Efficient sequential experimental design for surrogate modeling of nested codes

Sophie Marque-Pucheu, Guillaume Perrin, Josselin Garnier

▶ **To cite this version:**

Sophie Marque-Pucheu, Guillaume Perrin, Josselin Garnier. Efficient sequential experimental design for surrogate modeling of nested codes. ESAIM: Probability and Statistics, 2019, 23, pp.245-270. 10.1051/ps/2018011 . hal-02925329

HAL Id: hal-02925329
https://hal.science/hal-02925329

Submitted on 29 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ESAIM: PS 23 (2019) 245–270
https://doi.org/10.1051/ps/2018011

ESAIM: Probability and Statistics
www.esaim-ps.org

EFFICIENT SEQUENTIAL EXPERIMENTAL DESIGN FOR SURROGATE MODELING OF NESTED CODES

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

Abstract. In this paper we consider two nested computer codes, with the first code output as one of the second code inputs. A predictor of this nested code is obtained by coupling the Gaussian predictors of the two codes. This predictor is non Gaussian and computing its statistical moments can be cumbersome. Sequential designs aiming at improving the accuracy of the nested predictor are proposed. One of the criteria allows to choose which code to launch by taking into account the computational costs of the two codes. Finally, two adaptations of the non Gaussian predictor are proposed in order to compute the prediction mean and variance rapidly or exactly.

Mathematics Subject Classification. 62L05, 60G15, 62M20.

Received September 27, 2017. Accepted May 11, 2018.

1. Introduction

Thanks to computing power increase, the certification and the design of complex systems rely more and more on simulation. To this end, predictive codes are needed, which have generally to be evaluated at a large number of input points. When the computational cost of these codes is high, surrogate models are introduced to emulate their responses. A lot of industrial issues involve multi-physics phenomena, which can be associated with a series of computer codes. However, when these code networks are used for optimization, uncertainty quantification, or risk analysis purposes, they are generally considered as a single code. In that case, all the inputs characterizing the system of interest are gathered in a single input vector, and little attention is paid to the potential intermediate results. When trying to emulate such code networks, this is clearly sub-optimal, as much information is lost in the statistical learning, so that too many evaluations of each code are likely to be required to get a satisfying prediction precision.

In this paper, we focus on the case of two nested computer codes, where the output of the first code is one of the inputs of the second code. We assume that these two computer codes are deterministic, but expensive to evaluate. To predict the value of this nested code at an unobserved point, a Bayesian formalism [30] is adopted in the following. Each computer code is *a priori* modeled by a Gaussian process, and the idea is to identify the

Keywords and phrases: Nested computer codes, surrogate model, Gaussian process, uncertainty quantification, Bayesian formalism, sequential design, computer experiments.

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

* Corresponding author: sophie.marque-pucheu@cea.fr

© The authors. Published by EDP Sciences, SMAI 2019

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

posterior distribution of the combination of these two processes given a limited number of evaluations of the two codes. The Gaussian process hypothesis is widely used in computer experiments [5, 17–19, 23, 29, 31, 32], as it allows a very good trade-off between error control, complexity, and efficiency. The two main issues of this approach, also called Kriging, concern the choice of the statistical properties of the Gaussian processes that are used, and the choice of the points where to evaluate the codes. When a single computer code is considered, several methods exist to add one new point or a batch of new points sequentially to an already existing Design of Experiments. Depending on the purpose, optimization or reconstruction of the objective function on its whole input set, the criteria are based on the mean, variance or covariance of the predictor [4, 7, 9, 31, 32]. Given that our aim is to predict the output of the nested code on its whole input set, sequential designs based on a reduction of the integrated prediction variance (IMSE) are an appropriate choice. In the case of a single code, the variance expression can be explicitly derived under mild restrictive conditions on the mean and the covariance of the prior Gaussian distribution.

The adaptation of these selection criteria to the case of two nested codes is not direct. Indeed, the combination of two Gaussian processes is not Gaussian, so that the prediction variance is much more complicated to estimate. The challenges posed by the composition of two Gaussian processes have been studied in the Deep Gaussian processes literature and the proposed methods are based on the Monte-Carlo computation of the likelihood of the nested Gaussian processes [24] or on the computation of a lower bound of this likelihood [8]. The composition of Gaussian processes can also be used in the multi-fidelity framework [24]. This framework enables to use several levels of convergence of a simulator (for example in a finite element model a coarse mesh corresponds to the low fidelity simulator and the finer mesh corresponds to the high fidelity simulator) and therefore to have a trade-off between accuracy and computation time [17, 20, 21, 28, 35].

Moreover, if the two codes can be launched separately, the selection criterion has also to indicate which one of the two codes to launch. The sequential designs are based on the prediction variance, which has to be computed in a large number of points. To reduce the computational cost associated with these computations, we propose several adaptations of the Gaussian Process formalism to the nested case. These adaptations make it possible to compute the two first statistical moments of the code output predictor exactly or quickly. Then, original sequential selection criteria are introduced, which try to exploit as much as possible the nested structure of the studied codes. In particular, these criteria are able to integrate the fact that the computational costs associated with the evaluation of each code can be different.

The outline of this paper is the following. Section 2 presents the theoretical framework of the Gaussian process-based surrogate models, its generalization to the nested case, and introduces two selection criteria based on the prediction variance to reduce the prediction uncertainty sequentially. Section 3 introduces a series of simplifications to allow a quick computation of the prediction variance. In section 4, the presented methods are applied to two examples.

The technical proofs of the results presented in the following sections are given in the appendix.

2. Surrogate modeling for two nested computer codes

2.1. Notations

In this paper, the following notations will be adopted:

- $\overset{d}{=}$ denotes the equality in distribution.
- x, y correspond to scalars.
- $\boldsymbol{x}, \boldsymbol{y}$ correspond to vectors.
- $\boldsymbol{X}, \boldsymbol{Y}$ correspond to matrices.
- The entries of a vector $\boldsymbol{x}$ are denoted by $(\boldsymbol{x})_i$, whereas the entries of a matrix $\boldsymbol{X}$ are denoted by $(\boldsymbol{X})_{ij}$.
- $\boldsymbol{X}^T$ denotes the transpose of a matrix $\boldsymbol{X}$.
- $\mathcal{N}(\boldsymbol{x}, \boldsymbol{X})$ corresponds to the multidimensional Gaussian distribution, whose mean vector and covariance matrix are respectively given by $\boldsymbol{x}$ and $\boldsymbol{X}$.

- $\mathrm{GP}(m,k)$ corresponds to the distribution of a Gaussian process whose mean function is m, and whose covariance function is k.
- $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$ are the mathematical expectation and the variance respectively.
- For all real-valued functions y and z that are square integrable on $\mathbb{X}$, $(\cdot,\cdot)_{\mathbb{X}}$ and $\|\cdot\|_{\mathbb{X}}$ denote respectively the classical scalar product and norm in the space of square integrable real-valued functions on $\mathbb{X}$:

$$(y,z)_{\mathbb{X}} := \int_{\mathbb{X}} y(\boldsymbol{x})z(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \quad \|y\|_{\mathbb{X}}^2 := (y,y)_{\mathbb{X}}. \tag{2.1}$$

2.2. General framework

Let $\mathcal{S}$ be a system that is characterized by a vector of input parameters, $\boldsymbol{x}_{\mathrm{nest}} \in \mathbb{X}_{\mathrm{nest}}$. Let $y_{\mathrm{nest}} : \mathbb{X}_{\mathrm{nest}} \to \mathbb{R}$ be a deterministic mapping that is used to analyze the studied system. In this paper, we focus on the case where the function $\boldsymbol{x}_{\mathrm{nest}} \mapsto y_{\mathrm{nest}}(\boldsymbol{x}_{\mathrm{nest}})$ can be modeled by two nested codes. Two quantities of interest, y_1 and y_2, are thus introduced to characterize these two codes, which are supposed to be two real-valued continuous functions on their respective definition domains $\mathbb{X}_1$ and $\mathbb{R}\times\mathbb{X}_2$. Given these two functions, the nested code is defined as follows:

$$\begin{array}{ccccc} & & \boldsymbol{x}_2 \in \mathbb{X}_2 & & \\ & \nearrow & & \searrow & \\ \boldsymbol{x}_1 \in \mathbb{X}_1 & \to & y_1(\boldsymbol{x}_1) \in \mathbb{R} & & y_{\mathrm{nest}}(\boldsymbol{x}_{\mathrm{nest}}) := y_2(y_1(\boldsymbol{x}_1),\boldsymbol{x}_2) \in \mathbb{R}, \end{array} \tag{2.2}$$

where $\boldsymbol{x}_{\mathrm{nest}} := (\boldsymbol{x}_1,\boldsymbol{x}_2) \in \mathbb{X}_{\mathrm{nest}} = \mathbb{X}_1\times\mathbb{X}_2$. The sets $\mathbb{X}_1$ and $\mathbb{X}_2$ are moreover supposed to be two compact subsets of $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$ respectively, where d_1 and d_2 are two positive integers. In theory, the definition domains may be unbounded, but the reduction to compact sets enables the square integrability of y_{nest} on $\mathbb{X}_{\mathrm{nest}}$.

Given a limited number of evaluations of y_1 and y_2, the objective is to accurately predict y_{nest} on the whole input set.

2.3. Gaussian process-based surrogate models

2.3.1. Background

The Gaussian process regression (GPR), or Kriging, is a technique that is widely used to replace an expensive computer code by a surrogate model, that is to say a fast to evaluate mathematical function. The GPR is based on the assumption that the two code outputs, y_1 and y_2, can be seen as the sample paths of two stochastic processes, Y_1 and Y_2, which are supposed to be Gaussian for the sake of tractability:

$$Y_i \sim \mathrm{GP}(\mu_i, C_i), \quad i \in \{1,2\}, \tag{2.3}$$

where for all $1 \leqslant i \leqslant 2$, μ_i and C_i denote respectively the mean and the covariance functions of Y_i.

Let $\bar{\boldsymbol{x}}_1^{\mathrm{obs}} := \left(\boldsymbol{x}_1^{(1)},\ldots,\boldsymbol{x}_1^{(N_1)}\right)$ be N_1 elements of $\mathbb{X}_1$ and $\bar{\boldsymbol{x}}_2^{\mathrm{obs}} := \left(\left(\varphi_1^{(1)},\boldsymbol{x}_2^{(1)}\right),\ldots,(\varphi_1^{(N_2)},\boldsymbol{x}_2^{(N_2)})\right)$ be N_2 elements of $\mathbb{R}\times\mathbb{X}_2$. Denoting by

$$\boldsymbol{y}_1^{\mathrm{obs}} := (y_1(\boldsymbol{x}_1^{(1)}),\ldots,y_1(\boldsymbol{x}_1^{(N_1)})), \quad \boldsymbol{y}_2^{\mathrm{obs}} := (y_2(\varphi_1^{(1)},\boldsymbol{x}_2^{(1)}),\ldots,y_2(\varphi_1^{(N_2)},\boldsymbol{x}_2^{(N_2)})), \tag{2.4}$$

the vectors that gather the evaluations of y_1 and y_2 at these points, it can be shown that:

$$Y_i^c := Y_i \mid \boldsymbol{y}_i^{\mathrm{obs}} \ \sim \ \mathrm{GP}(\mu_i^c, C_i^c), \tag{2.5}$$

and the detailed expressions of the conditioned mean functions, μ_i^c, and the conditioned covariance functions, C_i^c are presented in equations (2.10) and (2.12) for the "Universal Kriging" framework. For further details on these expressions in the other frameworks, the interested reader may refer to [31, 32].

The relevance of the Gaussian process predictor strongly depends on the definitions of μ_i and C_i. When the only information about y_i is a finite set of evaluations, these functions are generally chosen in general parametric families. In this paper, functions C_i are chosen in the Gaussian and Matérn-5/2 classes (see [32, 33] for further details about classical parametric expressions for C_i).

The Gaussian class defines a parametric family of covariance functions that can be written in the form:

$$K_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) = \exp\left(-d\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right)^2\right), \tag{2.6}$$

where $d\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) = \left\|\mathrm{diag}\left(\boldsymbol{\ell}_i\right)^{-1}\left(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^{'}\right)\right\|$, $\mathrm{diag}\left(\boldsymbol{\ell}_i\right)$ denotes a square matrix whose diagonal is equal to the vector $\boldsymbol{\ell}_i$ of correlation lengths and $\|\cdot\|$ is the Euclidian norm.

Regarding the Matérn kernel, we consider the radial Matérn kernel, obtained by substituting the (weighted) Euclidean distance into the 1-dimensional Matérn kernel, and not the tensor product kernel obtained by multiplication of 1-dimensional kernels. So the covariance functions of the Matérn $\frac{5}{2}$ class can be written in the form:

$$K_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) = \left(1 + \sqrt{5}d\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) + \frac{5}{3}d\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right)^2\right)\exp\left(-\sqrt{5}d\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right)\right). \tag{2.7}$$

Linear representations are considered for the mean functions:

$$\mu_i = \boldsymbol{h}_i^T\boldsymbol{\beta}_i, \tag{2.8}$$

where $\boldsymbol{h}_i$ is a given M_i-dimensional vector of functions (see [27] for further details on the choice of the basis functions). In the following, the framework of the "Universal Kriging" is adopted, which consists in:

- assuming an (improper) uniform distribution for $\boldsymbol{\beta}_i$,
- conditioning all the results by an estimator of the hyper-parameters that characterize the covariance functions C_i (obtained by cross-validation, as explained below),
- integrating over $\boldsymbol{\beta}_i$ the conditioned distribution of Y_i.

In that case, the distribution of Y_i^c, which is defined by equation (2.5), is Gaussian, and its statistical moments can explicitly be derived (see [4, 6, 15, 27, 31]).

If we denote by

$$\widehat{\boldsymbol{\beta}}_i := \left[\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)^T\right]^{-1}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}\boldsymbol{y}_i^{\mathrm{obs}}, \tag{2.9}$$

the posterior mean of the parameters, the prediction mean and variance can be written:

$$\mu_i^c\left(\bar{\boldsymbol{x}}_i\right) = \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i\right)^T\widehat{\boldsymbol{\beta}}_i + C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}\left[\boldsymbol{y}_i^{\mathrm{obs}} - \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)^T\widehat{\boldsymbol{\beta}}_i\right], \tag{2.10}$$

and:

$$\left(\sigma_i^c\left(\bar{\boldsymbol{x}}_i\right)\right)^2 = C_i^c\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i\right), \tag{2.11}$$

$$\begin{aligned} C_i^c\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) = {} & C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) - C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{'}\right) \\ & + \left[\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i\right)^T - C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)^T\right] \\ & \times \left[\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)^T\right]^{-1} \\ & \times \left[\boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{'}\right) - \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\left(C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{\mathrm{obs}}\right)\right)^{-1}C_i\left(\bar{\boldsymbol{x}}_i^{\mathrm{obs}}, \bar{\boldsymbol{x}}_i^{'}\right)\right], \end{aligned} \tag{2.12}$$

where:

$$\bar{\boldsymbol{x}}_i := \begin{cases} \boldsymbol{x}_1 \text{ if } i = 1, \\ (\varphi_1, \boldsymbol{x}_2) \text{ if } i = 2. \end{cases} \tag{2.13}$$

In this paper, the hyperparameters of the covariance functions are estimated for each set of observations by maximizing the Leave-One-Out log predictive probability (see [29], chap. 5, and [1, 2]).

2.3.2. Coupling the surrogate models of the two codes

According to equation (2.2), the nested code, $\boldsymbol{x}_{\text{nest}} \mapsto y_{\text{nest}}(\boldsymbol{x}_{\text{nest}})$, can thus be seen as a particular realization of the conditioned process Y^c_{nest}, so that for all $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_1 \times \mathbb{X}_2$,

$$Y^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2) := Y^c_2(Y^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2). \tag{2.14}$$

Under this Gaussian formalism, the best prediction of y_{nest} at any unobserved point $\boldsymbol{x}_{\text{nest}} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ in $\mathbb{X}_1 \times \mathbb{X}_2$ is given by the mean value of $Y^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, whereas its variance can be used to characterize the confidence in the prediction. As explained in Introduction, there is no reason for Y^c_{nest} to be Gaussian, but according to Proposition 2.1, the first- and second-order moments can be obtained by computing two one-dimensional integrals with respect to a Gaussian measure.

Proposition 2.1. *For all* $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_1 \times \mathbb{X}_2$, *if* $\xi \sim \mathcal{N}(0,1)$, *then*:

$$\mathbb{E}\left[Y^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right] = \mathbb{E}\left[\mu^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\right], \tag{2.15}$$

$$\mathbb{E}\left[\left(Y^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right)^2\right] = \mathbb{E}\left[\begin{array}{l} \{\mu^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\}^2 \\ + \{\sigma^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\}^2 \end{array}\right]. \tag{2.16}$$

The computation of these moments can be done by quadrature rules or by Monte-Carlo methods ([3]). However, the computation time can be expensive, especially if the moments have to be computed at a large number of points.

Note that the proposed predictor for y_{nest} can be built using observations of y_1 or y_2 alone and not only observations of y_{nest}. It can take into account the partial information. If the two codes can be launched separately, this property will be particularly useful for the sequential enrichment of the initial design of experiments, since the variance of Y^c_{nest} can be reduced by evaluating y_1 or y_2 alone.

2.4. Sequential designs for the improvement of Gaussian process predictors

The relevance of the predictor Y^c_{nest} strongly depends on the space filling properties of the sets gathering the inputs of the available observations of y_1 and y_2, which are generally called Designs of Experiments (DoE). Space-filling Latin hypercube sampling (LHS) or quasi-Monte-Carlo sampling are generally chosen to define such *a priori* DoE [10, 11, 26]. The relevance of the predictor can then be improved by adding new points to an already existing DoE, as the higher the values of N_1 and N_2, the more chance there is for $\|\mathbb{E}\left[Y^c_{\text{nest}}\right] - y_{\text{nest}}\|^2_{\mathbb{X}_{\text{nest}}}$ to be small.

In the case of a single code, the existing selection criteria are based on the prediction variance [4, 13, 31, 32], the prediction mean [16] or both [9] or the covariance between the observations [31, 32] and depend on the goal of the experiments: optimization, reconstruction of the objective function on its whole input domain.

In this paper the objective is to predict the output of the nested code on its whole input domain. So, a stepwise uncertainty reduction (SUR) [7] strategy is adopted in order to define criteria to add a new point. The proposed criteria are based on a minimization of the IMSE (integral of the prediction variance over the input domain) or on a maximization of the reduction of IMSE per unit of computational time. Some criteria that

enable to take into account the different costs of several computer codes exist, for example in the multi-fidelity framework [34] or multi-objective constraints [25], but their adaptation to the case of two nested codes is not direct.

The use of IMSE is simplified by some properties of the Gaussian processes. Indeed, if Z is a Gaussian process that is indexed by $\boldsymbol{x}$ in $\mathbb{X}$, the variance of the conditioned random variable $Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{\text{new}})$, where $\boldsymbol{x}$ and $\boldsymbol{x}^{\text{new}}$ are any elements of $\mathbb{X}$, does not depend on the (unknown) value of $Z(\boldsymbol{x}^{\text{new}})$. So this variance can be denoted by abuse of notation $\mathbb{V}\left[Z(\boldsymbol{x}) \mid \boldsymbol{x}^{\text{new}}\right]$. To minimize the global uncertainty over Z at a reduced computational cost, a natural approach would consist in searching the value of $\boldsymbol{x}^{\text{new}}$ so that

$$\int_{\mathbb{X}} \mathbb{V}\left[Z(\boldsymbol{x}) \mid \boldsymbol{x}^{\text{new}}\right] \mathrm{d}\boldsymbol{x} \tag{2.17}$$

is minimal (under the condition that this integral exists).

In the nested case, we also have to choose to which code to add a new observation point. To this end, let τ_1 and τ_2 be the numerical costs (in CPU time for instance) that are associated with the evaluations of y_1 and y_2 respectively. For the sake of simplicity, we assume that these numerical costs are independent on the value of the input parameters, and that they are *a priori* known. Two selection criteria are eventually proposed to optimize the relevance of the predictor of the nested code output sequentially. To simplify the reading, the following notation is proposed:

$$(\widetilde{\boldsymbol{x}}_i, \widetilde{\mathbb{X}}_i) := \begin{cases} (\boldsymbol{x}_1^*, \mathbb{X}_1) \text{ if } i = 1, \\ \left((\varphi_1^*, \boldsymbol{x}_2^*), \mu_1^c(\mathbb{X}_1) \times \mathbb{X}_2\right) \text{ if } i = 2, \\ \left((\boldsymbol{x}_1^*, \boldsymbol{x}_2^*), \mathbb{X}_1 \times \mathbb{X}_2\right) \text{ if } i = 3, \end{cases} \tag{2.18}$$

where $\boldsymbol{x}_1^* \in \mathbb{X}_1$, $\varphi_1^* \in \mu_1^c(\mathbb{X}_1)$ and $\boldsymbol{x}_2^* \in \mathbb{X}_2$ and we denote by $\mathbb{V}(Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i)$ the variance of $Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})$ under the hypothesis that the code(s) corresponding to the new point $\widetilde{\boldsymbol{x}}_i$ is (are) evaluated at this point (in practice, we remind that these code evaluations are not required for the estimation of this variance). This variance can be defined as:

$$\mathbb{V}(Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i) := \begin{cases} \mathbb{V}(Y_2(Y_1(\boldsymbol{x}_1), \boldsymbol{x}_2) | \boldsymbol{y}_1^{\text{obs}}, \boldsymbol{y}_2^{\text{obs}}, y_i(\widetilde{\boldsymbol{x}}_i)), \ i \in \{1,2\}, \\ \mathbb{V}(Y_2(Y_1(\boldsymbol{x}_1), \boldsymbol{x}_2) | \boldsymbol{y}_1^{\text{obs}}, \boldsymbol{y}_2^{\text{obs}}, y_{\text{nest}}(\widetilde{\boldsymbol{x}}_i)), \ i = 3, \end{cases} \tag{2.19}$$

with $\boldsymbol{x}_{\text{nest}} := (\boldsymbol{x}_1, \boldsymbol{x}_2)$.

- First, the chained I-optimal criterion selects the best point in $\mathbb{X}_1 \times \mathbb{X}_2$ to minimize the integrated variance of the predictor of the nested code:

$$\widetilde{\boldsymbol{x}}_3^{\text{new}} = \underset{\widetilde{\boldsymbol{x}}_3 \in \widetilde{\mathbb{X}}_3}{\operatorname{argmin}} \int_{\mathbb{X}_{\text{nest}}} \mathbb{V}(Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_3) \mathrm{d}\boldsymbol{x}_{\text{nest}}. \tag{2.20}$$

  Such a criterion is *a priori* adapted to the case where it is not possible to run independently the codes 1 and 2.
- Secondly, the best I-optimal criterion selects the best among the candidates in $\mathbb{X}_1$ and $\mathbb{X}_2$ in order to maximize the decrease per unit of computational cost of the integrated prediction variance of the nested code:

$$(i^{\text{new}}, \widetilde{\boldsymbol{x}}_{i^{\text{new}}}^{\text{new}}) = \underset{\widetilde{\boldsymbol{x}}_i \in \widetilde{\mathbb{X}}_i, \ i \in \{1,2\}}{\operatorname{argmax}} \frac{1}{\tau_i} \times \int_{\mathbb{X}_{\text{nest}}} \left[\mathbb{V}\left(Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})\right) - \mathbb{V}\left(Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i\right)\right] \mathrm{d}\boldsymbol{x}_{\text{nest}}. \tag{2.21}$$

In that case, the difference in the computational costs is taken into account, and a linear expected improvement per unit of computational cost is assumed for the sake of simplicity.

For each new observation of the first code, the hyperparameters of the covariance function C_1 are re-estimated. In the same way, for each new observation of the second code, the hyperparameters of the covariance function C_2 are re-estimated.

An initial set of observations is necessary to estimate the hyperparameters of the covariance functions C_1 and C_2 and therefore to compute the prediction variance and the proposed sequential design criteria. This initial set will be chosen as a maximin LHS design on $\mathbb{X}_{\text{nest}}$.

3. Fast computation of the prediction variance

As explained in Section 2.4, choosing the position of the new point requires to compute the value of $\text{Var}(Y^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i)$ for each potential value of $\widetilde{\boldsymbol{x}}_i$ in $\widetilde{\mathbb{X}}_i$ and for a grid or a sample of $\boldsymbol{x}_{\text{nest}}$ used in a quadrature formula or an empirical average to approximate the integral in $\boldsymbol{x}_{\text{nest}}$ of equations (2.21) and (2.20).

For a given $\boldsymbol{x}_{\text{nest}}$, the variance is theoretically given by equations (2.15) and (2.16). If a quadrature rule or a Monte Carlo approach is used to approximate the variance, then the optimization procedure becomes prohibitively expensive from the computational point of view. To circumvent this problem, we present in this section several approaches to make the computation of $\text{Var}(Y^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i)$ explicit, and therefore extremely fast to compute.

3.1. Explicit derivation of the two first statistical moments of the nested code predictor

Lemma 3.1. *If $X \sim \mathcal{N}(\mu, \sigma^2)$ and $g(x,a,b,c) := x^a \exp\left(bx + cx^2\right)$, $(a,b,c) \in \mathbb{N}\times\mathbb{R}^2$, then, under the condition that $1 - 2c\sigma^2 > 0$, the mean of $g(X,a,b,c)$ can be computed analytically, and its expression is given by equation (B.1).*

Lemma 3.2. *If $g(x,a,b,c) := x^a \exp\left(bx + cx^2\right)$, $(a,b,c) \in \mathbb{N}\times\mathbb{R}^2$, then $g(x,a_i,b_i,c_i)\, g(x,a_j,b_j,c_j) = g(x, a_i+a_j, b_i+b_j, c_i+c_j)$, $(a_i,b_i,c_i) \in \mathbb{N}\times\mathbb{R}^2$ and $(a_j,b_j,c_j) \in \mathbb{N}\times\mathbb{R}^2$.*

Proposition 3.3. *Using the notations of the Universal Kriging framework that is introduced in Section 2.3, if:*

1. *for $1 \leqslant k \leqslant M_2$ the mean function $(\boldsymbol{h}_2)_k$ is of the form:*

$$(\boldsymbol{h}_2(\varphi_1, \boldsymbol{x}_2)_k = m_k(\boldsymbol{x}_2)\ \varphi_1^{a_k}, \tag{3.1}$$

   where m_k is a deterministic function from $\mathbb{X}_2$ to $\mathbb{R}$ and $a_k \in \mathbb{N}$,
2. *the covariance function C_2 is squared exponential, i.e. an element of the Gaussian class,*

then the conditional moments of order 1 and 2 of $Y^c_{nest}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, which are defined by equations (2.15) and (2.16) can be calculated analytically using Lemmas 3.1 and 3.2. Moreover, the expression of the first order moment is given by equations (D.5) and (B.1) and the expression of the second order moment is given by equations (D.8) and (B.1).

In other words, if the prior of the Gaussian process modeling the function y_2 has a trend which is a polynomial of φ_1, with coefficients as functions of $\boldsymbol{x}_2$, and a covariance function of the Gaussian class, then the moments of order 1 and 2 of the coupling of the predictors of the two codes can be computed explicitly.

In particular, if the process associated with y_2 has a constant or zero mean and a squared exponential (*i.e.* Gaussian) covariance, then the mean and the variance of the coupling of the predictors of y_1 and y_2 can be computed analytically.

However the use of a Gaussian covariance function is based on the assumption of infinite differentiability of the second code. This assumption is not necessarily verified.

Besides, the method cannot be applied to the case of more than two codes. Indeed, in the case of three codes, the coupling of the Gaussian predictors of the two first codes is no longer Gaussian. Even if the Gaussian process modeling the third code has a Gaussian covariance and a polynomial trend with respect to the output of the

second code, the analytical method cannot be applied because the predictor of the output of the chain of the two first codes is not Gaussian.

3.2. Linearized approach

In the cases where the conditions for Proposition 3.3 are not fulfilled (or if more than two codes are considered), another approach is proposed in this section, which is based on a linearization of the process modeling the nested code. Indeed, for $i \in \{1,2\}$, let ε_i^c be the Gaussian process so that:

$$Y_i^c = \mu_i^c + \varepsilon_i^c. \tag{3.2}$$

By construction, ε_i^c is the residual prediction uncertainty once Y_i has been conditioned by N_i evaluations of y_i. We remind that the two Gaussian processes Y_i are statistically independent, so Y_i^c and therefore ε_i^c are statistically independent. Under the condition that N_1 is large enough for Y_1^c being a reliable statistical model for y_1, then ε_1^c is small.

Proposition 3.4. *If:*

1. *the predictor of a nested computer code can be written* $Y_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2) := Y_2^c(Y_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)$, *where* Y_i^c *are independent Gaussian processes which can be written as* $Y_i^c = \mu_i^c + \varepsilon_i^c$, *where* $\varepsilon_i^c \sim GP(0, C_i^c)$, $i \in \{1,2\}$,
2. *and* ε_1^c *is small enough for the linearization to be valid,*

then the predictor of the nested computer code can be defined as a Gaussian process with the following mean and covariance functions:

$$c\mu_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mu_2^c(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2), \tag{3.3}$$

$$\begin{aligned} C_{nest}^c((\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}_1', \boldsymbol{x}_2')) = {} & C_2^c((\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2), (\mu_1^c(\boldsymbol{x}_1'), \boldsymbol{x}_2')) \\ & + \frac{\partial \mu_2^c}{\partial \varphi_1}(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2) \frac{\partial \mu_2^c}{\partial \varphi_1}(\mu_1^c(\boldsymbol{x}_1'), \boldsymbol{x}_2') C_1^c(\boldsymbol{x}_1, \boldsymbol{x}_1'), \end{aligned} \tag{3.4}$$

where μ_i^c, $i \in 1,2$ *is given by equation* (2.10) *and* C_i^c, $i \in 1,2$ *is given by equation* (2.12) *and* $\dfrac{\partial \mu_2^c}{\partial \varphi_1}(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)$ *is given by equation* (E.5).

It can also be written $Y_{nest}^c = \mu_{nest}^c + \varepsilon_{nest}^c$ *with:*

$$\varepsilon_{nest}^c(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{\partial \mu_2^c}{\partial \varphi_1}(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)\varepsilon_1^c(\boldsymbol{x}_1) + \varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2). \tag{3.5}$$

Corollary 3.5. *In the framework of Universal Kriging for* Y_1^c *and* Y_2^c *with explicit basis functions* $\boldsymbol{h}_i$ *and covariance functions* C_i, $i \in \{1,2\}$, *if the derivatives* $\dfrac{\partial \boldsymbol{h}_2}{\partial \varphi_1}(\varphi_1, \boldsymbol{x}_2)$ *and* $\dfrac{\partial C_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}_2^{obs}\right)$ *can be computed explicitly, then the predictor of the nested computer code can be defined, thanks to a linearization, as a Gaussian process with explicit mean and covariance functions. In particular, if the covariance function* C_2 *is in the Matérn* $\dfrac{5}{2}$ *or Gaussian class, the derivative* $\dfrac{\partial C_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}_2^{obs}\right)$ *can be computed analytically, and the associated expressions are given in equations* (F.5) *and* (F.8).

Corollary 3.6. *According to equations* (3.5), (2.21) *and* (2.20), *if the predictor of the nested code is obtained with the linearized method, then, thanks to the independence between* ε_1^c *and* ε_2^c, *the selection criteria of the sequential designs can be written:*

- *for the chained I-optimal design:*

$$(\boldsymbol{x}_1^{new}, \boldsymbol{x}_2^{new}) = \underset{(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) \in \mathbb{X}_1 \times \mathbb{X}_2}{\operatorname{argmin}} \int_{\mathbb{X}_{nest}} \left(\left(\frac{\partial \mu_2^c}{\partial \varphi_1} (\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2) \right)^2 \mathbb{V}\left[\varepsilon_1^c(\boldsymbol{x}_1) | \boldsymbol{x}_1^*\right] + \mathbb{V}\left[\varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2) | \mu_1^c(\boldsymbol{x}_1^*), \boldsymbol{x}_2^*\right] \right) \times \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2, \tag{3.6}$$

where $\frac{\partial \mu_2^c}{\partial \varphi_1}(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)$ *is given by equation* (E.5),
for the best I-optimal design:

$$(i^{new}, \boldsymbol{x}_i^{new}) = \underset{\tilde{\boldsymbol{x}}_i \in \tilde{\mathbb{X}}_i,\ i \in \{1,2\}}{\operatorname{argmax}} \frac{1}{\tau_i} \mathcal{V}_i\left(\tilde{\boldsymbol{x}}_i\right), \tag{3.7}$$

where:

$$\mathcal{V}_1\left(\tilde{\boldsymbol{x}}_1\right) = \int_{\mathbb{X}_{nest}} \left(\frac{\partial \mu_2^c}{\partial \varphi_1} (\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2) \right)^2 \left(\mathbb{V}\left[\varepsilon_1^c(\boldsymbol{x}_1)\right] - \mathbb{V}\left[\varepsilon_1^c(\boldsymbol{x}_1) | \tilde{\boldsymbol{x}}_1\right] \right) \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2, \tag{3.8}$$

$$\mathcal{V}_2\left(\tilde{\boldsymbol{x}}_2\right) = \int_{\mathbb{X}_{nest}} \left(\mathbb{V}\left[\varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)\right] - \mathbb{V}\left[\varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2) | \tilde{\boldsymbol{x}}_2\right] \right) \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2. \tag{3.9}$$

Hence, thanks to the proposed linearization, and the fact that the conditional distribution of a Gaussian process is still Gaussian with updated first and second order moments, the variance of $Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})$ and the one of $Y_{\text{nest}}^c(\boldsymbol{x}_{\text{nest}})|\tilde{\boldsymbol{x}}_i$ can be explicitly computed for all $(\boldsymbol{x}_{\text{nest}}, \tilde{\boldsymbol{x}}_i)$ in $\mathbb{X}_{\text{nest}} \times \tilde{\mathbb{X}}_i$. Under the condition that the linearization is valid, this approach can be applied to configurations with more than two nested codes.

However it can be inferred from equation (3.4) that the variance depends on $\boldsymbol{y}_1^{\text{obs}}$ through μ_1^c and $\boldsymbol{y}_2^{\text{obs}}$ through μ_2^c. To circumvent this problem for the computation of the forward variance in the sequential designs, we assume that for a candidate $\tilde{\boldsymbol{x}}_1$, μ_1^c corresponds to $\mathbb{E}\left[Y_1 | \boldsymbol{y}_1^{\text{obs}}\right]$ and by abuse of notation, that $(\sigma_1^c)^2 = C_1^c$ corresponds to $\mathbb{V}\left[Y_1 | \bar{\boldsymbol{x}}_1^{\text{obs}}, \tilde{\boldsymbol{x}}_1\right]$. In the same way, for a candidate $\tilde{\boldsymbol{x}}_2$, we assume that μ_2^c corresponds to $\mathbb{E}\left[Y_2 | \boldsymbol{y}_2^{\text{obs}}\right]$ and by abuse of notation, that $(\sigma_2^c)^2 = C_2^c$ corresponds to $\mathbb{V}\left[Y_2 | \bar{\boldsymbol{x}}_2^{\text{obs}}, \tilde{\boldsymbol{x}}_2\right]$. So, by doing this, we suppose that the estimate of $y_i\left(\tilde{\boldsymbol{x}}_i\right)$ can be replaced by its prediction mean $\mathbb{E}\left[Y_i\left(\tilde{\boldsymbol{x}}_i\right) | \boldsymbol{y}_i^{\text{obs}}\right]$, in accordance with the Kriging Believer strategy proposed in [12].

4. Applications

In this section, the proposed methods are applied to two examples: an analytical one-dimensional one and a multidimensional one.

In particular, the linearized method of Proposition 3.4 is compared with the analytical method of Proposition 3.3 in terms of prediction accuracy. The interest of the linearized method in terms of computation time is shown.

The linearized method is compared with the so-called blind box method. The blind box method corresponds to the case where the nested computer code is considered as a single computer code. In that case, only the inputs $\boldsymbol{x}_{\text{nest}}$ and the output y_{nest} are taken into account and a Gaussian process regression of this single computer code is done. The intermediary information φ_1 is not considered. The Gaussian process Y_{bb} can therefore be defined as follows (see also [27]):

$$Y_{bb} \sim GP\left(\boldsymbol{h}_2\left(\boldsymbol{h}_1\left(\boldsymbol{x}_1\right)^T \boldsymbol{\beta}_1^*, \boldsymbol{x}_2\right)^T \boldsymbol{\beta}_2^* + \frac{\partial \boldsymbol{h}_2}{\partial \varphi_1} \left(\boldsymbol{h}_1\left(\boldsymbol{x}_1\right)^T \boldsymbol{\beta}_1^*, \boldsymbol{x}_2\right)^T \boldsymbol{\beta}_2^* \boldsymbol{h}_1\left(\boldsymbol{x}_1\right)^T \left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_1^*\right),\ C_{bb} \right), \tag{4.1}$$

where $\left(\boldsymbol{\beta}_1^*, \boldsymbol{\beta}_2^*\right) = \underset{(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)}{\operatorname{argmin}} \sum_{i=1}^{N} \left[y_2\left(y_1\left(\boldsymbol{x}_1^{(i)}\right), \boldsymbol{x}_2^{(i)}\right) - \boldsymbol{h}_2\left(\boldsymbol{h}_1\left(\boldsymbol{x}_1^{(i)}\right)^T \boldsymbol{\beta}_1, \boldsymbol{x}_2^{(i)}\right)^T \boldsymbol{\beta}_2 \right]^2$, $N = N_1 = N_2$ and C_{bb} is a stationary covariance function chosen in a parametric family. In order to make the comparison between the blind box and the other methods easier, the mean function is defined as a linearization of the coupling of the mean functions used in the linearized method.

Finally, the performances of the sequential designs are compared with a space filling design (maximin LHS) on $\mathbb{X}_{\text{nest}}$.

4.1. Characteristics of the examples

4.1.1. Analytical example

In the analytical example, the properties of the mean functions of the Gaussian processes and of the codes are:

$$\boldsymbol{h}_1(x_1) = \begin{bmatrix} 1 \\ x_1 \\ x_1^2 \end{bmatrix}, \quad \boldsymbol{\beta}_1 = \begin{bmatrix} -2 \\ 0.25 \\ 0.0625 \end{bmatrix}, \quad y_1(x_1) = \boldsymbol{h}_1(x_1)^T \boldsymbol{\beta}_1 - 0.25\cos(2\pi x_1), \tag{4.2}$$

$$\boldsymbol{h}_2(\varphi_1) = \begin{bmatrix} 1 \\ \varphi_1 \\ \varphi_1^2 \\ \varphi_1^3 \end{bmatrix}, \qquad \boldsymbol{\beta}_2 = \begin{bmatrix} 6 \\ -5 \\ -2 \\ 1 \end{bmatrix}, \qquad y_2(\varphi_1) = \boldsymbol{h}_2(\varphi_1)^T \boldsymbol{\beta}_2 - 0.25\cos(2\pi\varphi_1), \tag{4.3}$$

where $x_1 \in [-7, 7]$. In this example $\mathbb{X}_2 = \theta$.

In the analytical example, the covariance functions are squared exponential (*i.e.* Gaussian). This implies that the Gaussian processes associated with the codes are mean square infinitely differentiable. This enables to apply Propositions 3.3 and 3.4 to this example.

4.1.2. Hydrodynamic example

In this example, the coupling of two computer codes is considered. The objective is to determine the impact point of a conical projectile.

The first code computes the drag coefficient of a cone divided by the height of the cone. Its inputs are the height and the half-angle of the cone, so the dimension of $\boldsymbol{x}_1$ is 2 and $\boldsymbol{x}_1 \in \left[\frac{\pi}{36}, \frac{\pi}{4}\right] \times [0.2, 2]$.

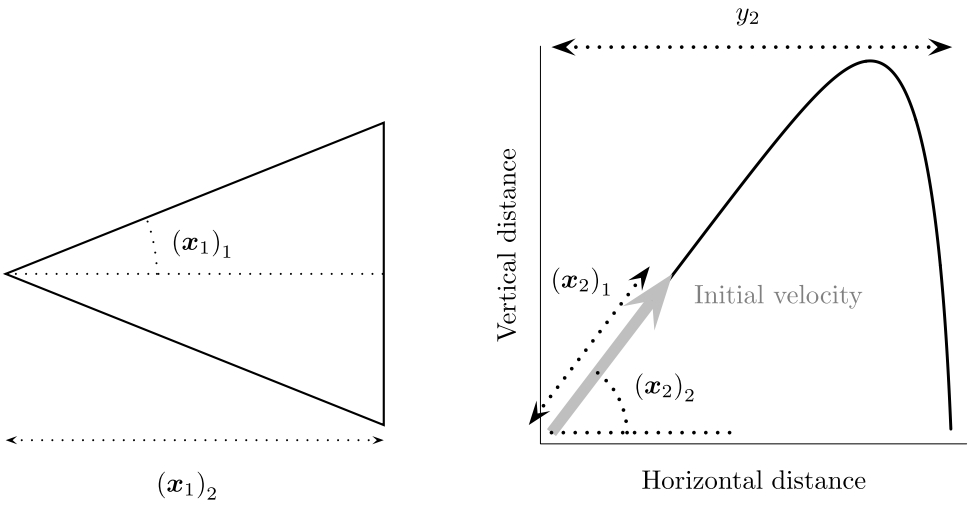
The second code computes the range of the ballistic trajectory of a cone. Its inputs are the output of the first code, associated with φ_1, and the initial velocity and angle of the ballistic trajectory of the cone, gathered in $\boldsymbol{x}_2$. The dimension of $\boldsymbol{x}_2$ is therefore 2 and $\boldsymbol{x}_2 \in [1500, 3000] \times \left[\frac{\pi}{12}, \frac{7\pi}{36}\right]$.

Figure 1 illustrates the two codes inputs and outputs.

Figure 2 presents, for each code, the scatter plots of the variations of the output with respect to the most sensitive components of their inputs. The inputs correspond to a set of 20 points drawn according to a maximin LHS design on $\mathbb{X}_{\text{nest}}$. These figures enable to propose a basis of functions for the prior mean of the processes associated with the two codes.

For the first code, the scatter plots highlight a linear variation with respect to $(\boldsymbol{x}_1)_1$ and a multiplicative inverse variation with respect to $(\boldsymbol{x}_1)_2$, so the proposed basis functions are:

$$\boldsymbol{h}_1(\boldsymbol{x}_1) = \left(1\ ,\ (\boldsymbol{x}_1)_1\ ,\ \frac{1}{(\boldsymbol{x}_1)_2}\right)^T. \tag{4.4}$$



(A) Code 1: drag coefficient / height of the cone

(B) Code 2: range of a ballistic trajectory

FIGURE 1. Hydrodynamic example: inputs and outputs of the two codes.

For the second code, only a multiplicative inverse variation with respect to φ_1 is evident, so the proposed basis functions are:

$$\boldsymbol{h}_2\left(\varphi_1, \boldsymbol{x}_2\right) = \left(\frac{1}{\max\left(\varphi_1, \varphi_{1_{\min}}\right)},\ 1\ ,\ 1\right)^T. \tag{4.5}$$

The denominator has a lower bound $\varphi_{1_{\min}}$ in order to avoid any inversion problem around zero. $\varphi_{1_{\min}}$ is set to the small arbitrary value 0.1.

The image plot Figure 2c represents the UK prediction mean of the first code, obtained with the proposed basis functions. The predicted value of y_1 for the maximum value of $(\boldsymbol{x}_1)_1$ and the minimum value of $(\boldsymbol{x}_1)_2$ is high compared with the values of the observations. So the first code has been evaluated at this input point and gives the value of 3.4, which is consistent with the prediction. This illustrates the relevance of the proposed basis, that is used to extrapolate the prediction at a point with no observations around. The image plot Figure 2e represents the UK prediction mean of the second code, obtained with the proposed basis at a value of 0.5 for φ_1.

In the hydrodynamic example, the covariance functions are Matérn $\frac{5}{2}$. This enables to perform the linearization of Proposition 3.4 and Corollary 3.5.

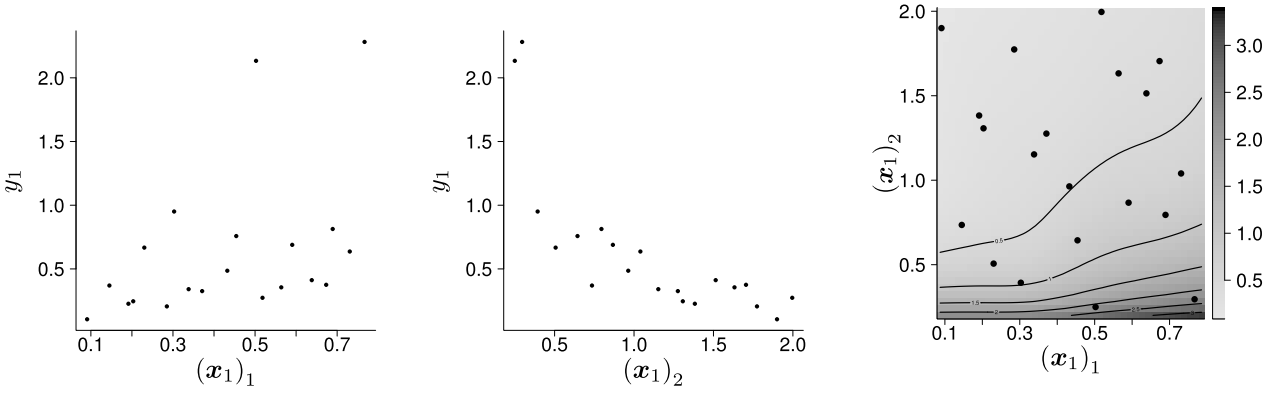
In both examples, the covariance functions include a non-zero nugget term (see [13] for further details), that means that they can be written as:

$$C_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) = \sigma_i^2\left[K_i\left(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{'}\right) + g\delta_{\bar{\boldsymbol{x}}_i = \bar{\boldsymbol{x}}_i^{'}}\right], \tag{4.6}$$

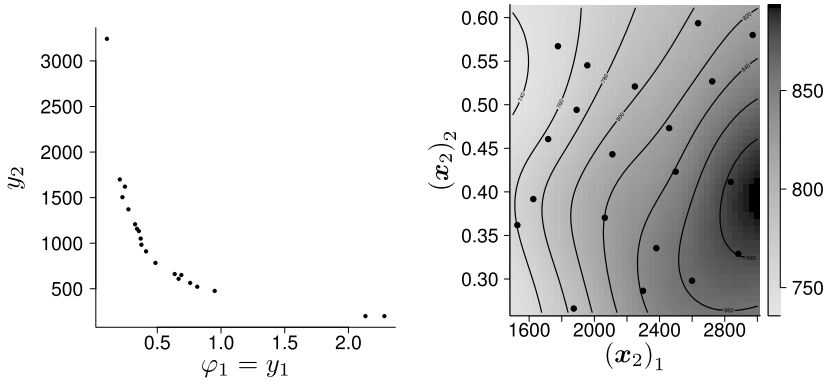
where $\sigma_i \in \mathbb{R}_+$, K_i is chosen in a parametric family (Gaussian or Matérn $\frac{5}{2}$), g is the nugget term whose value is 10^{-6}, and δ is the Kronecker delta function. This non-zero nugget term is used for reasons of numerical stability.

4.2. Prediction performance for a given set of observations

A set of validation observations is available. Let $\boldsymbol{x}_{\text{nest}}^{(1)} \ldots \boldsymbol{x}_{\text{nest}}^{(N_{\text{nest}})}$ be N_{nest} elements of $\mathbb{X}_{\text{nest}}$.



(A) Code 1 (B) Code 1 (C) UK prediction of code 1



(D) Code 2 (E) UK prediction of code 2 ($\varphi_1 = 0.5$)

FIGURE 2. Hydrodynamic example: variation of the outputs y_1 and y_2 of the two codes with respect to the most sensitive components of their inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for a set of 20 input points drawn according to a maximin LHS design on $\mathbb{X}_{\text{nest}}$. The image plots present the UK prediction (conditional mean of the GP) of y_1 and y_2 for the same set of observations.

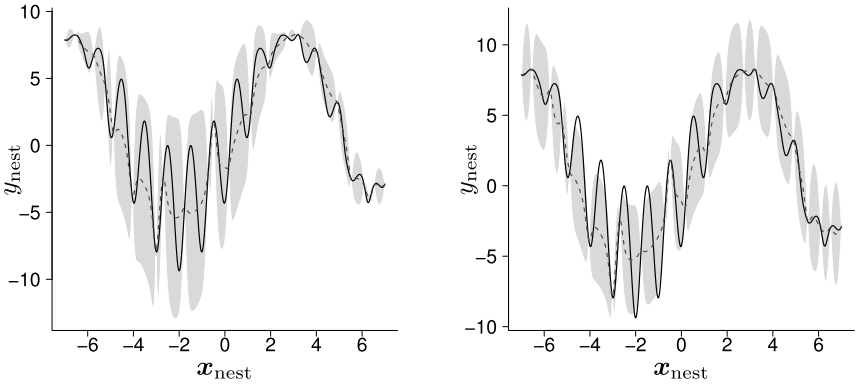
Denoting by $y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(1)}\right) \dots y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(N_{\text{nest}})}\right)$ the evaluations of the nested code at these points, the performance criterion of the nested predictor mean, also called error on the mean can be defined as:

$$\text{Error on the mean} = \frac{\sum_{i=1}^{N_{\text{nest}}} \left(y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right) - \widehat{y}_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right)\right)^2}{\sum_{i=1}^{N_{\text{nest}}} \left(y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right) - \frac{1}{N_{\text{nest}}} \sum_{j=1}^{N_{\text{nest}}} y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(j)}\right)\right)^2}, \tag{4.7}$$

where $\widehat{y}_{\text{nest}}$ denotes a prediction of the nested code, which can be obtained with the analytical, linearized or blind-box method.

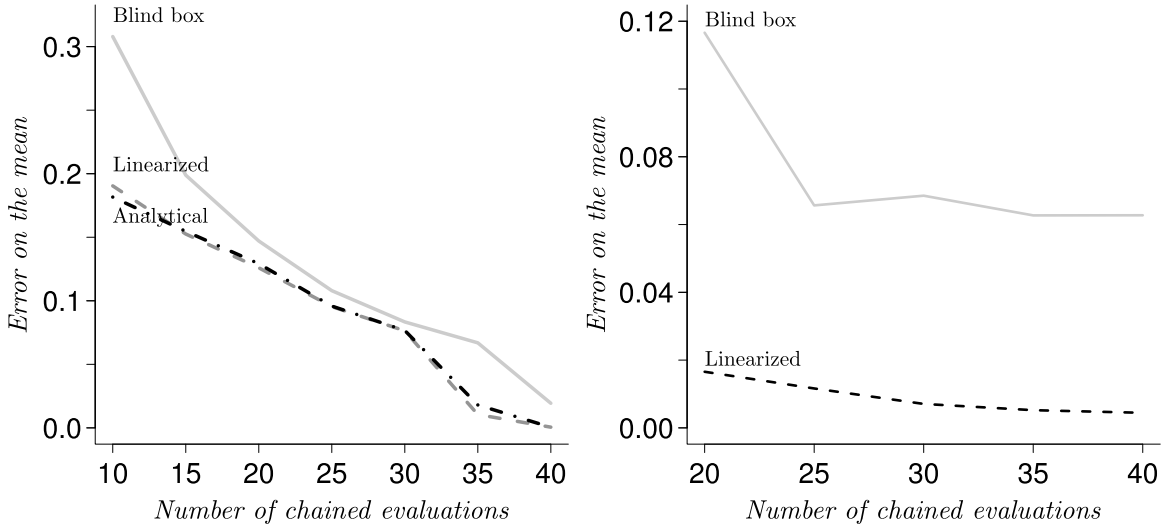
For both examples, the validation set of 150 points is drawn according to a maximin LHS on $\mathbb{X}_{\text{nest}}$.

Figure 3 presents, for the analytical example, an example of the prediction mean and 95% prediction interval computed with the linearized and the blind box methods. The two predictors are built with the same set of 20



(A) Linearized method (B) Blind box method

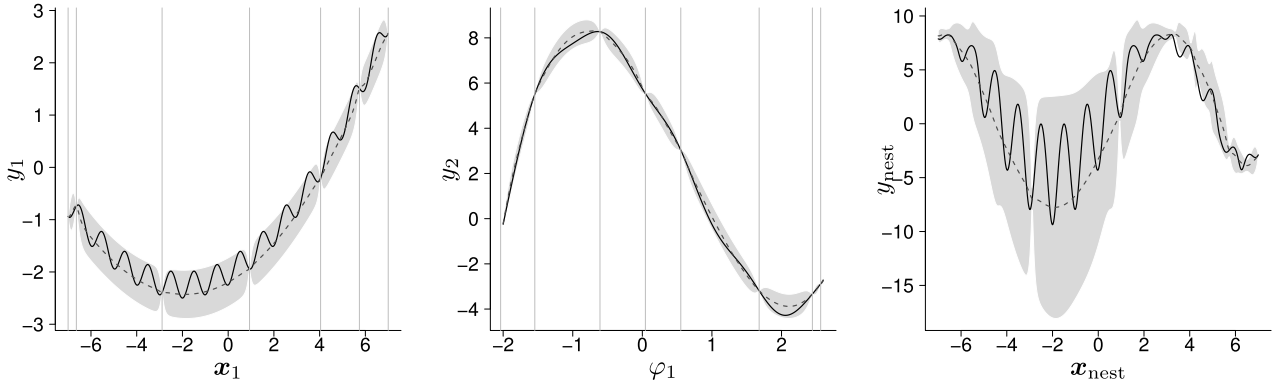
FIGURE 3. Analytical example: Predictors of the nested code obtained with the linearized and the blind box methods. The set of 20 observations is drawn according to a maximin LHS on $\mathbb{X}_{\text{nest}}$. Actual values shown by a continuous line, the prediction mean by a dotted line and the 95% prediction interval by a grey area.



(A) Analytical: Gaussian covariance (B) Hydrodynamic example: Matérn $\frac{5}{2}$ covariance

FIGURE 4. Comparison of the prediction mean accuracy for the blind box and the linearized (Prop. 3.4) methods, and, in the case of a Gaussian covariance function, the analytical method (Prop. 3.3). The curves correspond to the median of 50 draws of maximin LHS designs on $\mathbb{X}_1 \times \mathbb{X}_2$ of increasing size.

observation points drawn according to a maximin LHS design on $\mathbb{X}_{\text{nest}}$. It can be seen that, in the blind box method, the magnitude of the prediction interval is the same across the input domain and depends only on the distance to the observation points. The prediction interval is too big in the area with small variations and too small in the area with larger variations. On the contrary, the fact of taking into account the intermediary observations (linearized method here) enables to better take into account the non-stationarity of the variations of the nested code output.



(A) Code 1 (B) Code 2 (C) Nested code

FIGURE 5. Analytical example: an example of the predictors Y_1^c, Y_2^c and Y_{nest}^c. The black line represents the real values of y_1, y_2 and y_{nest}, the grey area, the 95% prediction interval and the grey dotted line, the prediction mean. The mean and prediction interval of Y_{nest}^c are computed thanks to the linearized method. The vertical lines of the two left plots represent the observations of the two codes, which are drawn according to LHS designs on $\mathbb{X}_1$ and $\mu_1^c(\mathbb{X}_1)$ of sizes 7 and 8. The number of observations is not the same for each code.

Figure 4 presents the error on the mean with the blind box and the linearized methods for both examples, and the analytical method for the analytical example. For all methods, the predictors are built with the same learning sets drawn according to maximin LHS designs on $\mathbb{X}_{\text{nest}}$ of increasing size.

The left figure, corresponding to the analytical example, shows the similar accuracies of the prediction mean computed with the analytical and linearized methods proposed in Propositions 3.3 and 3.4.

For both examples, the precision of the prediction mean is better with the linearized method than with the blind box method, showing the interest of taking into account the intermediary information.

The results show that the analytical and linearized methods lead to the same prediction mean accuracy. As a reminder, the analytical method require the infinite differentiability of the second code. This assumption is correct for the analytical example but not necessarily for the hydrodynamic example. The linearized method require the prediction error of the first code to be small enough for the linearization to be valid. Since the prediction error of the first code can be reduced thanks to a sequential enrichment of the initial design, the required assumption of the analytical method is stronger than the one of the linearized method.

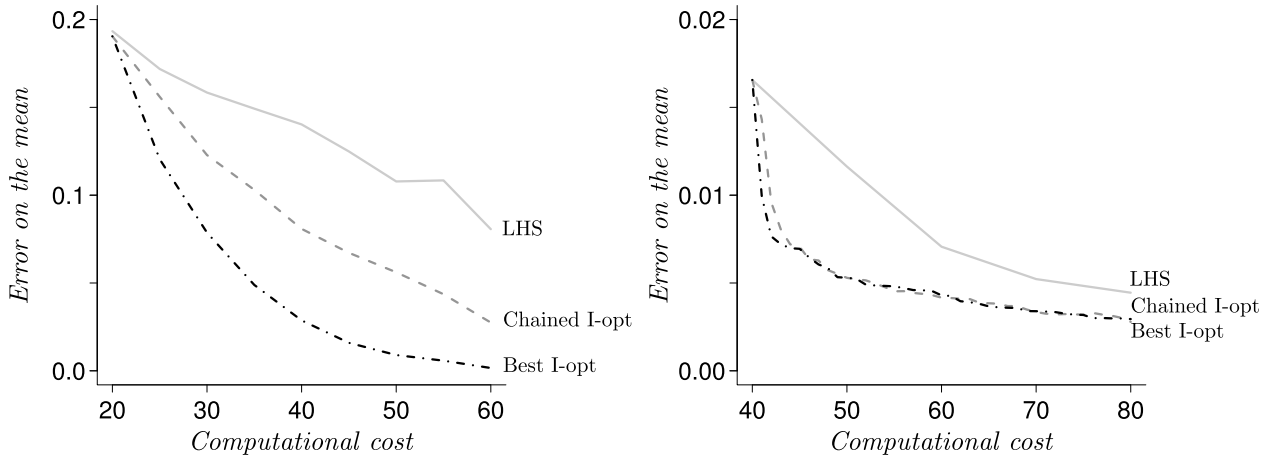
Consequently, the linearized method will be used in the remainder of the numerical applications.

4.3. Performances of the sequential designs

Figure 5 shows an example of the prediction mean and 95% prediction interval of the predictors Y_1^c, Y_2^c and Y_{nest}^c. The predictors Y_1^c and Y_2^c are not built with the same number of observations, so the predictor Y_{nest}^c is built with a different number of observations of the codes 1 and 2. The fact that the number of observations of the two codes can be different will be useful for the sequential designs. Moreover the estimation of the prediction variance of the nested code is accurate, and that will also be useful for the choice of the new observation point in the sequential designs.

4.3.1. With identical computational costs for both codes

Figure 6 presents the error on the mean of the linearized predictor for the proposed sequential designs and maximin LHS designs of increasing size. The initial designs of the sequential strategies are the same maximin LHS designs on $\mathbb{X}_{\text{nest}}$ with 10 points for the analytical example and 20 points for the hydrodynamic example. That is why the initial point of the three curves is the same on both line plots. The cost of the two codes are



(A) Analytical example (B) Hydrodynamic example

FIGURE 6. Comparison of the prediction mean accuracy of the linearized predictor with the maximin LHS design on $\mathbb{X}_{\text{nest}}$ and the sequential designs applied to the two examples. In the hydrodynamic example, the two curves representing the sequential designs are almost superimposed. The initial designs are the same for the three curves, with a size of 10 points for the analytical example and 20 points for the hydrodynamical example. The draw of the maximin LHS design on $\mathbb{X}_{\text{nest}}$ is repeated 50 times and the curves present the median of the associated results. The costs of the two codes are assumed to be the same.

considered to be the same, that means $\tau_1 = \tau_2 = 1$. The figure shows the relevance of the proposed sequential designs for improving the prediction mean of the linearized nested predictor, compared with the maximin LHS designs on $\mathbb{X}_{\text{nest}}$.

In the analytical example, the best I-optimal sequential design enables to obtain the most accurate prediction mean at a given computational cost. In the hydrodynamic example, in the first 10 iterations, the best I-optimal design outperforms the chained I-optimal design. After this initial stage, the best I-optimal design calls alternately code 1 and code 2 and becomes equivalent to the chained I-optimal design.

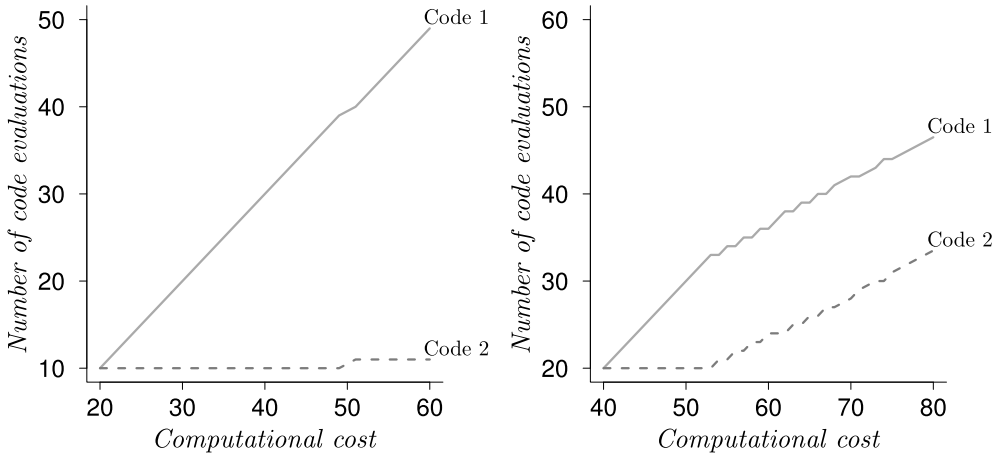
Figure 7 shows to which of the two codes the new observations points are added for the best I-optimal sequential design. In both examples new observation points of the first code are first added.

It seems that the uncertainty propagated from the first code into the second code is predominant at the beginning. The best I-optimal sequential design aims therefore at reducing this uncertainty by first adding new observation points of the first code. Then new observations of both codes are added.

4.3.2. With different computational costs

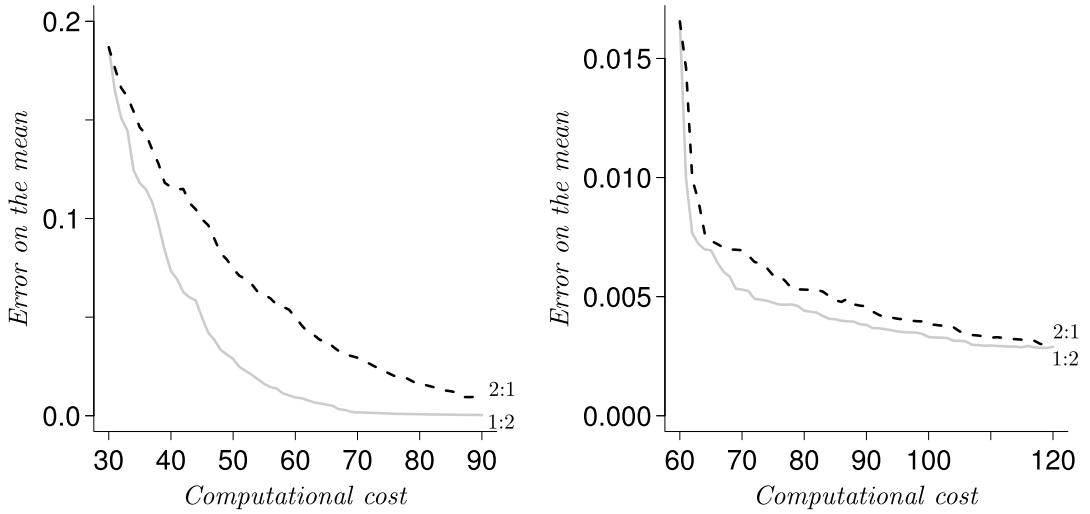
Figure 8 shows the prediction mean accuracy with the best I-optimal sequential design when the costs of the two codes are different. Two cases are presented. The first one corresponds to the case where the cost associated with the first code is twice the one associated with the second code, that means $\tau_1 = 2$ and $\tau_2 = 1$, the second corresponds to the case where the cost associated with the second code is twice the one associated with the first code, that means $\tau_1 = 1$ and $\tau_2 = 2$.

It can be seen that for both examples, the prediction accuracy at a given total computational cost is better when the cost of the first code is lower, that means when more observation points of the first code can be added for the same computational budget. These results are consistent with those of figure 7.



(A) Analytical example (B) Hydrodynamic example

FIGURE 7. Comparison of the number of evaluations of each code in the case of a sequential best I-optimal design applied to both examples. The curves correspond to the median of 50 draws of the initial design. The costs of the two codes are assumed to be the same.



(A) Analytical example (B) Hydrodynamic example

FIGURE 8. Performances of the best I-optimal sequential design in terms of prediction mean accuracy with different computational costs for the two codes. 1:2 $\leftrightarrow$ $\tau_1 = 1$ and $\tau_2 = 2$, 2:1 $\leftrightarrow$ $\tau_1 = 2$ and $\tau_2 = 1$. The curves correspond to the median of 50 draws of the initial maximin LHS design on $\mathbb{X}_{\text{nest}}$. The initial designs are the same for the two curves corresponding to each example and contain 15 observations and 30 observations on both codes for the analytical and the hydrodynamical example.

5. Conclusions and future work

In this paper the Gaussian process formalism is adapted to the case of two nested computer codes.

Two methods to compute quickly the mean and variance of the nested code predictor have been proposed. The first one, called "analytical" computes the exact value of the two first moments of the predictor. But it

cannot be applied to the coupling of more than two codes. The second one, called "linearized", enables to obtain a Gaussian predictor of the nested code, with mean and variance that can be instantly computed. The approach could be generalized to the coupling of more than two codes.

Both proposed methods take into account the intermediary information, that means the output of the first code. A comparison with the reference method, called "blind box", is made. In this method a Gaussian process regression of the block of the two codes is made without considering the intermediary observations. The numerical examples illustrate the interest of taking into account the intermediary information in terms of prediction mean accuracy.

Moreover, two sequential designs are proposed in order to improve the prediction accuracy of the nested predictor. The first one, the "chained" I-optimal sequential design, corresponds to the case where the two codes cannot be launched separately. The second one, the "best" I-optimal sequential design, allows to choose to which of the two codes to add a new observation point and to take into account the different computational costs of the two codes.

The numerical applications show the interest of the sequential designs compared with a space-filling design (maximin LHS). Furthermore, they illustrate the advantage, in terms of prediction mean accuracy, of choosing to which code to add a new observation point compared with simply adding new observation points of the nested code. The results show an amplification of the uncertainties in the chain of codes, leading to the addition of observation points of the first code firstly in the best I-optimal sequential design. It can be assumed that this should be similar with the coupling of more than two codes. In other words, the uncertainty of the beginning of the chain should be reduced as a priority.

This paper has been focused on the case of two nested codes with a scalar intermediary variable. Considering the case of a functional intermediary variable seems promising for future work.

Appendix A. Proof of Proposition 2.1

According to equation (2.5):

$$Y_i^c\left(\boldsymbol{x_i}\right) \overset{d}{=} \mu_i^c\left(\boldsymbol{x_i}\right) + \sigma_i^c\left(\boldsymbol{x_i}\right)\xi_i, \quad \xi_i \sim \mathcal{N}(0,1), \quad i \in \{1,2\},$$

where ξ_1 and ξ_2 are independent according to the independence of the initial processes Y_1 and Y_2 and the fact that $Y_i^c := Y_i|\boldsymbol{y}_i^{\text{obs}}$.

Therefore the process modeling the nested code can be written:

$$\begin{aligned} Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2) = &\quad Y_2^c(Y_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2) \\ = &\quad \mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right) + \sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\xi_2. \end{aligned}$$

Given the independence of ξ_1 and ξ_2 and the fact that $\mathbb{E}\left(\xi_2\right) = 0$, it can be inferred that the first moment of Y_{nest}^c can be written:

$$\mathbb{E}\left(Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)\right) = \mathbb{E}\left(\mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\right).$$

By noting that:

$$\begin{aligned} \left(Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)\right)^2 = &\quad \left(Y_2^c(Y_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)\right)^2 \\ = &\quad \left(\mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right) + \sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\xi_2\right)^2 \\ = &\quad \left(\mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\right)^2 + \left(\sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\right)^2\xi_2^2 \\ &+2\mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\xi_2, \end{aligned}$$

- ξ_1 and ξ_2 are independent,
- $\mathbb{E}(\xi_2) = 0$ and $\mathbb{E}\left(\xi_2^2\right) = 1$,

the second moment of Y^c_{nest} can be written:

$$\mathbb{E}\Big(\left(Y_2^c(Y_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2)\right)^2 \Big) = \mathbb{E}\left[\begin{array}{l} \left(\mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\right)^2 \\ + \left(\sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right) + \sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1, \boldsymbol{x_2}\right)\right)^2 \end{array} \right].$$

Appendix B. Proof of Lemma 3.1

If $X \sim \mathcal{N}(\mu, \sigma^2)$ and $g(x, a, b, c) := x^a \exp\left[bx + cx^2\right]$, then the mean of $g(x, a, b, c)$ is equal to:

$$\mathbb{E}\left[g(X, a, b, c)\right] = \int_{\mathbb{R}} g(x, a, b, c) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \mathrm{d}x.$$

It can be rewritten:

$$\begin{aligned} \mathbb{E}\left[g(X, a, b, c)\right] &= \int_{\mathbb{R}} x^a \exp\left(bx + cx^2\right) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \mathrm{d}x \\ &= \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\left(\sigma^2 b + \mu\right)^2}{2c\sigma^2 - 1} + \mu^2\right)\right) \int_{\mathbb{R}} x^a \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{1-2c\sigma^2}{\sigma^2}\left(x - \frac{\sigma^2 b + \mu}{1-2c\sigma^2}\right)^2\right) \mathrm{d}x \\ &= \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\left(\sigma^2 b + \mu\right)^2}{2c\sigma^2 - 1} + \mu^2\right)\right) \frac{1}{\sqrt{1-2c\sigma^2}} \mathbb{E}\left[X_g^a\right], \end{aligned}$$

where $X_g \sim \mathcal{N}\left(\frac{\sigma^2 b + \mu}{1-2c\sigma^2}, \frac{\sigma^2}{1-2c\sigma^2}\right)$, under the condition that $1 - 2c\sigma^2 > 0$.

Moreover, for $Y \sim \mathcal{N}\left(\mu_Y, \sigma_Y^2\right)$, any moment of order k, $k \in \mathbb{N}$, of Y can be computed analytically [22]:

$$\mathbb{E}\left[Y^k\right] = \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{2i} \mu_Y^{k-2i} \frac{(2i)!}{2^i i!} \sigma_Y^{2i}.$$

Hence, given that all the moments of a Gaussian variable can be computed analytically, the mean $\mathbb{E}\left[g(X, a, b, c)\right]$ can be computed analytically, and its expression is:

$$\mathbb{E}\left[g(X, a, b, c)\right] = \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\left(\sigma^2 b + \mu\right)^2}{2c\sigma^2 - 1} + \mu^2\right)\right) \frac{1}{\sqrt{1-2c\sigma^2}} \sum_{i=0}^{\lfloor \frac{a}{2} \rfloor} \binom{a}{2i} \left(\frac{\sigma^2 b + \mu}{1-2c\sigma^2}\right)^{a-2i} \frac{(2i)!}{2^i i!} \left(\frac{\sigma^2}{1-2c\sigma^2}\right)^i. \tag{B.1}$$

Appendix C. Proof of Lemma 3.2

We have:

$$\begin{aligned} g(x, a_i, b_i, c_i)\, g(x, a_j, b_j, c_j) &= x^{a_i} x^{a_j} \exp\left(b_i x + c_i x^2 + b_j x + c_j x^2\right) \\ &= x^{a_i + a_j} \exp\left((b_i + b_j)x + (c_i + c_j)x^2\right) \\ &= g(x, a_i + a_j, b_i + b_j, c_i + c_j). \end{aligned}$$

Appendix D. Proof of Proposition 3.3

D.1 First moment

In the framework of Universal Kriging, according to equation (2.10) the conditional mean function of the process modeling the second code can be written:

$$\begin{aligned}\mu_2^c(\varphi_1, \boldsymbol{x}_2) &= \boldsymbol{h_2}(\varphi_1, \boldsymbol{x}_2)^T \widehat{\boldsymbol{\beta}}_2 + C_2\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}_2^{\text{obs}}\right) \boldsymbol{v}_c \\ &= \sum_{i=1}^{M_2} \left(\boldsymbol{h_2}(\varphi_1, \boldsymbol{x}_2)\right)_i \left(\widehat{\boldsymbol{\beta}}_2\right)_i + \sum_{i=1}^{N_1} C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1^{(i)}, \boldsymbol{x}_2^{(i)}\right)\right) (\boldsymbol{v}_c)_i \\ &= (1) + (2),\end{aligned} \tag{D.1}$$

where $\varphi_1 \sim \mathcal{N}\left(\mu_1^c, (\sigma_1^c)^2\right)$, and

$$\boldsymbol{v}_c = \left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} \left[\boldsymbol{y}_2^{\text{obs}} - \boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T \widehat{\boldsymbol{\beta}}_2\right]. \tag{D.2}$$

According to the assumptions of Proposition 3.3 the mean basis functions $\boldsymbol{h}_2$ can be written:

$$\left(\boldsymbol{h}_2(\varphi_1, \boldsymbol{x}_2)\right)_i = m_i(\boldsymbol{x}_2)\ g(\varphi_1, a_i, 0, 0),$$

with m_i deterministic functions and $g(x, a, b, c) := x^a \exp\left(bx + cx^2\right),\ (a, b, c) \in \mathbb{N} \times \mathbb{R}^2$.

In the same way, the covariance function C_2 is in the Gaussian class, so according to equation (2.6), it can be written:

$$C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1', \boldsymbol{x}_2'\right)\right) = \sigma_2^2 k\left(\frac{\varphi_1 - \varphi_1'}{\ell_{\varphi_1}}\right) \prod_{i=1}^{d_2} k\left(\frac{(\boldsymbol{x}_2)_i - (\boldsymbol{x}_2')_i}{\ell_i}\right),$$

with $k : x \mapsto \exp\left(-x^2\right)$. So, one can write that:

$$\begin{aligned}C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1', \boldsymbol{x}_2'\right)\right) &= k\left(\frac{\varphi_1 - \varphi_1'}{\ell_{\varphi_1}}\right) \ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2'\right), \\ C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1', \boldsymbol{x}_2'\right)\right) &= \exp\left(-\left(\frac{\varphi_1'}{\ell_{\varphi_1}}\right)^2\right) g\left(\varphi_1, 0, \frac{2\varphi_1'}{\ell_{\varphi_1}^2}, \frac{-1}{\ell_{\varphi_1}^2}\right) \ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2'\right),\end{aligned} \tag{D.3}$$

where ℓ is a deterministic function defined by:

$$\ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2'\right) = \sigma_2^2 \prod_{i=1}^{d_2} \exp\left(-\left(\frac{(\boldsymbol{x}_2)_i - (\boldsymbol{x}_2')_i}{\ell_i}\right)^2\right), \tag{D.4}$$

with $\ell_i, 1 \leqslant i \leqslant d_2$ the correlation lengths associated with $\boldsymbol{x}_2$.

So the terms (1) and (2) of the equation (D.1) can be written:

$$\begin{aligned}(1) &= \sum_{i=1}^{M_2} g(\varphi_1, a_i, 0, 0)\ \ m_i(\boldsymbol{x}_2) \left(\widehat{\boldsymbol{\beta}}_2\right)_i, \\ (2) &= \sum_{i=1}^{N_1} (\boldsymbol{v}_c)_i\, \ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) \exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right) g\left(\varphi_1, 0, \frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2}, \frac{-1}{\ell_{\varphi_1}^2}\right).\end{aligned}$$

According to the fact that m_i and ℓ are deterministic functions, $\widehat{\boldsymbol{\beta}}_2$, $\boldsymbol{v}_c$, $\boldsymbol{x}_2^{(i)}$ and $\boldsymbol{x}_2$ deterministic vectors, and $\varphi_1^{(i)}$ deterministic real numbers, then:

$$\mathbb{E}\left[(1)\right] = \sum_{i=1}^{M_2} \mathbb{E}\left[g\left(\varphi_1, a_i, 0, 0\right)\right]\ m_i(\boldsymbol{x}_2)\left(\widehat{\boldsymbol{\beta}}_2\right)_i,$$

$$\mathbb{E}\left[(2)\right] = \sum_{i=1}^{N_1} \left(\boldsymbol{v}_c\right)_i \ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) \exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right) \mathbb{E}\left[g\left(\varphi_1, 0, \frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2}, \frac{-1}{\ell_{\varphi_1}^2}\right)\right].$$

According to Lemma 3.1, and the fact that $1 - 2\left(\frac{-1}{\ell_{\varphi_1}^2}\right)\sigma^2 > 0$, the means $\mathbb{E}\left[(1)\right]$ and $\mathbb{E}\left[(2)\right]$ can be calculated analytically, and consequently, the mean $\mathbb{E}\left[\mu_2^c\left(\varphi_1, \boldsymbol{x}_2\right)\right]$ can be calculated analytically, and its expression is:

$$\begin{aligned}\mathbb{E}\left[\mu_2^c\left(\varphi_1, \boldsymbol{x}_2\right)\right] = &\sum_{i=1}^{M_2} \mathbb{E}\left[g\left(\varphi_1, a_i, 0, 0\right)\right]\ m_i(\boldsymbol{x}_2)\left(\widehat{\boldsymbol{\beta}}_2\right)_i \\ &+ \sum_{i=1}^{N_1} \left(\boldsymbol{v}_c\right)_i \ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) \exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right) \mathbb{E}\left[g\left(\varphi_1, 0, \frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2}, \frac{-1}{\ell_{\varphi_1}^2}\right)\right],\end{aligned} \tag{D.5}$$

where $\boldsymbol{v}_c$ is defined by equation (D.2), $\ell\left(\boldsymbol{x}_2 - \boldsymbol{x}_2'\right)$ is defined by equation (D.4), ℓ_{φ_1} is the correlation length associated with φ_1 and $\widehat{\boldsymbol{\beta}}_2$ is given by equation (2.9).

D.2 Second moment

From equation (2.10) and (2.12), we have:

$$\mu_2^c\left(\varphi_1, \boldsymbol{x}_2\right) = \boldsymbol{h}_2\left(\varphi_1, \boldsymbol{x}_2\right)^T \widehat{\boldsymbol{\beta}}_2 + C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \bar{\boldsymbol{x}}_2^{\text{obs}}\right) \boldsymbol{v}_c,$$

and:

$$\begin{aligned}\left(\sigma_2^c\left(\varphi_1, \boldsymbol{x}_2\right)\right)^2 = &\ C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1, \boldsymbol{x}_2\right)\right) - C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \left(\varphi_1, \boldsymbol{x}_2\right)\right) \\ &+ \left[\boldsymbol{h}_2\left(\varphi_1, \boldsymbol{x}_2\right)^T - C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} \boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\right] \\ &\times \left[\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} \boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\right]^{-1} \\ &\times \left[\boldsymbol{h}_2\left(\varphi_1, \boldsymbol{x}_2\right) - \boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \left(\varphi_1, \boldsymbol{x}_2\right)\right)\right],\end{aligned}$$

Hence, it can be written that:

$$\begin{aligned}\left(\mu_2^c\left(\varphi_1, \boldsymbol{x}_2\right)\right)^2 + \left(\sigma_2^c\left(\varphi_1, \boldsymbol{x}_2\right)\right)^2 = \sigma_2^2 &+ \underbrace{\boldsymbol{h_2}\left(\varphi_1, \boldsymbol{x}_2\right)^T \boldsymbol{A}_h \boldsymbol{h_2}\left(\varphi_1, \boldsymbol{x}_2\right)}_{(1)} + \underbrace{C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \bar{\boldsymbol{x}}_2^{\text{obs}}\right) \boldsymbol{A}_c\ C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}, \left(\varphi_1, \boldsymbol{x}_2\right)\right)}_{(2)} \\ &+ \underbrace{C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \bar{\boldsymbol{x}}_2^{\text{obs}}\right)}_{(3)} \boldsymbol{A}_{ch}\ \boldsymbol{h_2}\left(\varphi_1, \boldsymbol{x}_2\right),\end{aligned} \tag{D.6}$$

where:

$$\begin{aligned}
\boldsymbol{A}_h = &\ \widehat{\boldsymbol{\beta}}_2\widehat{\boldsymbol{\beta}}_2^T + \left(\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\right)^{-1},\\
\boldsymbol{A}_c = &\ \boldsymbol{v}_c\boldsymbol{v}_c^T - \left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1} + \left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\\
&\times\left[\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\right]^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1},\\
\boldsymbol{A}_{ch} = &\ 2\boldsymbol{v}_c\widehat{\boldsymbol{\beta}}_2^T - 2\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\left[\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\left(C_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}},\bar{\boldsymbol{x}}_2^{\text{obs}}\right)\right)^{-1}\boldsymbol{h}_2\left(\bar{\boldsymbol{x}}_2^{\text{obs}}\right)^T\right]^{-1}.
\end{aligned} \tag{D.7}$$

According to the assumptions of Proposition 3.3 and to Lemma 3.2, the terms (1), (2) and (3) of the equation (D.6) can be rewritten:

$$\begin{aligned}
(1) = &\ \sum_{i=1}^{M_2}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_h\right)_{ij}\left(\boldsymbol{h_2}\left(\varphi_1,\boldsymbol{x}_2\right)\right)_i\left(\boldsymbol{h_2}\left(\varphi_1,\boldsymbol{x}_2\right)\right)_j = \sum_{i=1}^{M_2}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_h\right)_{ij} m_i\left(\boldsymbol{x}_2\right) m_j\left(\boldsymbol{x}_2\right) g\left(\varphi_1,a_i,0,0\right) g\left(\varphi_1,a_j,0,0\right)\\
= &\ \sum_{i=1}^{M_2}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_h\right)_{ij} m_i\left(\boldsymbol{x}_2\right) m_j\left(\boldsymbol{x}_2\right) g\left(\varphi_1,a_i+a_j,0,0\right),
\end{aligned}$$

$$\begin{aligned}
(2) = &\ \sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\left(\boldsymbol{A}_c\right)_{ij} C_2\left(\left(\varphi_1,\boldsymbol{x}_2\right),\left(\varphi_1^{(i)},\boldsymbol{x}_2^{(i)}\right)\right) C_2\left(\left(\varphi_1,\boldsymbol{x}_2\right),\left(\varphi_1^{(j)},\boldsymbol{x}_2^{(j)}\right)\right)\\
= &\ \sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\left(\boldsymbol{A}_c\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(j)}\right)\exp\left(-\frac{\left(\varphi_1^{(i)}\right)^2+\left(\varphi_1^{(j)}\right)^2}{\ell_{\varphi_1}^2}\right) g\left(\varphi_1,0,\frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right)\\
&\times g\left(\varphi_1,0,\frac{2\varphi_1^{(j)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right)\\
= &\ \sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\left(\boldsymbol{A}_c\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(j)}\right)\exp\left(-\frac{\left(\varphi_1^{(i)}\right)^2+\left(\varphi_1^{(j)}\right)^2}{\ell_{\varphi_1}^2}\right) g\left(\varphi_1,0,2\frac{\varphi_1^{(i)}+\varphi_1^{(j)}}{\ell_{\varphi_1}^2},\frac{-2}{\ell_{\varphi_1}^2}\right),
\end{aligned}$$

$$\begin{aligned}
(3) = &\ \sum_{i=1}^{N_1}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_{ch}\right)_{ij} C_2\left(\left(\varphi_1,\boldsymbol{x}_2\right),\left(\varphi_1^{(i)},\boldsymbol{x}_2^{(i)}\right)\right)\left(\boldsymbol{h_2}\left(\varphi_1,\boldsymbol{x}_2\right)\right)_j\\
= &\ \sum_{i=1}^{N_1}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_{ch}\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right) m_j\left(\boldsymbol{x}_2\right) g\left(\varphi_1,0,\frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right) g\left(\varphi_1,a_j,0,0\right)\\
= &\ \sum_{i=1}^{N_1}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_{ch}\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right) m_j\left(\boldsymbol{x}_2\right) g\left(\varphi_1,a_j,\frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right).
\end{aligned}$$

According to the fact that m_i and ℓ are deterministic functions, $\boldsymbol{x}_2$ and $\boldsymbol{x}_2^{(i)}$ deterministic vectors, $\boldsymbol{A}_h$, $\boldsymbol{A}_c$ and $\boldsymbol{A}_{ch}$ deterministic matrices, and $\varphi_1^{(i)}$ and ℓ_{φ_1} deterministic real numbers, it can be written:

$$\mathbb{E}\left[(1)\right] = \sum_{i=1}^{M_2}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_h\right)_{ij} m_i\left(\boldsymbol{x}_2\right) m_j\left(\boldsymbol{x}_2\right)\mathbb{E}\left[g\left(\varphi_1,a_i+a_j,0,0\right)\right],$$

$$\mathbb{E}\left[(2)\right] = \sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\left(\boldsymbol{A}_c\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(j)}\right)\exp\left(-\frac{\left(\varphi_1^{(i)}\right)^2+\left(\varphi_1^{(j)}\right)^2}{\ell_{\varphi_1}^2}\right)$$
$$\times\,\mathbb{E}\left[g\left(\varphi_1,0,2\frac{\varphi_1^{(i)}+\varphi_1^{(j)}}{\ell_{\varphi_1}^2},\frac{-2}{\ell_{\varphi_1}^2}\right)\right],$$

$$\mathbb{E}\left[(3)\right] = \sum_{i=1}^{N_1}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_{ch}\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right)m_j\left(\boldsymbol{x}_2\right)\mathbb{E}\left[g\left(\varphi_1,a_j,\frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right)\right].$$

Hence, according to the Lemma 3.1, the mean $\mathbb{E}\left[(1)\right]$ can be computed analytically. In the same way, according to the Lemma 3.1, and the fact that $1-4\left(\frac{-1}{\ell_{\varphi_1}^2}\right)(\sigma)^2>0$ and $1-2\left(\frac{-1}{\ell_{\varphi_1}^2}\right)(\sigma)^2>0$, the means $\mathbb{E}\left[(2)\right]$ and $\mathbb{E}\left[(3)\right]$ can be calculated analytically. Consequently, the mean $\mathbb{E}\left[\left(\mu_2^c\left(\varphi_1,\boldsymbol{x}_2\right)\right)^2+\left(\sigma_2^c\left(\varphi_1,\boldsymbol{x}_2\right)\right)^2\right]$ can be calculated analytically, and its expression is:

$$\mathbb{E}\left[\left(\mu_2^c\left(\varphi_1,\boldsymbol{x}_2\right)\right)^2+\left(\sigma_2^c\left(\varphi_1,\boldsymbol{x}_2\right)\right)^2\right] = \sigma_2^2+\sum_{i=1}^{M_2}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_h\right)_{ij}m_i\left(\boldsymbol{x}_2\right)m_j\left(\boldsymbol{x}_2\right)\mathbb{E}\left[g\left(\varphi_1,a_i+a_j,0,0\right)\right]$$
$$+\sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\left(\boldsymbol{A}_c\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(j)}\right)\exp\left(-\frac{\left(\varphi_1^{(i)}\right)^2+\left(\varphi_1^{(j)}\right)^2}{\ell_{\varphi_1}^2}\right)\mathbb{E}\left[g\left(\varphi_1,0,2\frac{\varphi_1^{(i)}+\varphi_1^{(j)}}{\ell_{\varphi_1}^2},\frac{-2}{\ell_{\varphi_1}^2}\right)\right]$$
$$+\sum_{i=1}^{N_1}\sum_{j=1}^{M_2}\left(\boldsymbol{A}_{ch}\right)_{ij}\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{(i)}\right)m_j\left(\boldsymbol{x}_2\right)\exp\left(-\left(\frac{\varphi_1^{(i)}}{\ell_{\varphi_1}}\right)^2\right)\mathbb{E}\left[g\left(\varphi_1,a_j,\frac{2\varphi_1^{(i)}}{\ell_{\varphi_1}^2},\frac{-1}{\ell_{\varphi_1}^2}\right)\right], \tag{D.8}$$

where $\boldsymbol{A}_h$, $\boldsymbol{A}_c$ and $\boldsymbol{A}_{ch}$ are defined in equation (D.7), $\boldsymbol{v}_c$ is defined in equation (D.2), $\ell\left(\boldsymbol{x}_2-\boldsymbol{x}_2'\right)$ is defined by equation (D.4), ℓ_{φ_1} is the correlation length associated with φ_1 and $\widehat{\boldsymbol{\beta}}_2$ is given by equation (2.9).

From the two previous paragraphs and Proposition 2.1, it can be inferred that, if verifying the assumptions of Proposition 3.3, then the first and the second moments of $Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)$ can be calculated analytically.

Appendix E. Proof of Proposition 3.4

If $Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)=Y_2^c(Y_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)$ where $Y_i^c=\mu_i^c+\varepsilon_i^c$, $\varepsilon_i^c\sim\text{GP}\left(0,C_i^c\right)$, $i\in\{1,2\}$, then, if ε_1^c is small enough, the process $Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)$ can be linearized:

$$\begin{aligned}Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2) &= \mu_2^c(\mu_1^c(\boldsymbol{x}_1)+\varepsilon_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)+\varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1)+\varepsilon_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2),\\ &\approx \mu_2^c(\mu_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)+\frac{\partial\mu_2^c}{\partial\varphi_1}(\mu_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)\varepsilon_1^c(\boldsymbol{x}_1)+\varepsilon_2^c(\mu_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2).\end{aligned}$$

So it can be written:

$$Y_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2)\approx\mu_{\text{nest}}^c+\varepsilon_{\text{nest}}^c(\mu_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2), \tag{E.1}$$

with

$$\mu_{\text{nest}}^c\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)=\mu_2^c(\mu_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2), \tag{E.2}$$

and

$$\varepsilon^c_{\text{nest}} = \frac{\partial \mu^c_2}{\partial \varphi_1}(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2)\varepsilon^c_1(\boldsymbol{x}_1) + \varepsilon^c_2(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2). \tag{E.3}$$

ε^c_1 and ε^c_2 are independent centred Gaussian processes, so $\varepsilon^c_{\text{nest}}$ is a centred Gaussian process, whose covariance function, C^c_{nest}, is given by:

$$\begin{aligned} C^c_{\text{nest}}((\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}'_1, \boldsymbol{x}'_2)) = {} & C^c_2((\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2), (\mu^c_1(\boldsymbol{x}'_1), \boldsymbol{x}'_2)) \\ & + \frac{\partial \mu^c_2}{\partial \varphi_1}\left((\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2)\right) \frac{\partial \mu^c_2}{\partial \varphi_1}\left((\mu^c_1(\boldsymbol{x}'_1), \boldsymbol{x}'_2)\right) C^c_1(\boldsymbol{x}_1, \boldsymbol{x}'_1). \end{aligned} \tag{E.4}$$

From Eqs (E.1), (E.2), (E.3) and (E.5), it can be inferred that the predictor of the nested code can be defined as a Gaussian process with mean μ^c_{nest} defined by equation (E.2), and covariance function C^c_{nest} defined by equation (E.5).

Moreover, it can be inferred from equation (2.10):

$$\frac{\partial \mu^c_2}{\partial \varphi_1}(\varphi_1, \boldsymbol{x}_2) = \left(\frac{\partial \boldsymbol{h}_2}{\partial \varphi_1}(\varphi_1, \boldsymbol{x}_2)\right)^T \widehat{\boldsymbol{\beta}}_2 + \frac{\partial C^c_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}^{\text{obs}}_2\right)\left(C_2\left(\bar{\boldsymbol{x}}^{\text{obs}}_2, \bar{\boldsymbol{x}}^{\text{obs}}_2\right)\right)^{-1}\left[\boldsymbol{y}^{\text{obs}}_2 - \boldsymbol{h}_2\left(\bar{\boldsymbol{x}}^{\text{obs}}_2\right)^T \widehat{\boldsymbol{\beta}}_2\right]. \tag{E.5}$$

Appendix F. Proof of Corollary 3.5

F.1 Explicit mean

According to equation (2.10), if $\boldsymbol{h}_i$ and C_i can be computed explicitly, then μ^c_i can be computed explicitly. Therefore, according to equation (3.4), the mean of the Gaussian linearized predictor can be computed explicitly.

F.2 Explicit variance

According to equation (2.12), if $\boldsymbol{h}_i$ and C_i can be computed explicitly, then C^c_i can be computed explicitly.

According to equation (E.5), if $\boldsymbol{h}_2$, C_2 and the derivatives $\frac{\partial \boldsymbol{h}_2}{\partial \varphi_1}(\varphi_1, \boldsymbol{x}_2)$ and $\frac{\partial C_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}^{\text{obs}}_2\right)$ can be computed explicitly, then the derivative of μ^c_2 with respect to φ_1 can be computed explicitly.

Therefore, according to equation (3.4), the variance of the Gaussian linearized predictor can be computed explicitly.

Hence it can be inferred that, if $\boldsymbol{h}_i$ and C_i and the derivatives $\frac{\partial \boldsymbol{h}_2}{\partial \varphi_1}(\varphi_1, \boldsymbol{x}_2)$ and $\frac{\partial C_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}^{\text{obs}}_2\right)$ can be computed explicitly, then the mean and the variance of the Gaussian linearized predictor of the nested code can be computed explicitly.

Moreover, the derivative $\frac{\partial C_2}{\partial \varphi_1}\left((\varphi_1, \boldsymbol{x}_2), \bar{\boldsymbol{x}}^{\text{obs}}_2\right)$ can be computed explicitly if C_2 is in the Gaussian or Matérn $\frac{5}{2}$ class, and the associated explicit formulas are given in what follows.

F.3 Matérn class

If we denote by:

$$\begin{aligned}\delta = & \quad d\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right) \\ = & \quad \sqrt{\frac{\left(\varphi_1 - \varphi_1^{'}\right)^2}{\ell_{\varphi_1}^2} + \sum_{i=1}^{d_2} \frac{\left(\left(\boldsymbol{x}_2\right)_i - \left(\boldsymbol{x}_2\right)_i\right)^2}{\ell_i^2}},\end{aligned} \tag{F.1}$$

then, according to equation (2.7), the Matérn kernel can be rewritten:

$$K_{\frac{5}{2}}\left(\delta\right) = \left(1 + \sqrt{5}\delta + \frac{5}{3}\delta^2\right) \exp\left(-\sqrt{5}\delta\right). \tag{F.2}$$

Moreover, we have:

$$\frac{\partial \delta}{\partial \varphi_1} = \frac{\varphi_1 - \varphi_1^{'}}{\ell_{\varphi_1}^2} \frac{1}{\delta}, \tag{F.3}$$

and

$$\begin{aligned}\frac{\partial K_{\frac{5}{2}}}{\partial \delta}\left(\delta\right) &= \exp\left(-\sqrt{5}\delta\right)\left[-\sqrt{5}\left(1 + \sqrt{5}\delta + \frac{5}{3}\delta^2\right) + \sqrt{5} + \frac{10}{3}\delta\right] \\ &= \exp\left(-\sqrt{5}\delta\right)\left[-5\delta - \sqrt{5}\frac{5}{3}\delta^2 + \frac{10}{3}\delta\right] \\ &= -\frac{5}{3}\delta\left(1 + \sqrt{5}\delta\right)\exp\left(-\sqrt{5}\delta\right),\end{aligned} \tag{F.4}$$

By noting that in the case of a Matérn $\frac{5}{2}$ kernel:

$$\frac{\partial C_2}{\partial \varphi_1} = \frac{\partial K_{\frac{5}{2}}}{\partial \delta} \frac{\partial \delta}{\partial \varphi_1},$$

the derivative of C_2 with respect to φ_1 is:

$$\frac{\partial C_2}{\partial \varphi_1}\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right) = -\frac{5}{3}\frac{\varphi_1 - \varphi_1^{'}}{\ell_{\varphi_1}^2}\left[1 + \sqrt{5}\ d\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right)\right] \exp\left[-\sqrt{5}\ d\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right)\right]. \tag{F.5}$$

F.4 Gaussian class

According to equation (2.6), the Gaussian kernel can be rewritten:

$$K_{\text{Gauss}}\left(\delta\right) = \exp\left(-\delta^2\right). \tag{F.6}$$

Hence, we have:

$$\frac{\partial K_{\text{Gauss}}}{\partial \delta}(\delta) = -2\delta \exp\left(-\delta^2\right). \tag{F.7}$$

By noting that, in the case of a Gaussian kernel:

$$\frac{\partial C_2}{\partial \varphi_1} = \frac{\partial K_{\text{Gauss}}}{\partial \delta} \frac{\partial \delta}{\partial \varphi_1},$$

the derivative of C_2 with respect to φ_1 is:

$$\frac{\partial C_2}{\partial \varphi_1}\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right) = -2\frac{\varphi_1 - \varphi_1^{'}}{\ell_{\varphi_1}^2} \exp\left[-d\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right)^2\right]. \tag{F.8}$$

Acknowledgements. The authors thank the anonymous reviewers for their constructive comments, which helped to improve the manuscript.

References

[1] F. Bachoc, Cross validation and maximum likelihood estimation of hyper-parameters of Gaussian processes with model misspecification. *Comput. Stat. Data Anal.* **66** (2013) 55–69.
[2] F. Bachoc, *Parametric estimation of covariance function in Gaussian-process based Kriging models. Application to uncertainty 520 quantification for computer experiments*. Ph.D. thesis. Université Paris-Diderot – Paris VII (2013).
[3] C.T.H. Baker, The Numerical Treatment of Integral Equations. Clarendon Press, Oxford (1977).
[4] J. Bect, D. Ginsbourger, L. Li, V. Picheny and E. Vazquez, Sequential design of computer experiments for the estimation of a probability of failure. *Stat. Comput.* **22** (2012) 773–793.
[5] J.O. Berger, V. De Oliveira and B. Sansó, Objective Bayesian analysis of spatially correlated data. *J. Am. Stat. Assoc.* **96** (2001) 1361–1374.
[6] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan and J.M. Mcfarland, Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA J.* **46** (2008) 2459–2468.
[7] C. Chevalier, J. Bect, D. Ginsbourger and E. Vazquez, Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics* **56** (2014) 455–465.
[8] A. Damianou and N.D. Lawrence, Deep Gaussian processes, in *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS), AISTATS '13, pages 207–215, JMLR W&CP 31*, edited by C. Carvalho and P. Ravikumar (2013).
[9] B. Echard, N. Gayton and M. Lemaire, AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Struct. Saf.* **33** (2011) 145–154.
[10] K.T. Fang and D.K. Lin, Uniform experimental designs and their applications in industry. *Handb. Stat.* **22** (2003) 131–178.
[11] K.T. Fang, R. Li and A. Sudjianto, Design and Modeling for Computer Experiments. Computer Science and Data Analysis Series. Chapman Hall, London (2006).
[12] D. Ginsbourger, R. Le Riche and L. Carraro, Kriging is well-suited to parallelize optimization, in Computational Intelligence in Expensive Optimization Problems. Vol. 2 of *Adaptation Learning and Optimization*. Springer, Berlin, Heidelberg (2010) 131–162.
[13] R.B. Gramacy and H.K.H. Lee, Cases for the nugget in modeling computer experiments. *Stat. Comput.* **22** (2012) 713–722.
[14] R. Gramacy and H. Lian, Gaussian process single-index models as emulators for computer experiments. *Technometrics* **54** (2012) 30–41.
[15] C. Helbert, D. Dupuy and L. Carraro, Assessment of uncertainty in computer experiments, from Universal to Bayesian Kriging. *Appl. Stoch. Model. Bus. Ind.* **25** (2009) 99–113.
[16] R. Hu and M. Ludkovski, Sequential design for ranking response surfaces. *SIAM/ASA J. Uncertain. Quantif.* **5** (2017) 212–239.
[17] M.C. Kennedy and A. O'Hagan, Predicting the output from a complex computer code when fast approximations are avalaible. *Biometrika* **87** (2000) 1–13.
[18] M.C. Kennedy and A. O'Hagan, Bayesian calibration of computer models. *J. Royal Stat. Soc. Ser. B (Stat. Methodol.)* **63** (2001) 425–464.
[19] J.P.C. Kleijnen, Regression and Kriging metamodels with their experimental designs in simulation: a review. *Eur. J. Oper. Res.* **256** (2017) 1–16.
[20] L. Le Gratiet Bayesian analysis of hierarchical multifidelity codes. *SIAM/ASA J. Uncertain. Quantif.* **1** (2013) 244–269.
[21] L. Le Gratiet and J. Garnier, Recursive co-Kriging model for design of computer experiments with multiple levels of fidelity. *Int. J. Uncertain. Quantif.* **4** (2014) 365–386.

[22] A. Papoulis and S.U. Pillai, Probability, Random Variables and Stochastic Processes. McGraw-Hill, Boston (2002).
[23] R. Paulo, Default priors for Gaussian processes. *Ann. Stat.* **33** (2005) 556–582.
[24] P. Perdikaris, M. Raissi, A. Damianou, N.D. Lawrence and G.E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **473** (2017) 20160751.
[25] G. Perrin, Active learning surrogate models for the conception of systems with multiple failure modes. *Reliab. Eng. Syst. Saf.* **149** (2016) 130–136.
[26] G. Perrin and C. Cannamela, A repulsion-based method for the definition and the enrichment of optimized space filling designs in constrained input spaces. *J. Soc. Française de Stat.* **158** (2017) 37–67.
[27] G. Perrin, C. Soize, S. Marque-Pucheu and J. Garnier, Nested polynomial trends for the improvement of Gaussian process-based predictors. *J. Comput. Phys.* **346** (2017) 389–402.
[28] V. Picheny and D. Ginsbourger, A nonstationary space-time Gaussian process model for partially converged simulations. *SIAM/ASA J. Uncertain. Quantif.* **1** (2013) 37–67.
[29] C.E. Rasmussen and C.K.I. Williams, Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2006).
[30] C. Robert, The Bayesian Choice. Springer-Verlag, New York (2007).
[31] J. Sacks, W. Welch, T.J. Mitchell and H.P. Wynn, Design and analysis of computer experiments. *Stat. Sci.* **4** (1989) 409–435.
[32] T.J. Santner, B.J. Williams and W. Notz, The Design and Analysis of Computer Experiments. *Springer Series in Statistics.* Springer, New York (2003).
[33] M.L. Stein, Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York (1999).
[34] R. Stroh, S. Demeyer, N. Fischer, J. Bect and E. Vazquez, Sequential design of experiments to estimate a probability of exceeding a threshold in a multi-fidelity stochastic simulator, in *61th World Statistics Congress of the International Statistical Institute (ISI 2017), Marrakech, Morocco, July 2017* (2017).
[35] R. Tuo, C.F Jeff Wu and D. Yu, Surrogate modeling of computer experiments with different mesh densities. *Technometrics* **56** (2014) 372–380.