text{obs}}\right)^T\right), \\ &= \underset{\boldsymbol{\ell}_i \in \mathbb{R}^{d_i}}{\text{argmin}}\ \left\|\boldsymbol{Y}_i^{\text{obs}}\boldsymbol{R}_i^-(\boldsymbol{\ell}_i)\,\text{diag}\left(\boldsymbol{R}_i^-(\boldsymbol{\ell}_i)\right)^{-1}\right\|_F^2,\end{aligned} \tag{40}$$

where:

$$\boldsymbol{R}_i^-(\boldsymbol{\ell}_i) = \left(\boldsymbol{R}_i^{\text{obs}}(\boldsymbol{\ell}_i)\right)^{-1} + \left(\boldsymbol{R}_i^{\text{obs}}(\boldsymbol{\ell}_i)\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}(\boldsymbol{\ell}_i)\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T\right)^{-1}\boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}(\boldsymbol{\ell}_i)\right)^{-1}. \tag{41}$$

$\boldsymbol{R}_i^{\text{obs}}(\boldsymbol{\ell}_i)$ is based on Eq. (37) and takes into account the fact that C_i depends on the correlation length $\boldsymbol{\ell}_i$. Thanks to this matrix-form criterion, the correlation lengths of the correlation functions can be estimated quickly.

As for the correlation lengths, the size of the projection basis associated with r_m can be chosen according to a Cross Validation criterion (see Eq. (40)) of the form:

$$m^* = \underset{m \in \{1,\ldots,n_2\}}{\text{argmin}}\ \left\|\boldsymbol{Y}_2^{\text{obs}}\boldsymbol{R}_2^-(m)\,\text{diag}\left(\boldsymbol{R}_2^-(m)\right)^{-1}\right\|_F^2, \tag{42}$$

where:

$$\begin{aligned}\boldsymbol{R}_2^-(m) = &\left(\boldsymbol{R}_2^{\text{obs}}(m)\right)^{-1} + \left(\boldsymbol{R}_2^{\text{obs}}(m)\right)^{-1}\left(\boldsymbol{H}_2^{\text{obs}}(m)\right)^T \\ &\left(\boldsymbol{H}_2^{\text{obs}}(m)\left(\boldsymbol{R}_2^{\text{obs}}(m)\right)^{-1}\left(\boldsymbol{H}_2^{\text{obs}}(m)\right)^T\right)^{-1}\boldsymbol{H}_2^{\text{obs}}(m)\left(\boldsymbol{R}_2^{\text{obs}}(m)\right)^{-1}.\end{aligned} \tag{43}$$

$\boldsymbol{R}_2^{\text{obs}}(m)$ is a $(n_2 \times n_2)$-dimensional matrix and $\boldsymbol{H}_2^{\text{obs}}(m)$ is a $(p_2(m) \times n)$-dimensional matrix, such that:

$$\left(\boldsymbol{R}_2^{\text{obs}}(m)\right)_{ij} = C_2\left(\left(r_m\left(\varphi_1^{(i)}\right), \boldsymbol{x}_2^{(i)}\right), \left(r_m\left(\varphi_1^{(j)}\right), \boldsymbol{x}_2^{(j)}\right)\right), \tag{44}$$

and

$$\left(\boldsymbol{H}_2^{\text{obs}}(m)\right)_{.i} = \boldsymbol{h}_2\left(\left(r_m\left(\varphi_1^{(i)}\right), \boldsymbol{x}_2^{(i)}\right)\right). \tag{45}$$

Notice that the hyperparameters of the covariance function C_2 have to be re-estimated for each value of m. These computations can be speeded up using the criterion of Eq. (40).

3.2 Sequential enrichment of the design of experiments

From Eq. (33), it can be inferred that the accuracy of the Gaussian posterior predictor depends on the choice of the set of observations $\boldsymbol{Y}_i^{\text{obs}}$. Consequently, the predictor can be improved by choosing an appropriate design.
In the case of a code with scalar output, if one aims at improving the prediction accuracy of a given quantity of interest, a design criterion based on the integrated prediction variance can be used (see Sacks et al. (1989); Santner et al. (2003)). However, the computation of such a design criterion in one step can be cumbersome, because of the high number of possible sets. Indeed, if a discrete search is performed, the number of possible combinations is $\binom{\mathcal{N}_i}{n_i}$, where $\mathcal{N}_i$ is the number of candidates. Moreover, the covariance function of the Gaussian process C_i is generally not known at the initial stage and it has to be estimated from an initial set of observations of the codes associated with the Gaussian process. Then this initial design can be enriched thanks to sequential design criteria or Stepwise Uncertainty Reduction (SUR) methods (Bect et al., 2012; Picheny et al., 2010). Besides, the hyperparameters of the covariance function of the Gaussian process can be re-estimated from the new set of observations at each step. Following Bates et al. (1996) and Picheny et al. (2010), a sequential design criterion based on the reduction of the integrated prediction variance and a SUR approach can be used.
In this paper, since the codes have functional outputs, we propose a criterion based on the trace of the integrated prediction variance. Based on Eq. (38), this criterion can be written:

$$\bar{\boldsymbol{x}}_i^{new} = \underset{\bar{\boldsymbol{x}}_i^* \in \overline{\mathbb{X}}_i}{\operatorname{argmin}} \int_{\overline{\mathbb{X}}_i} \operatorname{Tr}\left(\mathbb{V}\left[\boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i) | \boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{y}_i(\bar{\boldsymbol{x}}_i^*)\right]\right) d\mu_{\overline{\mathbb{X}}_i}(\bar{\boldsymbol{x}}_i), \tag{46}$$

where

$$\mu_{\overline{\mathbb{X}}_i} := \begin{cases} \mu_{\mathbb{X}_1}, \text{ if } i = 1 \text{ or } i = 3, \\ \mu_{\boldsymbol{\rho}} \times \mu_{\mathbb{X}_2}, \text{ if } i = 2, \end{cases} \tag{47}$$

with $\mu_{\boldsymbol{\rho}}$ denoting the probability measure over $\mathbb{R}^m$ associated with the distribution of $r_m(\boldsymbol{y}_1(\boldsymbol{x}_1))$ under $\mu_{\mathbb{X}_1}$. This probability measure is unknown, because the distribution of $\boldsymbol{y}_1(\boldsymbol{x}_1)$ under $\mu_{\mathbb{X}_1}$ is not known. However, draws according to $\mu_{\boldsymbol{\rho}}$ can be obtained approximately by drawing independent points according to $\mu_{\mathbb{X}_1}$ and then considering the image of these points under $\boldsymbol{\mu}_3^c$, which is defined by Eq. (34).

Furthermore, $\boldsymbol{y}_i(\bar{\boldsymbol{x}}_i^*)$ is not known. In order to compute the design criterion, a Kriging Believer approach (Ginsbourger et al., 2010) can be used, as mentioned in the Proposition below. Moreover, the Proposition is based on the simplifications presented in the Lemma 1.

Lemma 1 *In the Kriging Believer approach, the posterior mean of $\boldsymbol{M}_i$ and the maximum likelihood estimate of $\boldsymbol{R}_{t_i}$ given the observations $\boldsymbol{Y}_i^{obs}$ and a new observation at input point $\bar{\boldsymbol{x}}_i^*$, are given by:*

$$\mathbb{E}\left[\boldsymbol{M}_i | \{\boldsymbol{Y}_i^{obs}, \boldsymbol{Y}_i(\bar{\boldsymbol{x}}_i^*) = \boldsymbol{\mu}_i^c(\bar{\boldsymbol{x}}_i^*)\}, C_i\right] = \mathbb{E}\left[\boldsymbol{M}_i | \boldsymbol{Y}_i^{obs}, C_i\right], \tag{48}$$

and

$$\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{obs}, \boldsymbol{\mu}_i^c(\bar{\boldsymbol{x}}_i^*)\right) = \frac{n_i}{n_i + 1} \boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{obs}\right), \tag{49}$$

where $\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right)$ is defined by Eq. (36)
The proof of this lemma can be found in Section 8.

Proposition 2 *In the Kriging Believer framework, the criterion of Eq.* (46) *can be written:*

$$\bar{\boldsymbol{x}}_i^{new} = \underset{\bar{\boldsymbol{x}}_i^* \in \overline{\mathbb{X}}_i}{\operatorname{argmin}} \int_{\overline{\mathbb{X}}_i} v_i\left(\bar{\boldsymbol{x}}_i; \overline{\boldsymbol{X}}_i^{obs}, \bar{\boldsymbol{x}}_i^*\right) d\mu_{\overline{\mathbb{X}}_i}\left(\bar{\boldsymbol{x}}_i\right), \tag{50}$$

where $v_i\left(\bar{\boldsymbol{x}}_i; \overline{\boldsymbol{X}}_i^{obs}, \bar{\boldsymbol{x}}_i^*\right)$ *is defined by Eq.* (39)*, with* $\overline{\boldsymbol{X}}_i^{obs}$ *defined by Eq.* (30)*.*

The proof of this Proposition is given in Section 8. The integral of the criterion can be computed using the following Monte-Carlo method: the function $v_i\left(\bar{\boldsymbol{x}}_i; \overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right)$ is computed at independent points $\bar{\boldsymbol{x}}_i$, drawn according to $\mu_{\overline{\mathbb{X}}_i}$, and an estimate of the integral is obtained by computing the empirical mean of these computed outputs of v_i.

Once a predictor of all the studied functions can be defined, we can study the surrogate modeling of the nested code.

4 Surrogate modeling of the nested code

In this section, we propose a predictor of the nested code which can take into account all the available observations. This predictor is obtained through the coupling of two predictors associated with the two codes and the linearization of this coupling. We also define sequential design criteria aiming at improving the accuracy of the predictor of the nested code. The predictor obtained is Gaussian, with fast to compute mean and variance.

4.1 A Gaussian predictor of the nested code thanks to a linearization of the coupling of two predictors

In the previous section, two Gaussian predictors were obtained: $\boldsymbol{Y}_2^c : \mathbb{R}^m \times \mathbb{R}^{d_2} \to \mathbb{R}^{N_t}$ and $\boldsymbol{Y}_3^c : \mathbb{R}^{d_1} \to \mathbb{R}^m$. Based on these predictors, we propose the following predictor of $(\boldsymbol{x}_1, \boldsymbol{x}_2) \mapsto \boldsymbol{y}_2\left(\boldsymbol{y}_1\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)$:

$$\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) := \boldsymbol{Y}_2^c\left(\boldsymbol{Y}_3^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right). \tag{51}$$

with $\boldsymbol{Y}_2^c$ and $\boldsymbol{Y}_3^c$ defined by Eq. (33).

Proposition 3 *For all* $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_1 \times \mathbb{X}_2$*, if* $\boldsymbol{\xi} \sim \mathcal{N}\left(0, \widehat{\boldsymbol{R}}_{t_3}\right)$*, then:*

$$\mathbb{E}\left[\boldsymbol{Y}_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2)\right] = \mathbb{E}\left[\boldsymbol{\mu}_2^c(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1) + \sigma_{\boldsymbol{x}_3}^c(\boldsymbol{x}_1)\boldsymbol{\xi}, \boldsymbol{x}_2)\right], \tag{52}$$

$$\mathbb{E}\left[\boldsymbol{Y}_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2)\left(\boldsymbol{Y}_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2)\right)^T\right] = \mathbb{E}\left[\begin{array}{l} \boldsymbol{\mu}_2^c(\boldsymbol{\mu}_1^c(\boldsymbol{x}_1) + \sigma_{\boldsymbol{x}_1}^c(\boldsymbol{x}_1)\boldsymbol{\xi}, \boldsymbol{x}_2)\ \boldsymbol{\mu}_2^c(\boldsymbol{\mu}_1^c(\boldsymbol{x}_1) + \sigma_{\boldsymbol{x}_1}^c(\boldsymbol{x}_1)\boldsymbol{\xi}, \boldsymbol{x}_2)^T \\ + \{\sigma_{\boldsymbol{x}_2}^c(\boldsymbol{\mu}_1^c(\boldsymbol{x}_1) + \sigma_{\boldsymbol{x}_1}^c(\boldsymbol{x}_1)\boldsymbol{\xi}, \boldsymbol{x}_2)\}^2 \widehat{\boldsymbol{R}}_{t_2} \end{array}\right], \tag{53}$$

where $\sigma_{\boldsymbol{x}_1}^c(\boldsymbol{x}_1) = \sqrt{C_1^c(\boldsymbol{x}_1, \boldsymbol{x}_1)}$ *and* $\sigma_{\boldsymbol{x}_2}^c(\boldsymbol{x}_2) = \sqrt{C_2^c(\boldsymbol{x}_2, \boldsymbol{x}_2)}$*.*

The proof of this Proposition is given in Section 8. It can be seen that the computation of each of the two first moments of the coupling of the predictors involves the computation of a m-dimensional integral. This computation can be performed using Monte-Carlo methods (Baker, 1977), which can be computationally expensive. We therefore develop an adaptation of the linearized method proposed in Marque-Pucheu et al. (2018) to the case where the outputs of the two codes are time-varying functions. This enables to obtain a Gaussian predictor of the nested code with conditioned mean and variance which can be computed quickly.

Proposition 4 *If*

1. $\boldsymbol{Y}_i^c = \boldsymbol{\mu}_i^c + \boldsymbol{\varepsilon}_i^c$ *denotes a predictor of* $\boldsymbol{y}_i$ *for* $i \in \{2, 3\}$ *and* $\boldsymbol{\varepsilon}_i^c(\cdot) \sim GP\left(\boldsymbol{0}, \widehat{\boldsymbol{R}}_{t_i} C_i^c(\cdot, \cdot)\right)$*, where* $\boldsymbol{\mu}_i^c$ *and* C_i^c *are defined in Eq.* (34) *and* $\widehat{\boldsymbol{R}}_{t_i}$ *in Eq.* (36)*.*
2. *the magnitude of the prediction error* $\boldsymbol{\varepsilon}_3^c$ *is small,*

then a Gaussian predictor of the nested code can be obtained, and its mean and covariance functions are defined as follows:

$$\boldsymbol{\mu}^c_{nest}(\boldsymbol{x}_1,\boldsymbol{x}_2) = \boldsymbol{\mu}^c_2\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right), \tag{54}$$

$$\begin{aligned} \boldsymbol{C}^c_{nest}\left((\boldsymbol{x}_1,\boldsymbol{x}_2),\left(\boldsymbol{x}_1^{'},\boldsymbol{x}_2^{'}\right)\right) &= \widehat{\boldsymbol{R}}_{t_2} C^c_2\left(\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right),\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1^{'}\right),\boldsymbol{x}_2^{'}\right)\right) \\ &+\frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\widehat{\boldsymbol{R}}_{t_3}\left(\frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1^{'}\right),\boldsymbol{x}_2^{'}\right)\right)^T C^c_3\left(\boldsymbol{x}_1,\boldsymbol{x}_1^{'}\right). \end{aligned} \tag{55}$$

The predictor obtained can also be written in the form $\boldsymbol{Y}^c_{nest} := \boldsymbol{\mu}^c_{nest} + \boldsymbol{\varepsilon}^c_{nest}$*, where:*

$$\boldsymbol{\varepsilon}^c_{nest}(\boldsymbol{x}_1,\boldsymbol{x}_2) = \frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\boldsymbol{\varepsilon}^c_3(\boldsymbol{x}_1) + \boldsymbol{\varepsilon}^c_2\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right). \tag{56}$$

The proof of this Proposition is given in Section 8. The predictor obtained is therefore Gaussian and can take into account a different number of observations for each code. Moreover, the smaller C^c_3, the more valid the linearization. Moreover, since the linearization is performed using the conditioned process $\boldsymbol{Y}^c_3$, the norm of C^c_3 is more likely to be small. Besides, the norm of C^c_3 can be reduced by an appropriate enrichment of the set of observations of the first code.

4.2 Sequential designs

In the previous section, we have proposed a method to construct a Gaussian process emulator of the nested code output for a given set of observations of the two codes. By construction, the accuracy of this predictor depends on the set of observations of the two codes. A way to improve this accuracy is an appropriate choice of the set of observations. As for the case of a single code (see Section 3), sequential design criteria or Stepwise Uncertainty Reduction (Bect et al., 2012; Picheny et al., 2010) methods are chosen. Since we want to perform a sensitivity analysis of the nested code output with respect to its inputs, the design has to lead to an accurate prediction mean on the most likely regions of the input domain of the nested code.

The two proposed design criteria are therefore a generalization of Eq. (50) to the case of two nested codes. They are also an adaptation to the case of two codes with functional outputs of the criteria proposed in Marque-Pucheu et al. (2018) for the case of two nested codes with scalar outputs. One of the criteria corresponds to the case where the two codes can be run separately, the other one corresponds to the case where they cannot be run separately. It is worth noticing that the previously proposed predictor of the nested code can take into account a different number of observations for the two codes. This property enables to define a sequential design criterion which chooses the best candidate among the two codes and takes into account the computational costs of the codes.

Definition 1 In the case of two nested codes with functional outputs, two selection criteria based on the minimization of the integrated prediction variance can be defined:

- the Chained I-optimal criterion selects the candidate in $\mathbb{X}_{\text{nest}}$ which minimizes the trace of integrated prediction variance given this candidate:

$$\begin{aligned} (\boldsymbol{x}_1^{new},\boldsymbol{x}_2^{new}) = \underset{\left(\boldsymbol{x}_1^*,\boldsymbol{x}_2^*\right)\in\mathbb{X}_{\text{nest}}}{\operatorname{argmin}} \int_{\mathbb{X}_1\times\mathbb{X}_2} &\left\{\operatorname{Tr}\left(\boldsymbol{R}_{t_2}\left(\boldsymbol{Y}_2^{\text{obs}}\right)\right) v_2\left(\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right);\overline{\boldsymbol{X}}_2^{\text{obs}},\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1^*\right),\boldsymbol{x}_2^*\right)\right) + \right. \\ &\left.\operatorname{Tr}\left(\frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\boldsymbol{R}_{t_3}\left(\boldsymbol{Y}_3^{\text{obs}}\right)\left(\frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\right)^T\right) v_3\left(\boldsymbol{x}_1;\overline{\boldsymbol{X}}_1^{\text{obs}},\boldsymbol{x}_1^*\right)\right\} d\mu_{\mathbb{X}_1}(\boldsymbol{x}_1)\, d\mu_{\mathbb{X}_2}(\boldsymbol{x}_2), \end{aligned} \tag{57}$$

  where $\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right)$, $i\in\{2,3\}$ is defined by Eq. (36), and v_i is defined by Eq. (80),
- the Best I-optimal criterion selects the candidate among the input domains of the two Gaussian processes, which maximizes the reduction of the trace of the integrated prediction variance per unit of computational cost given this candidate:

$$(i^{new},\boldsymbol{x}_{i^{new}}^{new}) = \underset{(i,\tilde{\boldsymbol{x}}_i)\in\{1,2\}\times\tilde{\mathbb{X}}_i}{\operatorname{argmin}} \frac{1}{\tau_i}\frac{n_i}{n_i+1}\mathcal{V}_i(\tilde{\boldsymbol{x}}_i), \tag{58}$$

with τ_i denoting the computational cost of the code i, and:

$$\mathcal{V}_1(\tilde{\boldsymbol{x}}_1) := \int_{\mathbb{X}_1\times\mathbb{X}_2} \mathrm{Tr}\left(\frac{\partial \boldsymbol{\mu}_2^c}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\boldsymbol{R}_{t_3}\left(\boldsymbol{Y}_3^{\mathrm{obs}}\right)\left(\frac{\partial \boldsymbol{\mu}_2^c}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1),\boldsymbol{x}_2\right)\right)^T\right) \left[v_3\left(\boldsymbol{x}_1;\overline{\boldsymbol{X}}_1^{\mathrm{obs}}\right) - v_3\left(\boldsymbol{x}_1;\overline{\boldsymbol{X}}_1^{\mathrm{obs}},\tilde{\boldsymbol{x}}_1\right)\right] d\mu_{\mathbb{X}_1}(\boldsymbol{x}_1)\, d\mu_{\mathbb{X}_2}(\boldsymbol{x}_2), \tag{59}$$

$$\mathcal{V}_2(\tilde{\boldsymbol{x}}_2) := \int_{\mathbb{X}_1\times\mathbb{X}_2} \left[v_2\left(\left(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1),\boldsymbol{x}_2\right);\overline{\boldsymbol{X}}_2^{\mathrm{obs}}\right) - v_2\left(\left(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1),\boldsymbol{x}_2\right);\overline{\boldsymbol{X}}_2^{\mathrm{obs}},\left(\boldsymbol{\mu}_3^c(\boldsymbol{x}_1^*),\boldsymbol{x}_2^*\right)\right)\right] \mathrm{Tr}\left(\boldsymbol{R}_{t_2}\left(\boldsymbol{Y}_2^{\mathrm{obs}}\right)\right) d\mu_{\mathbb{X}_1}(\boldsymbol{x}_1)\, d\mu_{\mathbb{X}_2}(\boldsymbol{x}_2), \tag{60}$$

and:

$$\{\tilde{\boldsymbol{x}}_i, \tilde{\mathbb{X}}_i\} = \begin{cases} \{\boldsymbol{x}_1^*, \mathbb{X}_1\}, \text{ if } i=1, \\ \{(\boldsymbol{x}_1^*,\boldsymbol{x}_2^*)\ ,\ \mathbb{X}_1\times\mathbb{X}_2\}, \text{ if } i=2. \end{cases} \tag{61}$$

Moreover, if a candidate $\tilde{\boldsymbol{x}}_2 = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*)$ is chosen, the second code will be evaluated at $(\boldsymbol{\mu}_1^c(\boldsymbol{x}_1^*), \boldsymbol{x}_2^*)$.

Note that the Best I-optimal criterion is based on the assumption that the integrated variance decreases linearly at each step. Both criteria imply a multidimensional integration and optimization on $\mathbb{X}_1$ or $\mathbb{X}_1 \times \mathbb{X}_2$. The integration is computed using the empirical average of a Monte-Carlo draw. The optimization is performed on a finite set of candidates drawn according to $\mu_{\mathbb{X}_1}$ or $\mu_{\mathbb{X}_1\times\mathbb{X}_2}$. The fact that the prediction variance has a closed-form expression, and is thus fast to evaluate, is an advantage for the computation of the criteria. Indeed, they both require a high number of evaluations of the prediction variance.

The estimation of the projection basis, the hyperparameters of the covariance functions $C_i,\ i \in \{1,2,3\}$, the functional covariance matrices $\widehat{\boldsymbol{R}}_{t_i},\ i \in \{1,2,3\}$, as well as the conditioned mean functions $\boldsymbol{\mu}_i^c,\ i \in \{1,2,3\}$, require an initial design of experiments. Once again, since we aim at predicting the nested code output on its input set $\mathbb{X}_{\mathrm{nest}}$ and we have no *a priori* information, the initial set of observations is drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$.

The projection basis associated with r_m is estimated from the initial set of observations. The dimension m of the projection of the functional input of the second code is chosen according to the criterion of Eq. (42). The hyperparameters of the covariance function C_3 are estimated from the initial design and when a new observation of the first code is added. The hyperparameters of the covariance functions C_1 and C_2 are estimated from the initial design and when a new observation of the second code is added. The hyperparameters are estimated thanks to the Cross Validation criterion of Eq. (40).

5 First test case

In this section we apply the proposed methods to a test case.

5.1 Description of the test case

This first test case is the coupling of two codes: a detonation code and a damped oscillator. The second code has a closed-form expression and is thus very fast to evaluate. The first code is a quick and simplified version of a detonation code. The two codes of this test case are therefore fast to evaluate, which enables to repeat the draws and the validation of the proposed methods, thus quantifying their performance. Moreover, their features are representative of the features of the industrial problem which motivates this work:

- the first code has a low dimensional vectorial input and a functional output,
- the second code has low and high dimensional vectorial inputs and a functional output,
- the functional input of the second code is the output of the first code,
- the functional output of the first code can be non smooth.

The inputs of the first code have a uniform distribution on $\mathbb{X}_1$, with:

$$\mathbb{X}_1 = [0.01, 0.02] \times [-1, 1]^3. \tag{62}$$

Input name	Description and unit	Distribution of the input
$(\boldsymbol{x}_1)_1$	Radius of the explosive charge (m)	$\mathcal{U}([0.01, 0.02])$
$(\boldsymbol{x}_1)_2$	Temporal dilatation parameter (no unit)	$\mathcal{U}([-1, 1])$
$(\boldsymbol{x}_1)_3$	Shock magnitude parameter (no unit)	$\mathcal{U}([-1, 1])$
$(\boldsymbol{x}_1)_4$	Attenuation parameter (no unit)	$\mathcal{U}([-1, 1])$

Table 1 First test case: Input parameters of the first code. All variables are independent.

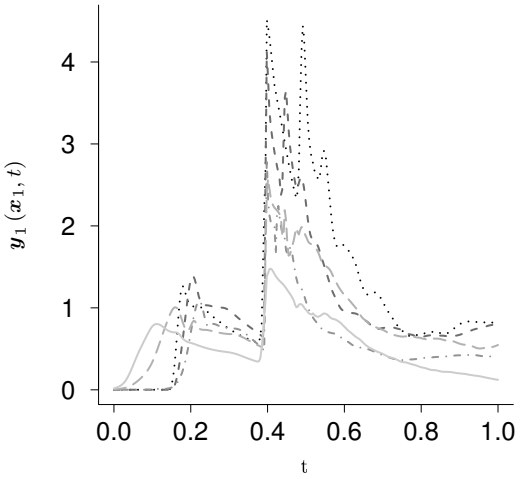


Fig. 1 First test case: Output of the first code for five different values of $\boldsymbol{x}_1$.

Table 1 presents the input variables of the first code and their distributions.

Figure 1 shows several trajectories of the functional output of the first code. The trajectories have a similar shape, but their magnitudes can vary considerably depending on the inputs of the code, $\boldsymbol{x}_1$.

The second code is a damped oscillator, which is defined by the following second order linear differential equation:

$$\ddot{\boldsymbol{y}}_2 + 2\,(\boldsymbol{x}_2)_1\,\big((\boldsymbol{x}_2)_2 + \omega_0\big)\,\dot{\boldsymbol{y}}_2 + \big((\boldsymbol{x}_2)_2 + \omega_0\big)^2\,\boldsymbol{y}_2 = \boldsymbol{y}_1.$$

This damped oscillator is characterized by an angular frequency ω_0. Two frequencies are studied: $\omega_0 \in \{2\pi, 20\pi\}$. These two values of ω_0 correspond to qualitatively different relationships between the two codes (low-pass filter and high-pass filter).
The inputs of the second code have a uniform distribution on $\mathbb{X}_2$, with:

$$\mathbb{X}_2 = [0, 0.2] \times [-0.1, 0.1]\,. \tag{63}$$

Consequently, the definition domain of the nested code is:

$$\mathbb{X}_{\text{nest}} = [0.01, 0.02] \times [-1, 1]^3 \times [0, 0.2] \times [-0.1, 0.1]\,. \tag{64}$$

Moreover the basis functions of the mean function of the Gaussian processes are defined as follows:

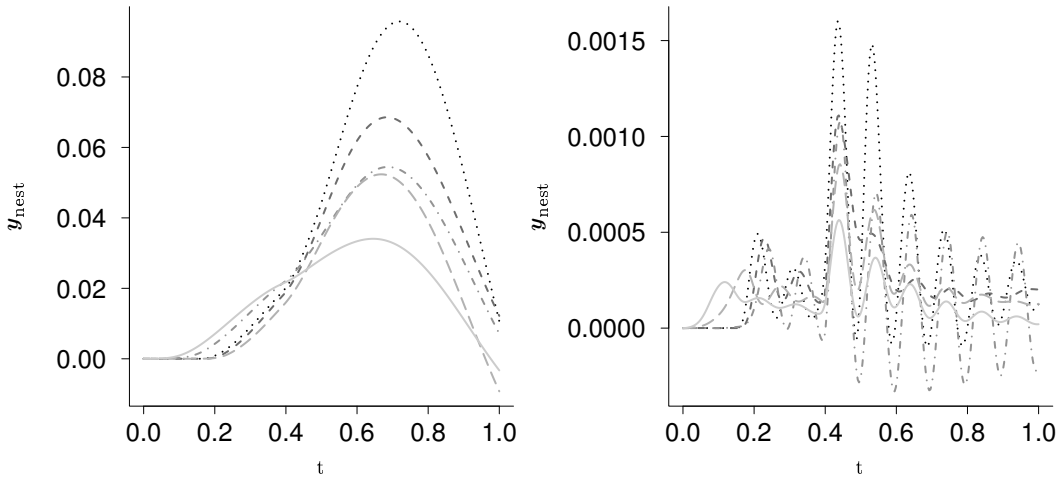
$$\boldsymbol{h}_i\,(\bar{\boldsymbol{x}}_i) = (1, \bar{\boldsymbol{x}}_i)\,. \tag{65}$$

Note that an increase in the order of the polynomials does not yield more accurate prediction.

Figure 2 shows several trajectories of the output of the second code for the two studied frequencies. For a given frequency, the trajectories have a similar shape, but their magnitudes vary depending on the inputs of the code.

5.2 Estimation of the proposed projection basis

In this section, we study the accurate estimation of the proposed basis from a small number of observations of the two codes.



(a) $\omega_0 = 2\pi$ (b) $\omega_0 = 20\pi$

Fig. 2 First test case: Output of the nested code for five values of $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and two values of ω_0.

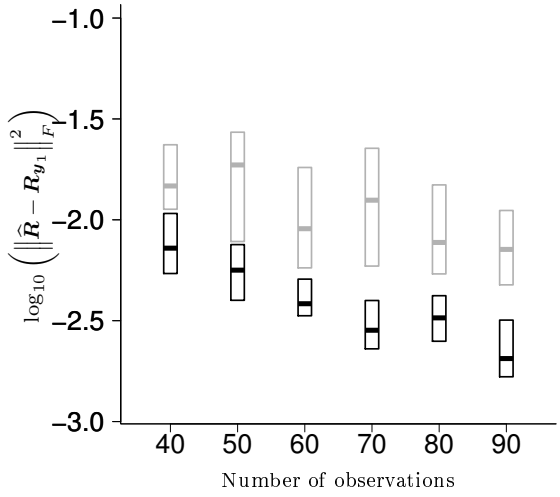


Fig. 3 First test case: Boxplots of the estimation error of the covariance matrix of the output of the first code. The estimation error is given by $\left\|\widehat{\boldsymbol{R}} - \boldsymbol{R}_{\boldsymbol{y}_1}\right\|_F^2$ where $\boldsymbol{R}_{\boldsymbol{y}_1}$ is the reference covariance matrix and $\widehat{\boldsymbol{R}}$ is the estimate of the covariance matrix. The reference covariance matrix is estimated from 10^3 observations of the first code output. The covariance matrices have been estimated from the observations (grey) of from the mean of a Gaussian predictor constructed from the observations (see Eq. (15)) (black). The design sets of increasing size contain observations which are independently drawn according to $\mu_{\mathbb{X}_1}$.

5.2.1 Estimation of the covariance matrix of the first code output

Figure 3 compares the error of the estimation of the covariance matrix of the output of the first code with an estimation from the observations only or from the mean of a Kriging predictor constructed using the observations (see Eq. (15)). The estimation error is given by $\left\|\widehat{\boldsymbol{R}} - \boldsymbol{R}_{\boldsymbol{y}_1}\right\|_F^2$ where $\boldsymbol{R}_{\boldsymbol{y}_1}$ is the reference covariance matrix and $\widehat{\boldsymbol{R}}$ is the estimate of the covariance matrix. The reference covariance matrix is estimated from 10^3 observations of the output of the first code. The figure shows that the estimation with the predictor of the first code is more accurate than the estimation from the small set of observations.

Angular frequency	Regularization matrix defined in Eq. (68)	Regularization matrix defined in Eq. (69)
$\omega_0 = 2\pi$	0.2	0.05
$\omega_0 = 20\pi$	1.3	0.6

Table 2 First test case: Order of magnitude of the error $\delta_2 = \frac{\|\boldsymbol{y}_2 - L(\boldsymbol{a}^*)\,\boldsymbol{y}_1\|^2}{\|\boldsymbol{y}_2 - \bar{\boldsymbol{y}}_2\|^2}$ depending on the regularization matrix used for the estimation of $\boldsymbol{a}$: the one defined in Eq. (68) or the one defined in Eq. (69). $\boldsymbol{a}^*$ is estimated from the draws according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$ of increasing size, repeated 20 times. The accuracy does not vary significantly with the size of the set of observations.

5.2.2 Estimation of the linear approximation

Given that $\sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T L\left(\boldsymbol{y}_1^{(i)}\right)$ is not always invertible, the estimation of the vector $\boldsymbol{a}$ given by Eq. (9) is not always numerically stable. In what follows, we study two methods of regularization of this matrix in order to invert it. In the case of a L_2 regularization, the estimate of $\boldsymbol{a}^*$ given by Eq. (9) can be rewritten :

$$\boldsymbol{a}^* = \boldsymbol{a}\left(\boldsymbol{D}, \boldsymbol{Y}_1^{\text{obs}}, \boldsymbol{Y}_2^{\text{obs}}\right) = \left(\sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T L\left(\boldsymbol{y}_1^{(i)}\right) + \boldsymbol{D}\right)^{-1} \sum_{i=1}^{n} L\left(\boldsymbol{y}_1^{(i)}\right)^T \boldsymbol{y}_2^{(i)}, \tag{66}$$

with $\boldsymbol{D}$ a positive-definite $(N_t \times N_t)$-dimensional matrix. Consequently, the following optimization problem is solved:

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}\in\mathbb{R}^{N_t}}{\text{argmin}} \sum_{i=1}^{n} \left\|\boldsymbol{y}_2^{(i)} - L\left(\boldsymbol{y}_1^{(i)}\right)\boldsymbol{a}\right\|^2 + \left\|\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{a}\right\|^2. \tag{67}$$

Two parametric forms are studied for the definition of the matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \delta \boldsymbol{I}_{N_t},\ \delta \in \mathbb{R}_+, \tag{68}$$

and:

$$\boldsymbol{D} = \text{diag}\left(\beta_0 + \beta_1\left(\frac{1}{N_t}, \frac{2}{N_t}, \ldots, \frac{N_t - 1}{N_t}, 1\right)\right),\ \beta_0 \in \mathbb{R}_+,\ \beta_1 \in \mathbb{R}_+. \tag{69}$$

This second increasing regularization is proposed because the considered system is a damped system, and thus $i \mapsto a_i$ is a decreasing function.
By noting that $L(\boldsymbol{a})\,\boldsymbol{y}_1 = L(\boldsymbol{y}_1)\,\boldsymbol{a}$, the parameters δ, β_0 and β_1 can be estimated with Cross Validation criteria of the form:

$$\underset{\delta\in\mathbb{R}_+}{\text{argmin}} \sum_{i=1}^{n} \left\|\boldsymbol{y}_2^{(i)} - L\left(\boldsymbol{y}_1^{(i)}\right)\boldsymbol{a}\left(\delta \boldsymbol{I}_{N_t}, \left(\boldsymbol{Y}_1^{\text{obs}}\right)_{-i}, \left(\boldsymbol{Y}_2^{\text{obs}}\right)_{-i}\right)\right\|^2, \tag{70}$$

and:

$$\underset{\beta_0,\beta_1\in\mathbb{R}_+^2}{\text{argmin}} \sum_{i=1}^{n} \left\|\boldsymbol{y}_2^{(i)} - L\left(\boldsymbol{y}_1^{(i)}\right)\boldsymbol{a}\left(\text{diag}\left(\beta_0 + \beta_1\left(\frac{1}{N_t}, \ldots, 1\right)\right), \left(\boldsymbol{Y}_1^{\text{obs}}\right)_{-i}, \left(\boldsymbol{Y}_2^{\text{obs}}\right)_{-i}\right)\right\|^2, \tag{71}$$

with $\left(\boldsymbol{Y}_1^{\text{obs}}\right)_{-i}$ and $\left(\boldsymbol{Y}_2^{\text{obs}}\right)_{-i}$ denoting the observations of the functional input and output of the second code, except the i-th.

Table 2 shows the error $\frac{\|\boldsymbol{y}_2 - L(\boldsymbol{a}^*)\,\boldsymbol{y}_1\|^2}{\|\boldsymbol{y}_2 - \bar{\boldsymbol{y}}_2\|^2}$ of the linear sparse model compared to the output of the code $\boldsymbol{y}_2$. The two regularization methods of Eqs. (68) and (69) are studied. The linear regularization of Eq. (69) leads to the most accurate prediction of $\boldsymbol{y}_2$ by the linear model.
The regularization matrix $\boldsymbol{D}$ of Eq. (69) will therefore be chosen for the estimation of $\boldsymbol{a}$ in the remainder of the numerical examples. Indeed, the more accurate the linear model is, the more accurate the projection basis based on this linear model will be (see Proposition 1).

Angular frequency	PLS	Proposed projection basis
$\omega_0 = 2\pi$	7	4
$\omega_0 = 20\pi$	6	6

Table 3 Median of the sizes of the projection basis estimated with the Leave One Out criterion of Eq. (42) from 20 sets of 40 observations.

5.3 Influence of the dimension reduction method on the accuracy of the prediction of the second code

In this section, we study the influence of the dimension reduction method on the accuracy of the prediction of the second code.
We assume that a set of N validation points of the codes is available. The prediction errors of the second code and the nested code can thus be defined by:

$$\Delta_2 = \frac{\sum_{i=1}^{N} \left\| \boldsymbol{y}_2\left(\boldsymbol{\varphi}_1^{(val,i)}, \boldsymbol{x}_2^{(val,i)}\right) - \boldsymbol{\mu}_2^c\left(r_m\left(\boldsymbol{\varphi}_1^{(val,i)}\right), \boldsymbol{x}_2^{(val,i)}\right)\right\|^2}{\sum_{i=1}^{N} \left\| \boldsymbol{y}_2\left(\boldsymbol{\varphi}_1^{(val,i)}, \boldsymbol{x}_2^{(val,i)}\right) - \frac{1}{N_2}\sum_{j=1}^{N} \boldsymbol{y}_2\left(\boldsymbol{\varphi}_1^{(val,j)}, \boldsymbol{x}_2^{(val,j)}\right)\right\|^2}, \tag{72}$$

and

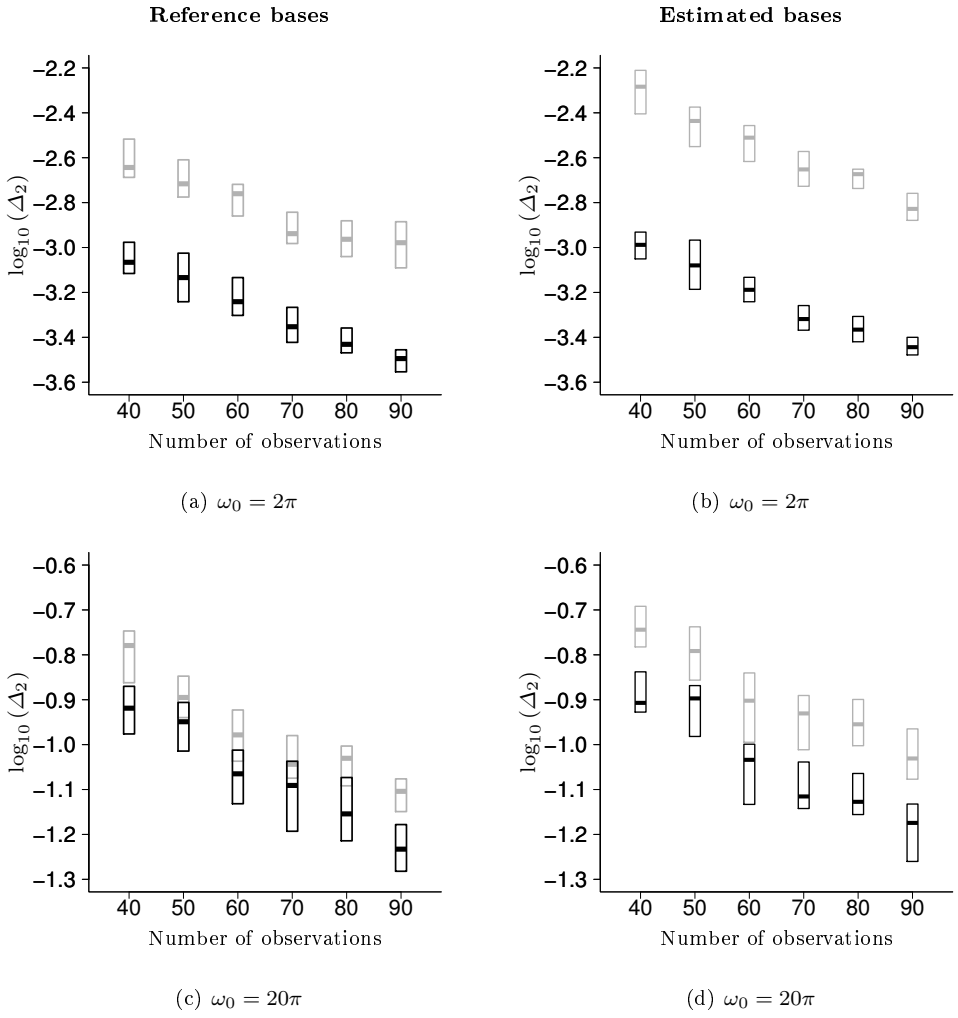
$$\Delta_{\text{nest}} = \frac{\sum_{i=1}^{N} \left\| \boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(val,i)}\right) - \boldsymbol{\mu}_{\text{nest}}^c\left(\boldsymbol{x}_{\text{nest}}^{(val,i)}\right)\right\|^2}{\sum_{i=1}^{N} \left\| \boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(val,i)}\right) - \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(val,j)}\right)\right\|^2}, \tag{73}$$

where $\boldsymbol{x}_{\text{nest}}^{(val,j)} = \left(\boldsymbol{x}_1^{(val,j)}, \boldsymbol{x}_2^{(val,j)}\right)$ and $\boldsymbol{\varphi}_1^{(val,j)} = \boldsymbol{y}_1\left(\boldsymbol{x}_1^{(val,j)}\right)$. The observations of the inputs $\{\boldsymbol{x}_{\text{nest}}^{(val,j)}, 1 \leq j \leq N\}$ are independently drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$.

Figure 4 shows the prediction error Δ_2 (defined by Eq. (72)) of the Gaussian predictor $\boldsymbol{Y}_2^c$ of the second code. Two projection functions r_m are studied (see Eqs. (19) and (18)). The first one is based on a PLS regression. The second one is based on the matrix $\boldsymbol{Z}^*$ of Proposition 1 with a linear model of the form $L(\boldsymbol{a})\,\boldsymbol{y}_1$ and $\boldsymbol{a}$ estimated from Eq. (66) with the regularization matrix of Eq. (69). Two cases are considered for the estimation of the projection bases. The latter are estimated either from a large set of 10^3 observations of the functional input and output of the second code or from the observations used for the construction of the predictor. Finally, in all cases, the size of the projection basis is estimated with the Leave One Out criterion of Eq. (42).
The dimension reduction of Proposition 1 based on the linear model of the form $L(\boldsymbol{a})\,\boldsymbol{y}_1$ leads to the most accurate prediction, whether the basis is estimated from a small or a high number of observations.
Besides, the accuracy of the prediction with the dimension reduction based on Proposition 1 with the linear model of the form $L(\boldsymbol{a})\,\boldsymbol{y}_1$ is quite the same if the projection basis is estimated from a small number of observations or from a high number of observations. Thanks to the increase in the rank of the covariance matrix of $\boldsymbol{y}_1$ with a predictor of the first code and the sparse structure of the linear model, the proposed projection basis can be estimated accurately from a relatively small number of observations.
In contrast, the projection basis based on PLS is much more sensitive to the decrease in the number of observations from which it is estimated.

Table 3 shows the median of the sizes of the bases estimated from the set of observations of 40 observations with the Cross Validation criterion of Eq. (42) for the proposed and the PLS dimension reduction methods. The size of the basis varies less and is smaller with the dimension reduction of Proposition 1 with a linear model of the form $L(\boldsymbol{a}^*)\,\boldsymbol{y}_1$ where $\boldsymbol{a}^*$ is given by Eqs. (66) and (69) than with the dimension reduction based on PLS.

In the remainder of the section, the dimension reduction of the functional intermediary variable will be performed using the method defined in Proposition 1. A linear model of the form $L(\boldsymbol{a})\,\boldsymbol{y}_1$ will be considered, and $\boldsymbol{a}$ will be estimated using Eqs. (66) and (69). Indeed, it has been shown that this method leads to the most accurate linear approximation of $\boldsymbol{y}_2$. The more accurate the linear model is, the more appropriate the dimension reduction based on this linear model will be.
The covariance matrix of the functional input of the second code will be computed using a predictor of the output of the first code, as defined in Eq. (15). Finally, the size m of the projection basis is estimated with the Cross Validation criterion of Eq. (42).



(a) $\omega_0 = 2\pi$ (b) $\omega_0 = 2\pi$

(c) $\omega_0 = 20\pi$ (d) $\omega_0 = 20\pi$

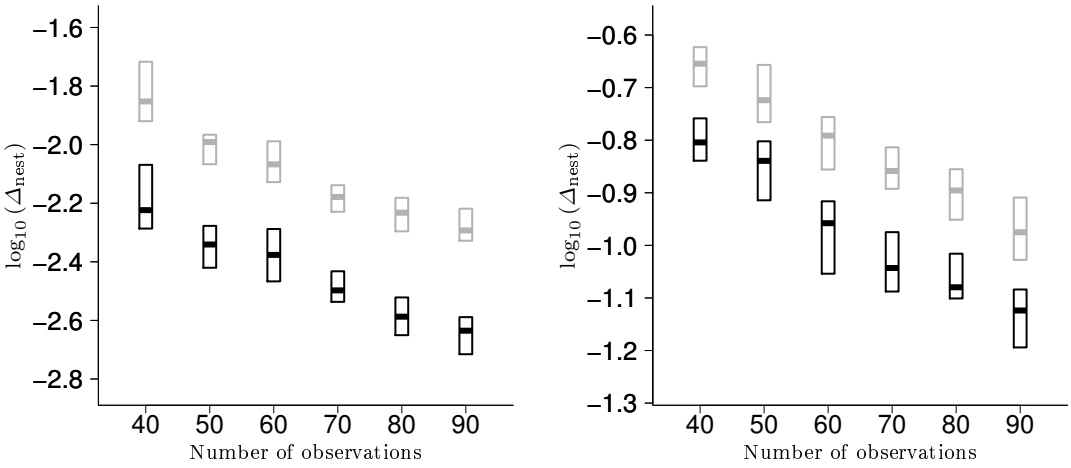
Fig. 4 First test case: Prediction error of the output of the second code Δ_2 (see Eq. (72)) in log scale. The observations are independently drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$ and the design sets are of increasing size. Two types of projection functions r_m are considered. The first one is based on PLS regression (grey) and the second one is based on the projection of Proposition 1 with a linear model of the form $L(\boldsymbol{a}^*)\,\boldsymbol{y}_1$ where $\boldsymbol{a}^*$ is given by Eqs. (66) and (69) (black). The size of the projection basis is estimated with the Leave One Out criterion of Eq. (42). The reference bases are computed from a set of 1000 observations of the functional input and output of the second code. The estimated bases are estimated from the observations of the design sets.

5.4 Prediction of the nested code

Once the dimension of the functional intermediary variable has been efficiently reduced, we can study the surrogate modeling of the nested code.

We first define the reference method which corresponds to the case where the nested code is considered to be a single code. This reference method is called "blind-box". With this method, a surrogate model of $\boldsymbol{x}_{\text{nest}} \mapsto \boldsymbol{y}_{\text{nest}}$ is constructed using the formalism of Section 3. This blind-box method cannot take into account the intermediary observations of the nested code. The "linearized" method refers to the construction of the predictor of the nested code proposed in Proposition 4.

Figure 5 compares the prediction error Δ_{nest} (see Eq. (73)) for the predictors of the nested code constructed with the linearized and the blind-box methods. The prediction accuracy is better with the linearized method.



(a) $\omega_0 = 2\pi$ (b) $\omega_0 = 20\pi$

Fig. 5 First test case: Boxplots of the prediction error Δ_{nest} (see Eq. (73)) for the linearized (black) and the blind-box (grey) predictors. The observations are drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$ and the design sets are of increasing size. The draws are repeated 20 times. The dimension reduction of the intermediary variable is performed using the method of Proposition 1 and a linear approximation of the form $L(\boldsymbol{a}^*)\,\boldsymbol{y}_1$ where $\boldsymbol{a}^*$ is given by Eqs. (66) and (69).

5.5 Sequential designs

Figure 6 shows the prediction error Δ_{nest} (see Eq. (73)) along the sequential designs of Definition 1. A reference corresponding to the case of the blind-box predictor with an I-optimal sequential design (see Eq. (50) and Proposition 2) is also shown. All the sequential designs have the same initial design sets drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$.
It can be seen that both proposed sequential designs with the linearized predictor enable to obtain a better prediction accuracy than the blind-box predictor with an I-optimal design. Furthermore, the Best I-optimal criterion leads to an improved prediction accuracy compared to the one obtained with the Chained I-optimal design. Besides, it is worth noticing that the Best I-optimal criterion leads to different numbers of code evaluations for the two codes.

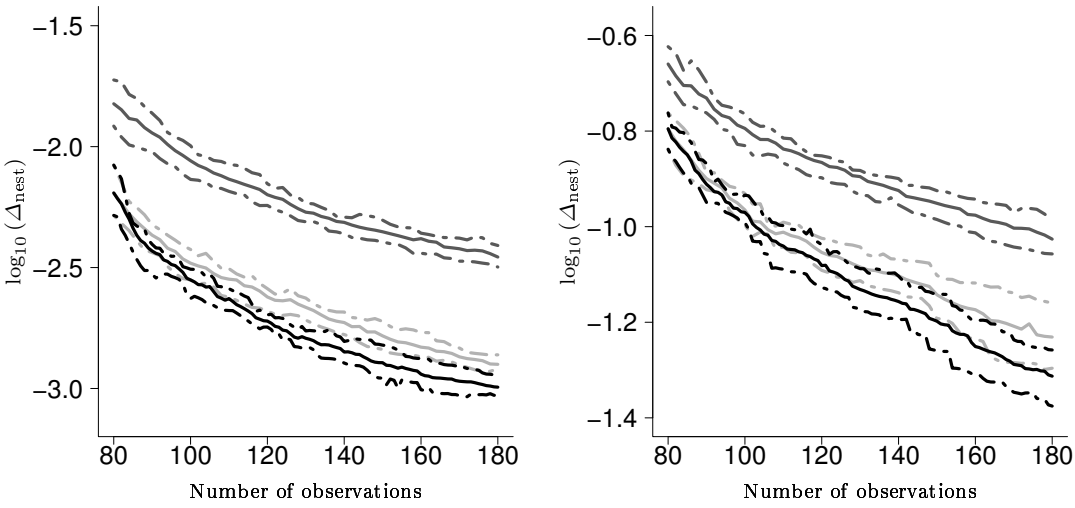
Figure 7 shows the distribution of the number of code evaluations of the two codes along the Best I-optimal design. The first new observation points are observations of the first code. However, the distribution of the number of observations of the two codes can vary a lot depending on the angular frequency of the oscillator of the second code. For the angular frequency equal to 2π, there are more evaluations of the first code. For the angular frequency equal to 20π, the number of evaluations is almost the same for both codes at the beginning, but then the number of observations of the second code increases faster than those of the first code.

6 Second test case

6.1 Description of the test case

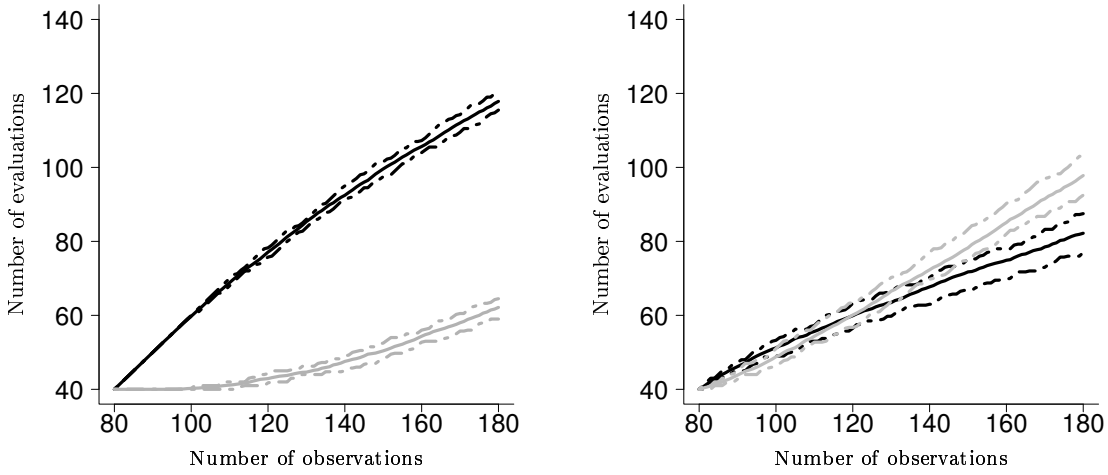
This second study deals with the motivating industrial problem exposed in Defaux and Evrard (2014) and adapted to this field of research. This industrial problem is the study of the mechanical response of a spherical tank subjected to an internal blast. The problem is model by the coupling of two codes and we aim at performing a sensitivity analysis of the output of the nested code with respect to its inputs.

This first code is an Eulerian 2-dimensional code, which simulates the burn of the explosive charge, the internal gas dynamics and the shock wave propagation. Its inputs are the same as those of the first code of the first test case (see Table 1). The second code provides the mechanical response of the tank through the



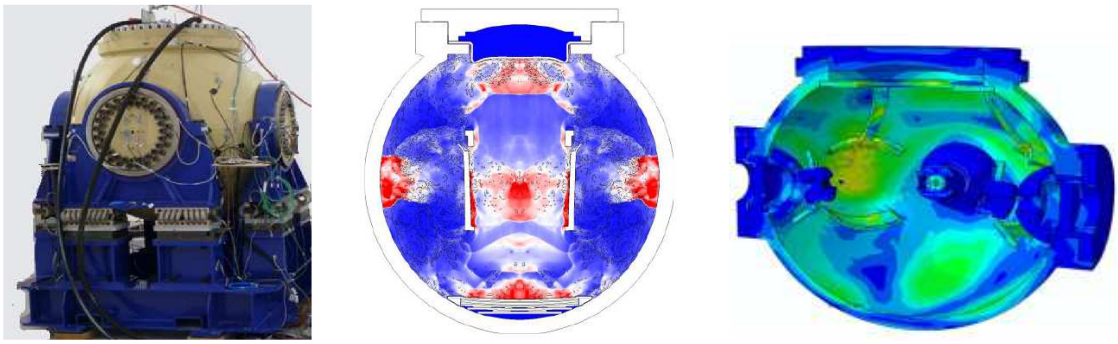
(a) $\omega_0 = 2\pi$ (b) $\omega_0 = 20\pi$

Fig. 6 First test case: Error of the nested code prediction along the sequential designs, in log scale. The initial design sets are 40-points design sets, drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$ and are the same for the three series. The draw of the initial design is repeated 20 times. The solid lines represent the median and the two-dashed lines represent the first and the third quartiles. The three series correspond to an I-optimal design with a blind-box predictor (mid-grey), the Chained I-optimal design with a linearized predictor (light grey) and the Best I-optimal design with a linearized predictor (black). The computational cost of both codes is the same: $\tau_1 = \tau_2$.



(a) $\omega_0 = 2\pi$ (b) $\omega_0 = 20\pi$

Fig. 7 First test case: Number of evaluations of the first code (black) and the second code (grey) along the Best I-optimal sequential designs of Figure 6. The solid lines represent the median and the two dashed-lines represent the first and the third quartiles.



(a) Tank (b) First code (c) Second code

Fig. 8 Second test case: Images from Defaux and Evrard (2014). The left-hand plot shows the tank. The middle plot shows the output of the first code, with the detonation products in white and red. The right-hand plot shows the output of the second code, the Von Mises stress at a given time at the inner surface of the tank. The areas in green are under stress and those in blue are not under stress.

Input name	Description and unit	Distribution of the input
$(\boldsymbol{x}_2)_1$	Internal radius of the tank (m)	$\mathcal{N}\left(0.72, 0.005^2\right)$
$(\boldsymbol{x}_2)_2$	Thickness (m)	$\mathcal{N}\left(0.073, 0.0015^2\right)$
$(\boldsymbol{x}_2)_3$	Young modulus of the tank (Pa)	$\mathcal{N}\left(2.1\ 10^{11}, \left(2.1\ 10^{10}\right)^2\right)$
$(\boldsymbol{x}_2)_4$	Young modulus of the tap (Pa)	$\mathcal{N}\left(2.1\ 10^{11}, \left(2.1\ 10^{10}\right)^2\right)$

Table 4 Input parameters of the second code in the second test case. All variables are independent.

von Mises stress at the inner surface of the tank.

The spherical tank is presented in Figure 8(a). The output of the first code is shown on Figure 8(b). Figure 8(c) shows the second code output.

In this work, in order to reduce the computational cost of a code evaluation, we use a 1-dimensional approximation of the first code, by considering that the explosion and the tank are spherically symmetric. The output of the first code corresponds here to the spherically symmetric time-varying shock-wave.
In the same way, a 2-dimensional approximation of the second code is considered. The approximation is based on an axisymmetry of the mechanical response of the tank. Then the maximum value of the Von Mises stress over the surface (in fact a curve) for each time step is considered. The time-varying output of the second code is therefore this time-varying maximal value.
Moreover, the elastic limits of the tank and the tap have been set to an artificially high value in order to obtain unsaturated output trajectories.

The components of $\boldsymbol{x}_2$ are independent and of dimension 4. Table 4 presents the input variables of the second code and their distributions.

Figure 9 shows some trajectories of the output of the nested code for this second test case. The trajectories have very similar shapes and are overall growing with time.

6.2 Results of the test case

Figure 10 shows the prediction accuracy for the output of the nested code along the sequential designs. Three cases are presented. The first one is based on the blind-box predictor with an I-optimal enrichment (see Proposition 2), the other two are based on the linearized predictor of Proposition 4 with the two sequential design criteria of Definition 1. The results show that the linearized predictor is more accurate than the blind box predictor. These results illustrate the interest of taking into account the intermediary observations of the nested code.

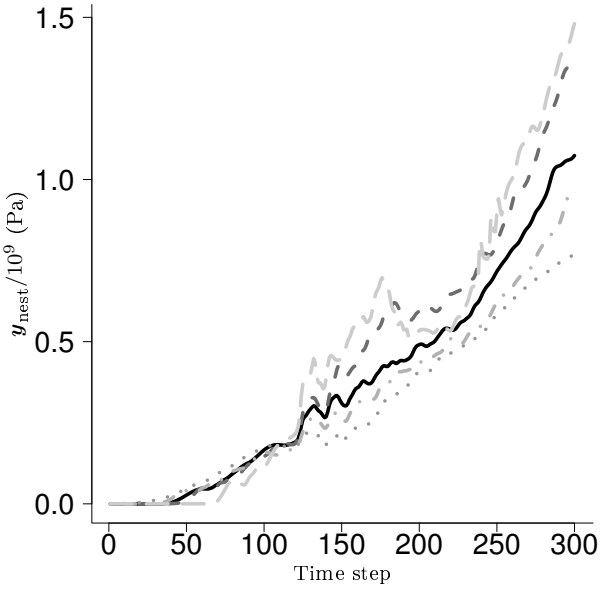


Fig. 9 Second test case: Example of trajectories of the output of the nested code for five values of $(\boldsymbol{x}_1, \boldsymbol{x}_2)$.

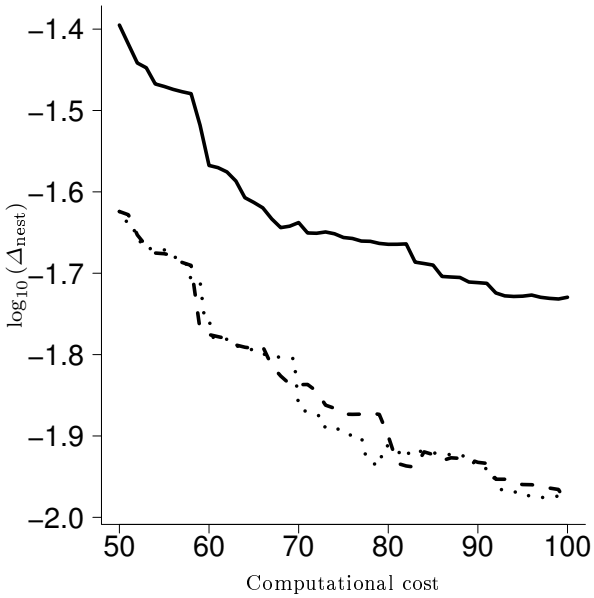


Fig. 10 Second test case: Prediction error Δ_{nest} (see Eq. (73)), in log scale, for the nested code along the sequential designs: I-optimal sequential design with the blind-box predictor (solid line), Chained I-optimal sequential design with the linearized predictor (dashed line), Best I-optimal sequential design with the linearized predictor (dotted line). The initial design of 50 points on $\mathbb{X}_{\text{nest}}$ is drawn according to $\mu_{\mathbb{X}_1} \times \mu_{\mathbb{X}_2}$.

Once an accurate predictor of the nested code has been obtained at the end of the sequential design (a Best I-optimal design with the linearized predictor), the predictor can be used in order to perform a sensitivity analysis (see Sobol (1993) for further details). A Kriging Believer approach (Picheny et al., 2010) is used, which means that $\boldsymbol{y}_{\text{nest}}$ is replaced by $\boldsymbol{\mu}^c_{\text{nest}}$ for the computation of the Sobol indices of the maximum value of the high dimensional vectorial output $\boldsymbol{y}_{\text{nest}}$.

The left-hand plot of Figure 11 shows the Sobol indices of the maximum value of the time-varying output of the nested code with respect to the inputs of the nested code. The first-order and total indices are very close.

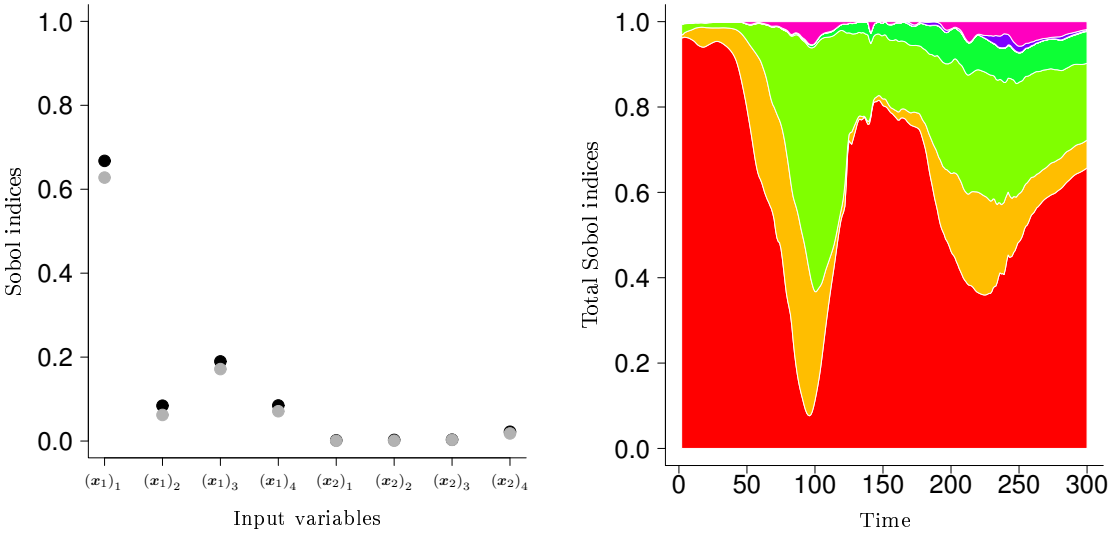


Fig. 11 Second test case: on the left figure, the Sobol indices (first-order in grey and total in black) of the maximum value of the functional output of the nested code. The indices are estimated with a predictor constructed from the set of observations at the end of the Best I-optimal sequential design. On the right plot, the first-order indices at each time step of the functional input of the nested code. At each time,the total indices are computed and normalized such that their sum is equal to 1. The figure presents the normalized indices at each time, stacked from bottom to top in the following order for the associated input variables: $(\boldsymbol{x}_1)_1, \ldots, (\boldsymbol{x}_1)_4, (\boldsymbol{x}_2)_1, \ldots, (\boldsymbol{x}_2)_4$

This means that the first-order effects are important and the interactions almost negligible (see (Sobol, 1993) for further details). Consequently, $\max\left(\boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}\right)\right)$ can be accurately approximated by an approximation of the form:

$$f_0 + \sum_{i=1}^{d_1+d_2} f_i\left(\left(\boldsymbol{x}_{\text{nest}}\right)_i\right), \tag{74}$$

where f_0 and f_i are defined by:

$$f_0 = \mathbb{E}\left[\max\left(\boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}\right)\right)\right], \tag{75}$$

$$f_i = \mathbb{E}\left[\max\left(\boldsymbol{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}\right)\right) \mid \left(\boldsymbol{x}_{\text{nest}}\right)_i\right] - f_0. \tag{76}$$

Besides, the indices of the first input of the first code $\boldsymbol{x}_1$ (radius of the explosive charge) are very high compared to those of the other inputs. The radius of the explosive charge is thus a very influential input for the maximum value of the Von Mises stress over the inner surface of the tank, over the time.
Besides, the indices associated with the inputs $\boldsymbol{x}_2$ of the second code are very low compared to those associated with the inputs $\boldsymbol{x}_1$. The second most influential input is the shock magnitude parameter $(\boldsymbol{x}_1)_3$.

The right-hand plot of Figure 11 shows the total Sobol indices for each time index of the output. The indices have been normalized such that their sum at each time index is equal to 1. Note that the first-order Sobol indices were very close to the total Sobol indices. The figure shows that the four input variables of the first code are the most influential input variables. They explain more than 90% of the variance at each time step.

7 Conclusions

In this paper, we have focused our attention on a system of two nested codes with functional outputs, where the functional output of the first code is one of the inputs of the second code. The objective was to construct a surrogate model of the output of the nested code in order to perform a sensitivity analysis.
We aimed at extending the approach proposed in Marque-Pucheu et al. (2018) for the case of two nested codes with scalar outputs. This latter is based on the coupling of the Gaussian predictors of the two codes.

The first challenge posed by this objective was the surrogate modeling of the second code, which has functional input and output. Indeed, the existing methods for the Gaussian process regression of a code are adapted to the case of a low dimensional vectorial input.

In order to build a predictor of the second code, we have proposed an efficient dimension reduction method of the functional input of the second code. This dimension reduction significantly outperforms the PLS when it is used for the Gaussian process regression of a computer code with a functional input.
Then a Gaussian predictor of the nested code has been obtained by composing the predictors associated with the two codes and linearizing this composition. This predictor can take into account all the observations. Moreover, it is Gaussian with fast to compute mean and variance.
Finally, two sequential designs aiming at improving the prediction accuracy have been proposed. The possibility of choosing which code to run enables an optimal allocation of the computational budget between the two codes and substantially reduces the prediction error.
The application to numerical examples has shown the interest of the proposed methods. The proposed dimension reduction method significantly improves the accuracy of the Gaussian predictor of the second code compared to a dimension reduction based on PLS, and is more robust when only a limited number of observations can be used for the estimation of the projection basis.
Moreover, the proposed predictor of the nested code is more accurate than a simple Gaussian process regression of the nested code output without considering the intermediary variable. This shows the interest of taking into account the observations of the intermediary variable.

8 Appendix

Proof of Proposition 1

We aim at finding a m-rank matrix $\boldsymbol{Z}^*$ such that:

$$\boldsymbol{Z}^* = \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \sum_{i=1}^{N_x} \left\| \boldsymbol{A}\boldsymbol{x}_t^{(i)} - \boldsymbol{A}\boldsymbol{Z}\boldsymbol{x}_t^{(i)} \right\|^2$$

where $\boldsymbol{Z}$ is a m-rank matrix and $\boldsymbol{x}_t^{(i)}$ denotes the i-th observation of the input.
The previous equation can be rewritten:

$$\begin{aligned}
\boldsymbol{Z}^* &= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \sum_{i=1}^{N_x} \left\| \boldsymbol{A}\boldsymbol{x}_t^{(i)} - \boldsymbol{A}\boldsymbol{Z}\boldsymbol{x}_t^{(i)} \right\|^2, \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\boldsymbol{A}\boldsymbol{X}_t - \boldsymbol{A}\boldsymbol{Z}\boldsymbol{X}_t\right)^T \left(\boldsymbol{A}\boldsymbol{X}_t - \boldsymbol{A}\boldsymbol{Z}\boldsymbol{X}_t\right) \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\boldsymbol{A}\boldsymbol{Z}^c\boldsymbol{X}_t\right)^T \left(\boldsymbol{A}\boldsymbol{Z}^c\boldsymbol{X}_t\right) \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\boldsymbol{X}_t^T \left(\boldsymbol{Z}^c\right)^T \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{Z}^c\boldsymbol{X}_t \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\boldsymbol{Z}^c\right)^T \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{Z}^c\boldsymbol{X}_t\boldsymbol{X}_t^T \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \left(\boldsymbol{Z}^c\right)^T \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{Z}^c \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \left(\boldsymbol{Z}^c\right)^T \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z}^c \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \operatorname{Tr}\left(\left(\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z}^c \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right)^T \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z}^c \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right), \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \left\| \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z}^c \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right\|_F^2, \\
&= \underset{\boldsymbol{z}\in\mathcal{M}_{N_t\times N_t}}{\operatorname{argmin}} \left\| \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} - \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} \right\|_F^2.
\end{aligned}$$

In (Eckart and Young, 1936), it is shown that the matrix $\boldsymbol{Q}_m$ of rank m which minimizes $\left\| \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} - \boldsymbol{Q}_m \right\|_F^2$ is given by the m first singular-values of the singular-value decomposition of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}}$. If we denote by $\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ the singular-value decomposition of $\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}}$, with $\boldsymbol{U}$ and $\boldsymbol{V}$ gathering the singular vectors associated with the singular values sorted in ascending order, then we have $\boldsymbol{Q}_m = \boldsymbol{U}_m\boldsymbol{D}_m\boldsymbol{V}_m^T$ where $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ gather the m first columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ and $\boldsymbol{D}_m$ contain the m first lines and columns of $\boldsymbol{D}$. It can therefore be inferred that:

$$\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{\frac{1}{2}} \boldsymbol{Z}^* \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{\frac{1}{2}} = \boldsymbol{U}_m\boldsymbol{D}_m\boldsymbol{V}_m^T.$$

Hence, we have:

$$\boldsymbol{Z}^* = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-\frac{1}{2}} \boldsymbol{U}_m\boldsymbol{D}_m\boldsymbol{V}_m^T \left(\boldsymbol{X}_t\boldsymbol{X}_t^T\right)^{-\frac{1}{2}}.$$

Proof of Lemma 1

First equation

From Eq. (35), it can be inferred that:

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{M}_i | \{\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{Y}_i\left(\bar{\boldsymbol{x}}_i^*\right) = \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\}, \boldsymbol{R}_{t_i}, C_i\right] = \boldsymbol{Y}_i^{\text{obs,new}} \left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1} \left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T \\
\left(\boldsymbol{H}_i^{\text{obs,new}} \left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1} \left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T\right)^{-1},
\end{aligned}$$

where

$$\begin{aligned} \boldsymbol{Y}_i^{\text{obs,new}} &= \left(\boldsymbol{Y}_i^{\text{obs}} \ , \ \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\right), \\ \boldsymbol{H}_i^{\text{obs,new}} &= \left(\boldsymbol{H}_i^{\text{obs}} \ , \ \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right), \end{aligned} \tag{77}$$

and

$$\boldsymbol{R}_i^{\text{obs,new}} = \begin{pmatrix} \boldsymbol{R}_i^{\text{obs,new}} & C\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) \\ C\left(\bar{\boldsymbol{x}}_i^*, \overline{\boldsymbol{X}}_i^{\text{obs}}\right) & 1 \end{pmatrix}.$$

Using the Schur complement formulae, one get:

$$\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}, \tag{78}$$

where

$$\begin{aligned} \boldsymbol{Q}_{11} &= \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} + \boldsymbol{Q}_{22}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) C\left(\bar{\boldsymbol{x}}_i^*, \overline{\boldsymbol{X}}_i^{\text{obs}}\right)\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} \\ &= \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} + \boldsymbol{Q}_{22}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T, \\ \boldsymbol{Q}_{12} &= -\boldsymbol{Q}_{22}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) = -\boldsymbol{Q}_{22}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right), \\ \boldsymbol{Q}_{21} &= -\boldsymbol{Q}_{22} C\left(\bar{\boldsymbol{x}}_i^*, \overline{\boldsymbol{X}}_i^{\text{obs}}\right)\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} = -\boldsymbol{Q}_{22}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T, \end{aligned} \tag{79}$$

$\boldsymbol{Q}_{22} \in \mathbb{R}$, and:

$$\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right) = \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1} C\left(\overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) \tag{80}$$

From the previous equations, one has:

$$\begin{aligned} &\boldsymbol{H}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T \\ &= \boldsymbol{H}_i^{\text{obs}}\boldsymbol{Q}_{11}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{21}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{H}_i^{\text{obs}}\boldsymbol{Q}_{12}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T + \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{22}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T \\ &= \boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\Big(\boldsymbol{H}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\right)^T - \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\right)^T \\ &\quad -\boldsymbol{H}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T + \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\Big) \\ &= \boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\left(\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right) - \boldsymbol{H}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\left(\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right) - \boldsymbol{H}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)^T \\ &= \boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T. \end{aligned} \tag{81}$$

In the same way, one has:

$$\begin{aligned} &\boldsymbol{Y}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T \\ &= \boldsymbol{Y}_i^{\text{obs}}\boldsymbol{Q}_{11}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{21}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Y}_i^{\text{obs}}\boldsymbol{Q}_{12}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T + \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{22}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T \\ &= \boldsymbol{Y}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\Big(\boldsymbol{Y}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\right)^T - \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\left(\boldsymbol{H}_i^{\text{obs}}\right)^T \\ &\quad -\boldsymbol{Y}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T + \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T\Big) \\ &= \boldsymbol{Y}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\left(\boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right) - \boldsymbol{Y}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\left(\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right) - \left(\boldsymbol{H}_i^{\text{obs}}\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)^T \\ &= \boldsymbol{Y}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\left(\boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right) - \boldsymbol{Y}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T. \end{aligned}$$

From Eq. (34), it can be inferred that:

$$\boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right) - \boldsymbol{Y}_i^{\text{obs}}\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right) = \widehat{\boldsymbol{M}}_i\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right).$$

where $\widehat{\boldsymbol{M}}_i = \mathbb{E}\left[\boldsymbol{M}_i|\boldsymbol{Y}_i^{\text{obs}}, C_i\right]$ is defined by Eq. (35).
Therefore, one has:

$$\boldsymbol{Y}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T = \boldsymbol{Y}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\widehat{\boldsymbol{M}}_i\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T.$$

From Eq. (81) it can be inferred that:

$$\begin{aligned}\boldsymbol{Y}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T &= \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\widehat{\boldsymbol{M}}_i\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T. \\ &= \widehat{\boldsymbol{M}}_i\left(\boldsymbol{H}_i^{\text{obs}}\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs}}\right)^T + \boldsymbol{Q}_{22}\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{u}_i\left(\bar{\boldsymbol{x}}_i^*\right)^T.\right) \\ &= \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T.\end{aligned}$$

Therefore, one has:

$$\boldsymbol{Y}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T\left(\boldsymbol{H}_i^{\text{obs,new}}\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{H}_i^{\text{obs,new}}\right)^T\right)^{-1} = \widehat{\boldsymbol{M}}_i.$$

Thus, one can conclude from the previous equation and Eq. (35) that:

$$\mathbb{E}\left[\boldsymbol{M}_i|\{\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{Y}_i\left(\bar{\boldsymbol{x}}_i^*\right) = \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\}, \boldsymbol{R}_{t_i}, C_i\right] = \mathbb{E}\left[\boldsymbol{M}_i|\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{R}_{t_i}, C_i\right].$$

Second equation

From Eq. (36) it can be inferred that:

$$\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\right) = \frac{1}{n_i+1}\left(\boldsymbol{Y}_i^{\text{obs,new}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs,new}}\right)\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1}\left(\boldsymbol{Y}_i^{\text{obs,new}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs,new}}\right)^T. \quad (82)$$

From Eq. (77), one has:

$$\boldsymbol{Y}_i^{\text{obs,new}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs,new}} = \left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\ ,\ \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right) - \widehat{\boldsymbol{M}}_i\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right). \quad (83)$$

From Eq. (34), one get:

$$\boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right) - \widehat{\boldsymbol{M}}_i\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^*\right) = \left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right),$$

where $\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)$ is defined by Eq. (80).
Therefore, Eq. (83) becomes:

$$\boldsymbol{Y}_i^{\text{obs,new}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs,new}} = \left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)\left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right).$$

Eq. (82) can thus be rewritten:

$$\begin{aligned}\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\right) = \frac{1}{n_i+1}&\left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)\left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1} \\ &\left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)^T\left(\boldsymbol{Y}_i^{\text{obs}} - \widehat{\boldsymbol{M}}_i\boldsymbol{H}_i^{\text{obs}}\right)^T.\end{aligned} \quad (84)$$

Besides, one has:

$$\begin{aligned}\left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\left(\boldsymbol{R}_i^{\text{obs,new}}\right)^{-1} &= \left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\begin{pmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22}\end{pmatrix} \\ &= \left(\boldsymbol{Q}_{11} + \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{21}\ ,\ \boldsymbol{Q}_{12} + \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\boldsymbol{Q}_{22}\right). \\ &= \left(\left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}\ ,\ \boldsymbol{0}_{n_i}\right)\end{aligned}$$

One can infer that:

$$\left(\boldsymbol{I}_{n_i}\ ,\ \boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\begin{pmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22}\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_{n_i}\\ \left(\boldsymbol{v}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)^T\end{pmatrix} = \left(\boldsymbol{R}_i^{\text{obs}}\right)^{-1}$$

One can conclude from Eq. (84) and the previous equation, that:

$$\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{\mu}_i^c\left(\bar{\boldsymbol{x}}_i^*\right)\right) = \frac{n_i}{n_i+1}\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right).$$

Proof of Proposition 2

From Eq. (39) it can be inferred that the criterion can be rewritten:

$$\bar{\boldsymbol{x}}_i^{new} = \underset{\bar{\boldsymbol{x}}_i^* \in \overline{\mathbb{X}}_i}{\operatorname{argmin}} \int_{\overline{\mathbb{X}}_i} \operatorname{Tr}\left(\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}, \boldsymbol{y}_i\left(\bar{\boldsymbol{x}}_i^*\right)\right)\right) v_i\left(\bar{\boldsymbol{x}}_i | \overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) d\mu_{\overline{\mathbb{X}}_i}\left(\bar{\boldsymbol{x}}_i\right). \tag{85}$$

Moreover, based on Lemma 1:

$$\bar{\boldsymbol{x}}_i^{new} = \underset{\bar{\boldsymbol{x}}_i^* \in \overline{\mathbb{X}}_i}{\operatorname{argmin}} \int_{\overline{\mathbb{X}}_i} \frac{n_i}{n_i+1} \operatorname{Tr}\left(\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right)\right) v_i\left(\bar{\boldsymbol{x}}_i | \overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) d\mu_{\overline{\mathbb{X}}_i}\left(\bar{\boldsymbol{x}}_i\right). \tag{86}$$

By noting that $\operatorname{Tr}\left(\boldsymbol{R}_{t_i}\left(\boldsymbol{Y}_i^{\text{obs}}\right)\right)$ does not depends on $\bar{\boldsymbol{x}}_i$ and $\bar{\boldsymbol{x}}_i^*$, the criterion can finally be written:

$$\bar{\boldsymbol{x}}_i^{new} = \underset{\bar{\boldsymbol{x}}_i^* \in \overline{\mathbb{X}}_i}{\operatorname{argmin}} \int_{\overline{\mathbb{X}}_i} v_i\left(\bar{\boldsymbol{x}}_i | \overline{\boldsymbol{X}}_i^{\text{obs}}, \bar{\boldsymbol{x}}_i^*\right) d\mu_{\overline{\mathbb{X}}_i}\left(\bar{\boldsymbol{x}}_i\right). \tag{87}$$

Proof of Proposition 3

By definition (see Eq. (51)) and given the independence of $\boldsymbol{Y}_2$ and $\boldsymbol{Y}_3$, one has:

$$\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) \overset{d}{=} \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) + \boldsymbol{\xi}_2 \sigma_{\boldsymbol{x}2}^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right),$$

where $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ are independent Gaussian variables, such that $\boldsymbol{\xi}_2 \sim \mathcal{N}\left(\boldsymbol{0}, \widehat{\boldsymbol{R}}_{t_2}\right)$ and $\boldsymbol{\xi}_3 \sim \mathcal{N}\left(\boldsymbol{0}, \widehat{\boldsymbol{R}}_{t_3}\right)$.

This implies:

$$\mathbb{E}\left[\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right)\right] = \mathbb{E}\left[\boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)\right],$$

because $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ are independent and $\mathbb{E}\left[\boldsymbol{\xi}_2\right] = \boldsymbol{0}$.

Besides, one has:

$$\begin{aligned}
\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) \boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right)^T = {} & \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)^T \\
& + \left(\boldsymbol{\sigma}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)\right)^2 \boldsymbol{\xi}_2 \boldsymbol{\xi}_2^T \\
& + \boldsymbol{\sigma}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \boldsymbol{\xi}_2 \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)^T \\
& + \boldsymbol{\sigma}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \boldsymbol{\xi}_2^T.
\end{aligned}$$

Given that $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ are independent, $\mathbb{E}\left[\boldsymbol{\xi}_2\right] = \boldsymbol{0}$ and $\mathbb{E}\left[\boldsymbol{\xi}_2 \boldsymbol{\xi}_2^T\right] = \widehat{\boldsymbol{R}}_{t_2}$, one has:

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) \boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right)^T\right] = {} & \mathbb{E}\left[\left(\boldsymbol{\sigma}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)\right)^2\right] \widehat{\boldsymbol{R}}_{t_2} \\
& + \mathbb{E}\left[\boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right)^T\right].
\end{aligned}$$

Proof of Proposition 4

By definition (see Eq. (51)), one has:

$$\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) \overset{d}{=} \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) + \boldsymbol{\xi}_2 \sigma_{\boldsymbol{x}2}^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right) + \boldsymbol{\xi}_3 \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right),$$

where $\boldsymbol{\xi}_2 \sim \mathcal{N}\left(\boldsymbol{0}, \widehat{\boldsymbol{R}}_{t_2}\right)$ and $\boldsymbol{\xi}_3 \sim \mathcal{N}\left(\boldsymbol{0}, \widehat{\boldsymbol{R}}_{t_3}\right)$.

If $\sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right)$ is small enough, then the previous equation can be linearized with respect to $\sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right)$. Thus, one has:

$$\boldsymbol{Y}_{\text{nest}}^c\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) \overset{d}{\approx} \boldsymbol{\mu}_2^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) + \frac{\partial \boldsymbol{\mu}_2^c}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \sigma_{\boldsymbol{x}3}^c\left(\boldsymbol{x}_1\right) \boldsymbol{\xi}_3 + \boldsymbol{\xi}_2 \sigma_{\boldsymbol{x}2}^c\left(\boldsymbol{\mu}_3^c\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right).$$

Thanks to the fact that $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ are independent, a Gaussian predictor of the nested code can be obtained from the previous equation. Furthermore, this Gaussian process has the following mean function:

$$\boldsymbol{\mu}^c_{\text{nest}}\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) = \boldsymbol{\mu}^c_2\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right), \tag{88}$$

and its covariance function is:

$$\begin{aligned}\boldsymbol{C}^c_{\text{nest}}\left(\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right), \left(\boldsymbol{x}_1^{'}, \boldsymbol{x}_2^{'}\right)\right) = \widehat{\boldsymbol{R}}_{t_2} C^c_2\left(\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right), \left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1^{'}\right), \boldsymbol{x}_2^{'}\right)\right) \\ + \frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1\right), \boldsymbol{x}_2\right) \widehat{\boldsymbol{R}}_{t_3}\left(\frac{\partial \boldsymbol{\mu}^c_2}{\partial \boldsymbol{\rho}}\left(\boldsymbol{\mu}^c_3\left(\boldsymbol{x}_1^{'}\right), \boldsymbol{x}_2^{'}\right)\right)^T C^c_3\left(\boldsymbol{x}_1, \boldsymbol{x}_1^{'}\right).\end{aligned}$$

References

Bachoc F (2013) Cross validation and maximum likelihood estimation of hyper-parameters of Gaussian processes with model misspecification. Computational Statistics and Data Analysis 66:55––69

Baker CTH (1977) The numerical treatment of integral equations. Clarendon Press, Oxford

Bates RA, Buck RJ, Riccomagno E, Wynn HP (1996) Experimental design and observation for large systems. Journal of the Royal Statistical Society Series B (Methodological) 58(1):77–94

Bect J, Ginsbourger D, Li L, Picheny V, Vasquez E (2012) Sequential design of computer experiments for the estimation of a probability of failure. Statistics and Computing 22:773–797

Constantine P, Dow E, Wang QQ (2014) Active Subspace methods in theory and practice: Applications to Kriging surfaces. SIAM Journal on Scientific Computing 36(4):1500–1524

Conti S, O'Hagan A (2010) Bayesian emulation of complex multi-output and dynamic computer models. Journal of Statistical Planning and Inference 140:640–651

Conti S, Gosling J, Oakley J, O'Hagan A (2009) Gaussian process emulation of dynamic computer codes. Biometrika 96(3):663–676

Cornford D, Nabney IT, Williams CKI (2002) Modelling frontal discontinuities in wind fields. Journal of Nonparameteric Statsitics 14:43–58

Defaux G, Evrard P (2014) Probabilistic analysis of a containment vessel subjected to dynamic pressure loading using surrogate models. In: Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures, CRC Press, pp 3203–3210, DOI 10.1201/b16387-463

Dubrule O (1983) Cross validation of Kriging in a unique neighborhood. Mathematical Geology 15(6):687–699

Eckart G, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1:211–218

Efroymson M (1960) Mathematical models for digital computers, vol 1, Wiley, chap Multiple regression analysis, pp 191—-203

Fricker T, Oakley J, Urban N (2013) Multivariate Gaussian process emulators with nonseparable covariance structures. Technometrics 55(1):47–56

Ginsbourger D, Le Riche R, Carraro L (2010) Computational intelligence in expensive optimization problems, Adaptation Learning and Optimization, vol 2, Springer Berlin Heidelberg, chap Kriging is well-suited to parallelize optimization, pp 131–162

Higdon D, Gattiker J, Williams B, Rightley M (2008) Computer model calibration using high-dimensional output. Journal of the American Statistical Association 103(482):570–583

Höskuldsson A (1988) PLS regression methods. Journal of chemometrics 2(3):211–228

Hung Y, Joseph V, Melkote S (2015) Analysis of computer experiments with functional response. Technometrics 5(1):35–44

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. Biometrika 13:455–492

Marque-Pucheu S, Perrin G, Garnier J (2018) Efficient sequential experimental design for surrogate modeling of nested codes. ESAIM Probability and Statistics

Nanty S, Helbert C, Marrel A, Pérot N, Prieur C (2017) Uncertainty quantification for functional dependent random variables. Computational Statistics 32(2):559–583

Perrin G (2018) Adaptive calibration of a computer code with time-series output. Journal of Statistical Planning and Inference

Picheny V, Ginsbourger D, Roustant O, Haftka R, Kim NH (2010) Adaptive designs of experiments for accurate approximation of a target region. Journal of Mechanical Design 132(7):071008–071008–9

Pinkus A (2015) Ridge functions. Cambridge University Press, Cambridge

Pinski F, Simpson G, Stuart A, Weber H (2015) Kullback–Leibler approximation for probability measures on infinite dimensional spaces. SIAM Journal on Mathematical Analysis 47(6):4091–4122

Rasmussen CE, Williams CK (2006) Gaussian processes for machine learning. The MIT Press, Cambridge

Rougier J (2008) Efficient emulators for multivariate deterministic functions. Journal of Computational and Graphical Statistics 17(4):827–843

Russi T (2010) Uncertainty quantification with experimental data and complex system models. PhD thesis, UC Berkeley

Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments. Statistical Science 4(4):409–423

Santner T, Williams B, Notz W (2003) The Design and Analysis of Computer Experiments. Springer-Verlag New York

Sobol I (1993) Sensitivity estimates for nonlinear mathematical models. Mathematical Modeling & Computational Experiment 1:407–414

Sobol M (2001) Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation 55:271–280

Williams B, Higdony D, Gattiker J, Moore L, McKay M, Keller-McNulty S (2006) Combining experimental data and computer simulations, with an application to flyer plate experiments. Bayesian Analysis 1(4):765–792

Wold H (1966) Estimation of Principal Components and Related Models by Iterative Least squares. Academic Press

Zahm O, Constantine P, Prieur C, Marzouk Y (2018) Gradient-based dimension reduction of multivariate vector-valued functions. HAL preprint : hal-180107922