# Efficient sequential experimental design for surrogate modeling of nested codes

Sophie Marque-Pucheu, Guillaume Perrin, Josselin Garnier

# Efficient sequential experimental design for surrogate modeling of nested codes

Sophie Marque-Pucheu[a,b], Guillaume Perrin[a], Josselin Garnier[c]

[a] *CEA/DAM/DIF, F-91297, Arpajon, France*
[b] *Laboratoire de Probabilités et Modèles Aléatoires, Université Paris Diderot, 75205 Paris Cedex 13, France*
[c] *Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France*

**Abstract**

Thanks to computing power increase, the certification and the conception of complex systems relies more and more on simulation. To this end, predictive codes are needed, which have generally to be evaluated in a huge number of input points. When the computational cost of these codes is high, surrogate models are introduced to emulate the response of these codes. In this paper, we consider the situation when the system response can be modeled by two nested computer codes. By two nested computer codes, we mean that some inputs of the second code are outputs of the first code. More precisely, the idea is to propose sequential designs to improve the accuracy of the nested code's predictor by exploiting the nested structure of the codes. In particular, a selection criterion is proposed to allow the modeler to choose the code to call, depending on the expected learning rate and the computational cost of each code. The sequential designs are based on the minimization of the prediction variance, so adaptations of the Gaussian process formalism are proposed for this particular configuration in order to quickly evaluate the mean and the variance of the predictor. The proposed methods are then applied to examples.

*Keywords:*
nested computer codes, surrogate model, Gaussian process, uncertainty quantification, Bayesian formalism.

## 1. Introduction

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 conception, 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

*Email address:* sophie.marque-pucheu@cea.fr (Sophie Marque-Pucheu)

clearly sub-optimal, as much information is lost in the statistical learning, such 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, which means that the output of the first code is an input 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 in a unobserved point, a Bayesian formalism [23] is adopted in the following. Each computer code is *a priori* modeled by a Gaussian process, and the idea is to identify the 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 sciences ([24, 25, 22, 14, 15, 4, 18, 16]), 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 ([24, 25, 3, 7, 6]), in order to minimize the global prediction uncertainty. These methods are generally based on a post-processing of the variance of the code output prediction, which expression can be explicitly derived under mildly 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, such that the prediction uncertainty is much more complicated to estimate. Moreover, if the two codes can be launched separately, the selection criterion has also to indicate which one of the two codes to launch. In that prospect, the first objective of this paper is to propose several adaptations of the Gaussian Process formalism to the nested case, in order to be able to evaluate the two first statistical moments of the code output predictor 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 cost 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 several selection criteria based on the prediction variance to reduce the prediction uncertainty sequentially. Section 3 introduces a series of simplifications to allow a quick evaluation of the prediction variance. In section 4, the presented methods are eventually applied to two examples.

The proofs of the results that will be presented in the following sections have been moved to the appendix.

## 2. Surrogate modeling for two nested computer codes

### *2.1. Notations*

In this paper, the following notations will be adopted:

- $x, y$ 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 $\mathbb{X}$:

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

### *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:

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

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 $y_{\text{nest}}$ on $\mathbb{X}_{\text{nest}}$.
Given a limited number of evaluations of the functions $\boldsymbol{x}_1 \mapsto y_1(\boldsymbol{x}_1)$ and $(\varphi_1, \boldsymbol{x}_2) \mapsto y_2(\varphi_1, \boldsymbol{x}_2)$, the objective is to build a stochastic predictor of $y_{\text{nest}}$ with the following properties:

- its mean is as close as possible to the real output of the nested code, that is, the bias is small,
- its uncertainty (given by its variance) is as small as possible.

In other words, the mean square error of the stochastic predictor has to be small.

*2.3. Gaussian process-based surrogate models*

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, $\widehat{y}_1$ and $\widehat{y}_2$, which are supposed to be Gaussian for the sake of tractability:

$$\widehat{y}_i \sim \text{GP}(\mu_i, C_i), \quad i \in \{1, 2\}, \tag{2.3}$$

where for all $1 \leq i \leq 2$, $\mu_i$ and $C_i$ denote respectively the mean and the covariance functions of $\widehat{y}_i$.
Let $\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_1^{(N_1)}$ be $N_1$ elements of $\mathbb{X}_1$ and $\left(\varphi_1^{(1)}, \boldsymbol{x}_2^{(1)}\right), \ldots, (\varphi_1^{(N_2)}, \boldsymbol{x}_2^{(N_2)})$ be $N_2$ elements of $\mathbb{R} \times \mathbb{X}_2$. Denoting by

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

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

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

and we refer to [24, 25] for further details about the expressions of conditioned mean functions, $\mu_i^c$, and conditioned covariance functions, $C_i^c$.
According to Eq. (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 $\widehat{y}_{\text{nest}}^c$, such that for all $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_1 \times \mathbb{X}_2$,

$$\widehat{y}_{\text{nest}}^c(\boldsymbol{x}_1, \boldsymbol{x}_2) := \widehat{y}_2^c(\widehat{y}_1^c(\boldsymbol{x}_1), \boldsymbol{x}_2). \tag{2.6}$$

Under this Gaussian formalism, the best prediction of $y_{\text{nest}}$ in 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 $\widehat{y}_{\text{nest}}^c(\boldsymbol{x}_1, \boldsymbol{x}_2)$, whereas its

variance can be used to characterize the trust we can put in that prediction. As explained in Introduction, there is no reason for $\widehat{y}^c_{\text{nest}}$ 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. This can be done by quadrature rules or by Monte-Carlo methods ([2]).

**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:*

$$\mathbb{E}\left[\widehat{y}^c_{nest}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right] = \mathbb{E}\left[\mu^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\right], \tag{2.7}$$

$$\mathbb{E}\left[\left(\widehat{y}^c_{nest}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right)^2\right] = \mathbb{E}\left[\begin{array}{l}\left\{\mu^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\right\}^2 \\ \quad + \left\{\sigma^c_2(\mu^c_1(\boldsymbol{x}_1) + \sigma^c_1(\boldsymbol{x}_1)\xi, \boldsymbol{x}_2)\right\}^2\end{array}\right], \tag{2.8}$$

*where for all* $i$ *in* $\{1,2\}$, $\left(\sigma^c_i\left(\boldsymbol{x}_i\right)\right)^2 = C^c_i\left(\boldsymbol{x}_i, \boldsymbol{x}_i\right)$.

*2.4. Parametric representations of the mean and covariance functions*

As explained in Introduction, the relevance of the Gaussian process predictor strongly depends on the definitions of $\mu_i$ and $C_i$. When the maximal 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 supposed to be two elements of the Matérn-5/2 class (see [25, 17] for further details about classical parametric expressions for $C_i$), with $\boldsymbol{\theta}_i$ be the hyper-parameters that characterize these covariance functions, whereas linear representations are considered for the mean functions,

$$\mu_i = \boldsymbol{h}_i^T \boldsymbol{\beta}_i, \tag{2.9}$$

where $\boldsymbol{h}_i$ is a given $M_i$-dimensional vector of functions (see [21] 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 the maximum likelihood estimate of $\boldsymbol{\theta}_i$,
- integrating over $\boldsymbol{\beta}_i$ the conditioned distribution of $\widehat{y}_i$.

In that case, the distribution of $\widehat{y}^c_i$, which is defined by Eq. 2.5 is Gaussian, and its statistical moments can explicitly be derived (see [24, 5, 3, 21]).

*2.5. Sequential designs for the improvement of Gaussian process predictors*

The relevance of the predictor $\widehat{y}^c_{\text{nest}}$ 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 Samplings (LHS) or quasi-Monte-Carlo samplings are generally chosen to define such *a priori* DoE ([9, 8, 20]). 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 $\|\mathbb{E}\left[\widehat{y}^c_{\text{nest}}\right] - y_{\text{nest}}\|^2_{\mathbb{X}_{\text{nest}}}$ to be small.

In the case of a single code, most of the existing selection criteria to add a new point are based on the minimization of a quantity associated with the predictor variance, such as its integral over the input domain for instance [24, 25, 7, 3, 6, 19, 13, 11]. Indeed, if $\widehat{z}$ is a Gaussian process that is indexed by $\boldsymbol{x}$ in $\mathbb{X}$, and if we denote by $k$ its covariance function, the variance of the conditioned random variable $\widehat{z}(\boldsymbol{x}) \mid \widehat{z}(\boldsymbol{x}^{\text{new}})$, where $\boldsymbol{x}$ and $\boldsymbol{x}^{\text{new}}$ are any elements of $\mathbb{X}$, is given by:

$$k(\boldsymbol{x}, \boldsymbol{x}) - k(\boldsymbol{x}, \boldsymbol{x}^{\text{new}})^2 / k(\boldsymbol{x}^{\text{new}}, \boldsymbol{x}^{\text{new}}), \tag{2.10}$$

such that it does not depend on the (unknown) value of $\widehat{z}(\boldsymbol{x}^{\text{new}})$. To minimize the global uncertainty over $\widehat{z}$ at a reduced computational cost, a natural approach would consist in searching the value of $\boldsymbol{x}^{\text{new}}$ such that

$$\int_{\mathbb{X}} \{k(\boldsymbol{x}, \boldsymbol{x}) - k(\boldsymbol{x}, \boldsymbol{x}^{\text{new}})^2 / k(\boldsymbol{x}^{\text{new}}, \boldsymbol{x}^{\text{new}})\} d\boldsymbol{x} \tag{2.11}$$

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

In the nested case, we also have to choose on which code to add a new observation point. To this end, let $\tau_1$ and $\tau_2$ 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 Gaussian process predictor sequentially. To simplify the reading, the following notation is proposed:

$$(\widetilde{\boldsymbol{x}}_i, \widetilde{\mathbb{X}}_i) := \begin{cases} (\boldsymbol{x}_1, \mathbb{X}_1) \text{ if } i = 1, \\ ((\varphi_1, \boldsymbol{x}_2), \mathbb{R} \times \mathbb{X}_2) \text{ if } i = 2, \\ ((\boldsymbol{x}_1, \boldsymbol{x}_2), \mathbb{X}_1 \times \mathbb{X}_2) \text{ if } i = 3, \end{cases} \tag{2.12}$$

and we denote by $\mathbb{V}(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}}) | \widetilde{\boldsymbol{x}}_i)$ the variance of $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})$ under the hypothesis that the code(s) corresponding to the new point $\widetilde{\boldsymbol{x}}_i$ is(are) evaluated in this point (in practice, we remind that these code evaluations are not required for the estimation of this variance).

- 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:

$$\widetilde{\boldsymbol{x}}_3^{\text{new}} = \underset{\widetilde{\boldsymbol{x}}_3 \in \widetilde{\mathbb{X}}_3}{\operatorname{argmin}} \int_{\mathbb{X}_{\text{nest}}} \mathbb{V}(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}}) | \widetilde{\boldsymbol{x}}_3) d\boldsymbol{x}_{\text{nest}}. \tag{2.13}$$

  Such a criterion is *a priori* adapted to the case when 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 predictor variance of the nested code:

$$(i^{\text{new}}, \widetilde{\boldsymbol{x}}_{i^{\text{new}}}^{\text{new}}) = \underset{\widetilde{\boldsymbol{x}}_i \in \widetilde{\mathbb{X}}_i,\ i \in \{1,2\}}{\operatorname{argmax}} \frac{1}{\tau_i} \times \int_{\mathbb{X}_{\text{nest}}} \left[ \mathbb{V}\left(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})\right) - \mathbb{V}\left(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}}) | \widetilde{\boldsymbol{x}}_i\right) \right] d\boldsymbol{x}_{\text{nest}}. \tag{2.14}$$

  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.

## 3. Fast evaluation of the prediction variance

As explained in Section 2.5, to choose the position of the new point, for each potential value of $\widetilde{\boldsymbol{x}}_i$ in $\widetilde{\mathbb{X}}_i$, we need to compute the value of $\operatorname{Var}(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}}) | \widetilde{\boldsymbol{x}}_i)$ for all $\boldsymbol{x}_{\text{nest}}$ in $\mathbb{X}_{\text{nest}}$. If quadrature rules or Monte Carlo approaches are used to evaluate this variance, as it is proposed in Section 2.3, the optimization procedure quickly becomes extremely demanding, even if discretized approximations of the optimization problem defined by Eqs. (2.14) and (2.13) are considered, that is to say where the integral over $\mathbb{X}_{\text{nest}}$ is replaced by an empirical mean over any $N_{\text{nest}}$-dimensional set of randomly chosen points of $\mathbb{X}_{\text{nest}}$. To circumvent this problem, we present in this section several approaches to make the evaluation of $\operatorname{Var}(\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}}) | \widetilde{\boldsymbol{x}}_i)$ explicit, and therefore extremely fast to evaluate.

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

**Proposition 3.1.** *Using the notations of the Universal Kriging framework that is introduced in Section 2.4, and denoting by $g$ the family of functions such that $g(x, \boldsymbol{\alpha}) := x^{(\boldsymbol{\alpha})_1} \exp\left[(\boldsymbol{\alpha})_2 x + (\boldsymbol{\alpha})_3 x^2\right]$, $\boldsymbol{\alpha} \in \mathbb{N} \times \mathbb{R}^2$ if:*

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

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

   *where $m_k$ is a deterministic function from $\mathbb{X}_2$ to $\mathbb{R}$ and $\boldsymbol{\alpha}_k \in \mathbb{N} \times \mathbb{R}^2$ is such that $2(\boldsymbol{\alpha}_k)_3 C_1^c(\boldsymbol{x}_1, \boldsymbol{x}_1) < 1$ for all $\boldsymbol{x}_1 \in \mathbb{X}_1$,*

2. *the covariance function $C_2$ is an element of the Gaussian class or corresponds to the covariance function of any derivative of a zero-mean process with covariance function of the Gaussian class,*

*then the conditional moments of order 1 and 2 of $\widehat{y}^c_{nest}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, which are defined by Eqs. (2.7) and (2.8) can be calculated analytically.*

In other words, if the prior of the Gaussian process modeling the function $y_2$ can be seen as any derivative of a Gaussian process with a trend which is a linear combination of products of polynomials by exponentials of order less than 2, and a covariance function of the Gaussian class, then conditionally to some integration criteria, the moments of order 1 and 2 of the coupling of the predictors of the two codes can be computed explicitly at a reduced cost. However, the approach cannot be generalized to the coupling of more than two codes.

### *3.2. Linearized approach*

In the cases where the conditions for Proposition 3.1 are not fulfilled (or if more than two codes were 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 $\varepsilon^c_i$ be the Gaussian process such that:

$$\widehat{y}^c_i = \mu^c_i + \varepsilon^c_i. \tag{3.2}$$

By construction, $\varepsilon^c_i$ is the residual prediction uncertainty once $\widehat{y}_i$ has been conditioned by $N_i$ evaluations of $y_i$. We remind that these two Gaussian processes are statistically independent. Under the condition that $N_1$ is not too small compared to the complexity of $y_1$, it is therefore reasonable to assume that $\varepsilon^c_1$ is small compared to $\mu^c_1$.

**Proposition 3.2.** *If:*

1. *the predictor of two nested computer codes can be written $\widehat{y}^c_{nest}(\boldsymbol{x}_1, \boldsymbol{x}_2) := \widehat{y}^c_2(\widehat{y}^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2)$, where $\widehat{y}^c_i$ are Gaussian processes which can be written as $\widehat{y}^c_i = \mu^c_i + \varepsilon^c_i$ where $\varepsilon^c_i \sim GP(0, C^c_i), \quad i \in \{1, 2\}$,*

2. *and $\varepsilon^c_1$ is small enough for the linearization to be valid,*

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

$$\begin{aligned} \mu^c_{nest} &= \mu^c_2(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2) \\ C^c_{nest}((\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{x}'_1, \boldsymbol{x}'_2)) &= C^c_2((\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2), (\mu^c_1(\boldsymbol{x}'_1), \boldsymbol{x}'_2)) \\ &\quad + \frac{\partial \mu^c_2}{\partial \varphi_1}(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2) \frac{\partial \mu^c_2}{\partial \varphi_1}(\mu^c_1(\boldsymbol{x}'_1), \boldsymbol{x}'_2) C^c_1(\boldsymbol{x}_1, \boldsymbol{x}'_1). \end{aligned} \tag{3.3}$$

Hence, thanks to the proposed linearization, the variance of $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})$ but also the one of $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_{\text{nest}})|\widetilde{\boldsymbol{x}}_i$ can explicitly be derived for all $(\boldsymbol{x}_{\text{nest}}, \widetilde{\boldsymbol{x}}_i)$ in $\mathbb{X}_{\text{nest}} \times \widetilde{\mathbb{X}}_i$. 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 (3.3) that the variance depends on $\boldsymbol{y}_1^{\text{obs}}$ through $\mu_1^c$. To circumvent this problem for the evaluation of the forward variance in the sequential designs, we assume that a candidate $\boldsymbol{x}_1$ is associated with the current estimate of the output of the first code $\mu_1^c(\boldsymbol{x}_1)$, in accordance with the Kriging Believer strategy proposed in [10].

## 4. Applications

The previously proposed methods are applied to two examples: an analytical one-dimensional one and a multidimensional one.

### *4.1. Characteristics of the examples*

#### *4.1.1. Analytical example*

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

$$\boldsymbol{h}_1(x_1) = \begin{bmatrix} 1 \\ x_1 \\ x_1^2 \end{bmatrix}, \quad \boldsymbol{\beta}_1^* = \begin{bmatrix} -2 \\ 0.25 \\ 0.0625 \end{bmatrix}, \quad y_1(x_1) = \boldsymbol{h}_1(x_1)^T \boldsymbol{\beta}_1^* - 0.25\cos(2\pi x_1), \tag{4.1}$$

$$\boldsymbol{h}_2(\varphi_1) = \begin{bmatrix} 1 \\ \varphi_1 \\ \varphi_1^2 \\ \varphi_1^3 \end{bmatrix}, \qquad \boldsymbol{\beta}_2^* = \begin{bmatrix} 6 \\ -5 \\ -2 \\ 1 \end{bmatrix}, \qquad y_2(\varphi_1) = \boldsymbol{h}_2(\varphi_1)^T \boldsymbol{\beta}_2^* - 0.25\cos(2\pi\varphi_1), \tag{4.2}$$

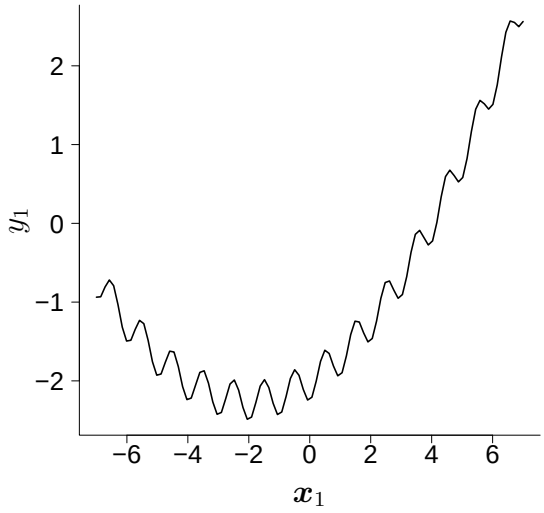
where $x_1 \in [-7, 7]$. In this example $\mathbb{X}_2 = \emptyset$.

Figure 1 shows the variations of the outputs of the codes 1, 2 and nested. The codes 1 and 2 outputs are relatively smooth compared with the one of the nested code. The amplitude of the variations is strongly non-stationary for the nested code.

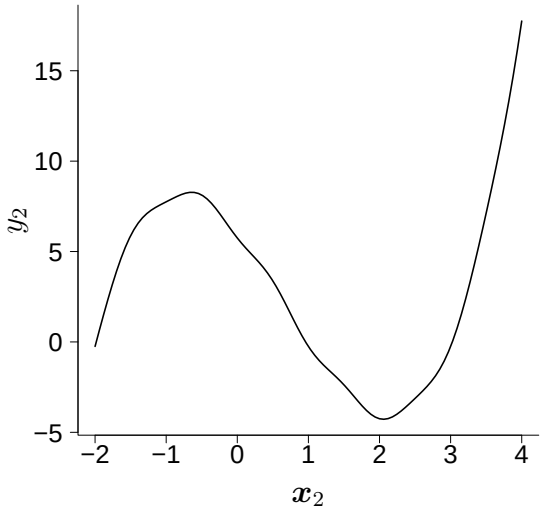
#### *4.1.2. Hydrodynamic example*

This example consists in the coupling of two computer codes. The objective is to determine the impact point of a conical projectile.

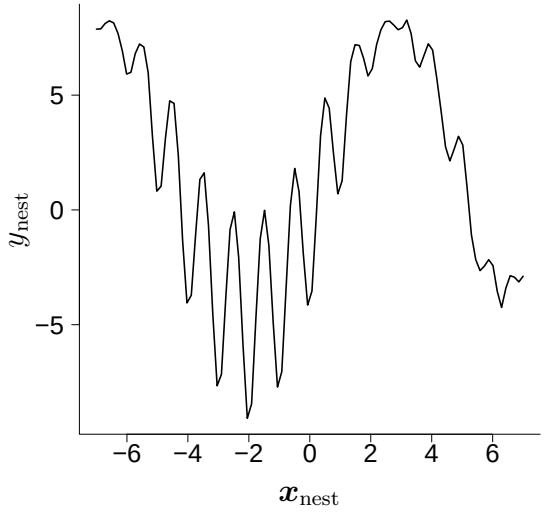
The first code computes the drag coefficient of a cone divided by the height of the cone. Its inputs are the height and the half-angle of the cone, so the dimension of $\boldsymbol{x}_1$ is 2.




(a) Code 1




(b) Code 2




(c) Nested code

Figure 1: Analytical example: variations of the outputs $y_1$, $y_2$ and $y_{\text{nest}}$ of the codes 1, 2 and nested with respect to their input.

The second code computes the range of the ballistic trajectory of a cone. Its inputs are the output of the first code, associated with $\varphi_1$, 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.

Figure 2 illustrates the two codes inputs and outputs.
Figure 3 shows the variations of the output with respect to each component of the input for each code. This figure enables 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:

$$\boldsymbol{h}_1(\boldsymbol{x}_1) = \left(1\ ,\ (\boldsymbol{x}_1)_1\ ,\ \frac{1}{(\boldsymbol{x}_1)_2}\right)^T. \tag{4.3}$$

For the second code only a multiplicative inverse variation with respect to $y_1$ is evident, so the proposed basis functions are:

$$\boldsymbol{h}_2(\varphi_1, \boldsymbol{x}_2) = \left(1\ ,\ \frac{1}{\max(\varphi_1, \varphi_{1_{\min}})}\right)^T. \tag{4.4}$$

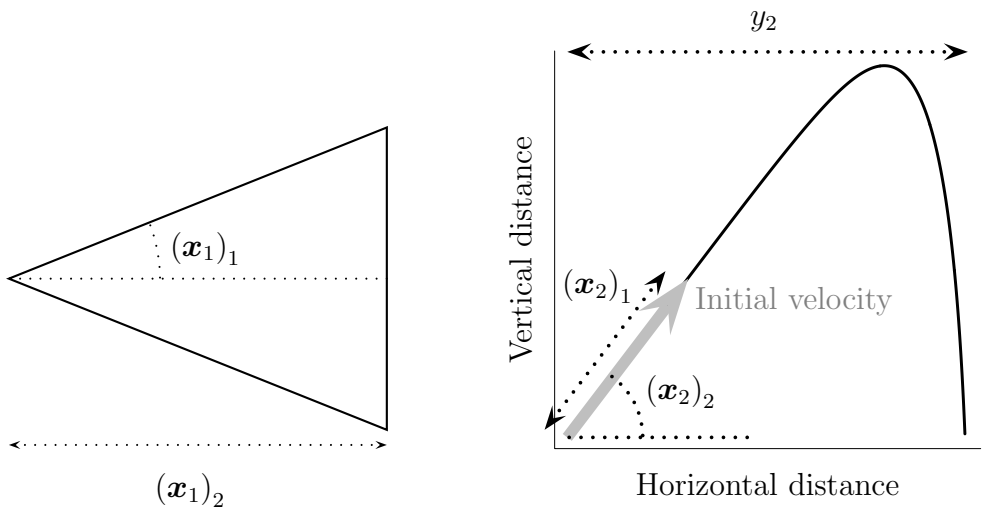
The denominator has a lower boundary $\varphi_{1_{\min}}$ in order to avoid any inversion problem around zero. This boundary is small and set arbitrarily.

*4.2. Reference: "blind box" method*

In this method, the nested computer code is considered as a single computer code. Only the inputs $\boldsymbol{x}_{\text{nest}}$ and the output $y_{\text{nest}}$ are taken into account. The intermediary information $\varphi_1$ is not considered. A Gaussian process regression of this single computer code is done.
Only the chained I-optimal sequential design could be applied in this framework, the other proposed sequential design requiring to consider the partial information.

*4.3. Choice of the covariance functions and estimation of their hyperparameters*

In the analytical example the covariance functions are Gaussian. This implies that the sample paths of the Gaussian processes associated with the codes are infinitely differentiable functions. This enables to apply Proposition 3.1 and Proposition 3.2 to this example.
In the hydrodynamic example the covariance functions are Matérn $\frac{5}{2}$, which implies that the sample paths of the Gaussian processes associated with the codes are mean square one time continuously differentiable functions (see [22]). This enables to perform the linearization of Proposition 3.2.
In both cases the covariance functions include a non-zero nugget term (see [12] for further details).




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

(b) Code 2: range of a ballistic trajectory

Figure 2: Hydrodynamic example: Inputs and outputs of the two codes.

The hyperparameters of the covariance functions are estimated for each set of observations, including the sequential designs. They are estimated by maximizing the Leave-One-Out log predictive probability (see [22], chapter 5, and [1]).

*4.4. Comparison between the analytical and the linearized method*

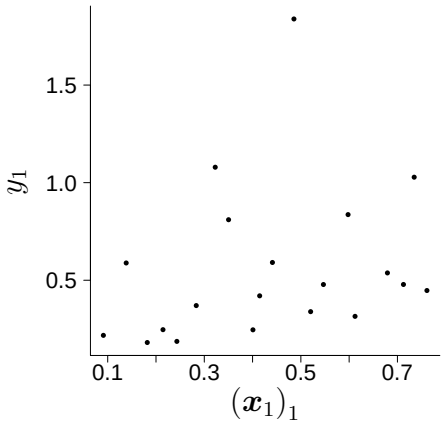
Figure 4 illustrates the convergence of the two first statistical moments estimated with the Monte Carlo (see Proposition 2.1) and the linearized methods (see Proposition 3.2) towards their real values calculated with the analytical method described in Proposition 3.1.

Both methods converge when the uncertainty of the first code predictor decreases. It can be seen that the linearized method is a very good compromise between computation time and accuracy compared to the Monte Carlo method.

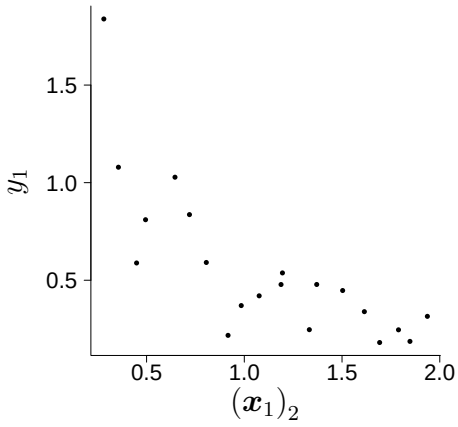
*4.5. Definition of the performance criterion of the predictor mean*

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

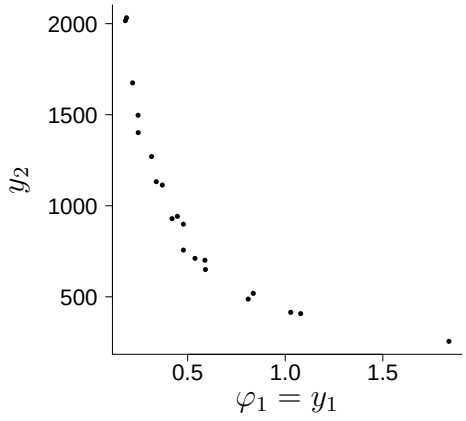
Denoting by $y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(1)}\right) \dots y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(N_{\text{nest}})}\right)$ the evaluations of the nested code in these points, the performance criterion of the nested predictor mean, also called error on the mean can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



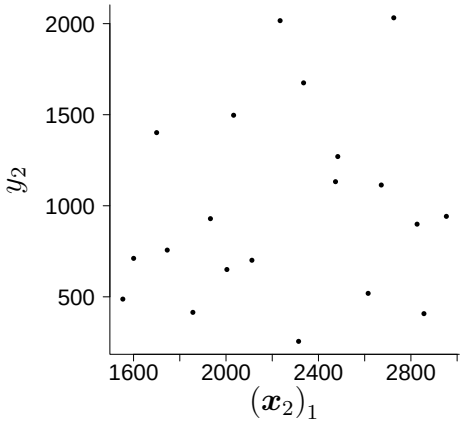

(a) Code 1



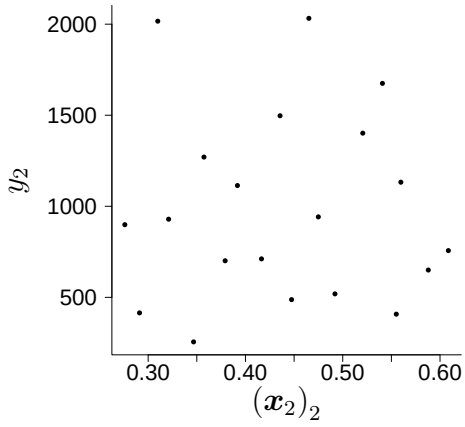

(b) Code 1




(c) Code 2

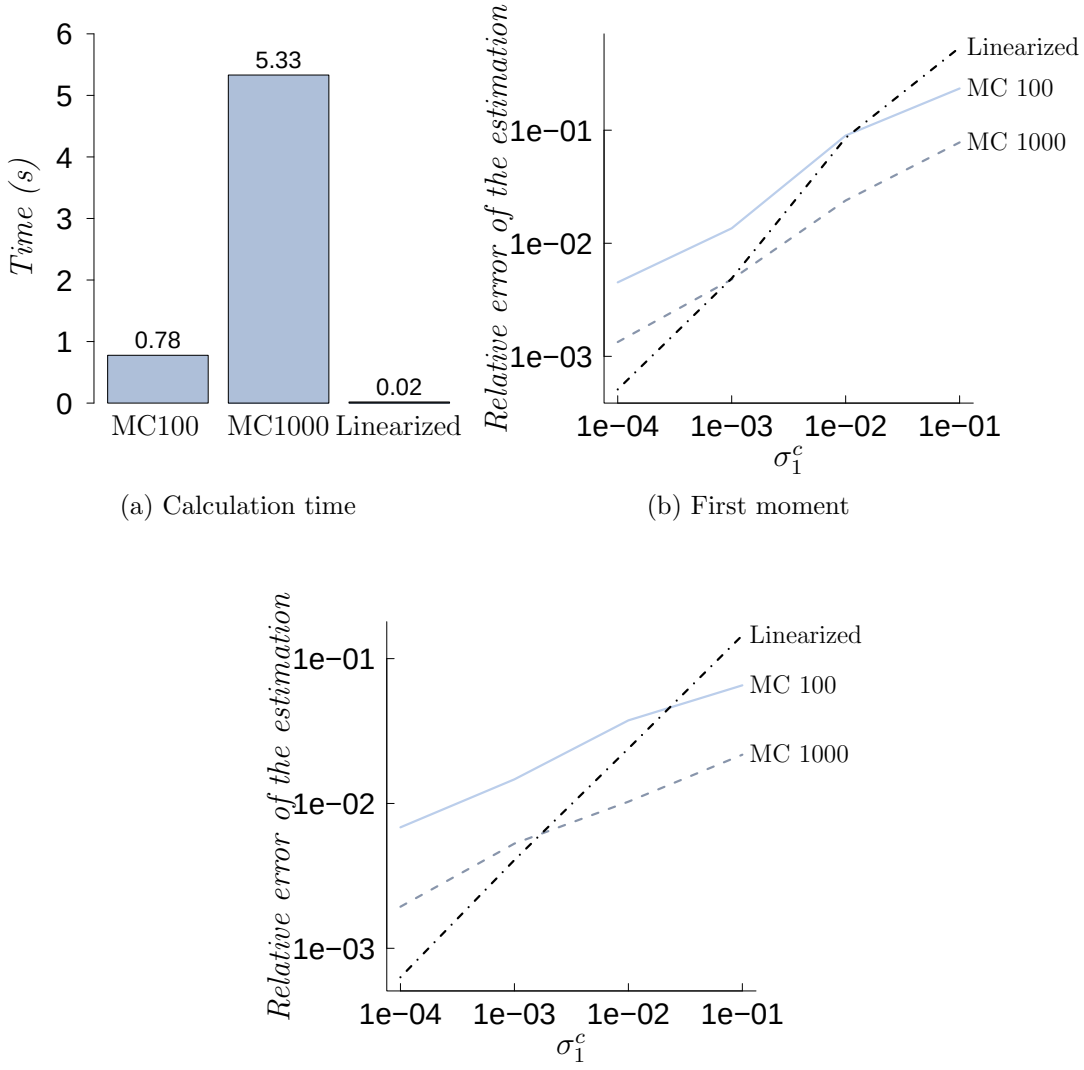



(d) Code 2




(e) Code 2

Figure 3: Hydrodynamic example: variation of the outputs $y_1$ and $y_2$ of the two codes with respect to the components of the inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The 20 input points are drawn according to a maximin LHS design on $\mathbb{X}_1 \times \mathbb{X}_2$.

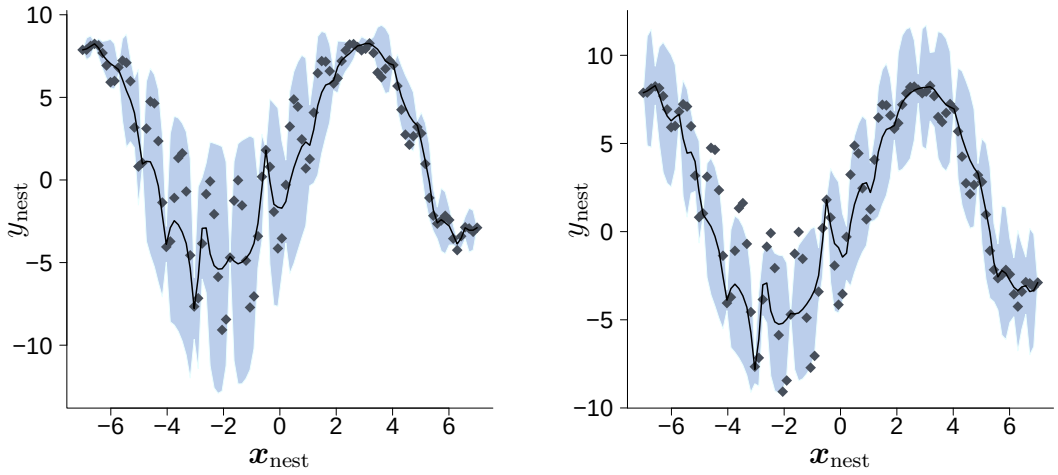



(a) Calculation time

(b) First moment

(c) Second moment

Figure 4: Comparison of the linearized (Proposition 3.2) and Monte-Carlo (Proposition 2.1) methods in terms of computation time and accuracy for the evaluation of the two first moments of the process $\widehat{y}^c_{\text{nest}}$. The Monte Carlo method is run with 100 and 1000 points to compute the one-dimensional integral with a Gaussian measure. The Monte Carlo draws are repeated 50 times and the curves correspond to the median of these repetitions.
The real values are computed with the analytical method (Proposition 3.1). The covariance functions are Gaussian. The predictor of the first code is of the form $y_1^c = \mu_1^c + \sigma_1^c u$ with $u \sim \mathcal{N}(0,1)$, $\sigma_1^c \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and for each value of $\sigma_1^c$, 100 values of $\mu_1^c$ on a grid on $[-2, 4]$ are considered. The predictor of the second code is build using 20 input observation points drawn on a grid on $[-2, 4]$ for the second code of the analytical example.




(a) Linearized method

(b) Blind box method

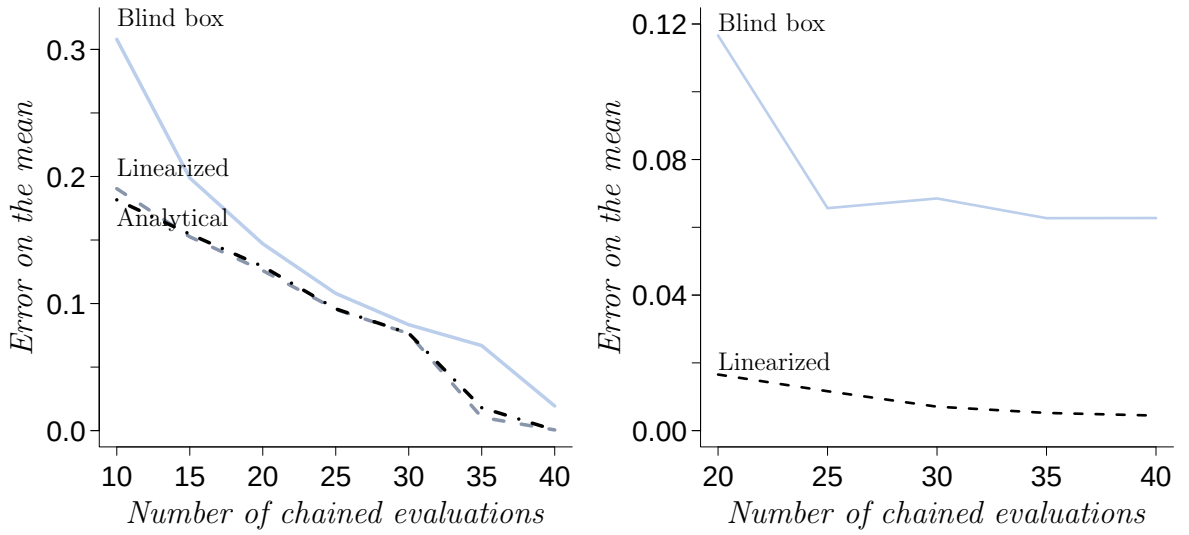
Figure 5: Analytical example: Predictors of the nested code obtained with the linearized and the blind box methods. The set of 20 observations is drawn according to a maximin LHS on $\mathbb{X}_1$. Actual values shown by dots, the mean of prediction by a line and the 95% prediction interval of prediction by a grey area.

$$\text{Error on the mean} = \frac{\sum_{i=1}^{N_{\text{nest}}} \left( y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right) - \mathbb{E}\left[\widehat{y}_{\text{nest}}^{c}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right)\right]\right)^2}{\sum_{i=1}^{N_{\text{nest}}} \left( y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(i)}\right) - \frac{1}{N_{\text{nest}}} \sum_{j=1}^{N_{\text{nest}}} y_{\text{nest}}\left(\boldsymbol{x}_{\text{nest}}^{(j)}\right)\right)^2}. \tag{4.5}$$

*4.6. Comparison between the blind box and the linearized methods*

Figure 5 shows that the linearized method enables to better take into account the non-stationarity of the variations of the nested code output. On the contrary, 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.

Figure 6 shows the similar accuracies of the prediction mean computed with the analytical and linearized methods proposed in Proposition 3.1 and Proposition 3.2.
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.




(a) Analytical: Gaussian covariance

(b) Hydrodynamic example: Matérn $\frac{5}{2}$ covariance

Figure 6: Comparison of the prediction mean accuracy for the blind box and the linearized (Proposition 3.2) methods, and, in case of a Gaussian covariance function, the analytical method (Proposition 3.1). The curves correspond to the median of 50 draws of maximin LHS designs on $\mathbb{X}_1 \times \mathbb{X}_2$ of increasing size.

*4.7. Performances of the sequential designs with identical computational costs*

Figure 7 shows the relevance of the proposed sequential designs for improving the prediction mean of the linearized nested predictor, compared to the maximin LHS design 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, the different sequential designs give similar results, except for the first new observation points added, where the best I-optimal is better.

In both examples the new observation points are mostly added on the first code, as shown in figure 8. 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 to reduce this uncertainty by first adding new observation points on the first code. Then new observations points can be added on both codes.

*4.8. Performances of the sequential designs with different computational costs*

Figure 9 shows the prediction mean accuracy with a best I-optimal sequential design when the costs of the two codes are different. It can be seen that at a given total computational cost the accuracy of prediction is better when the cost of the first code is lower. In other words the prediction mean accuracy is better at a given computational budget when more observation points can be added to the first code for the same computational budget. These results are consistent with those of figure 8.

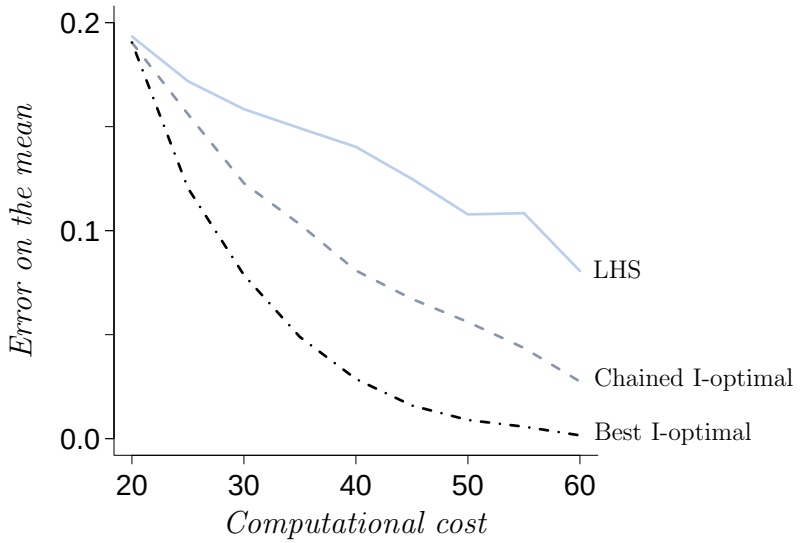
## 5. Conclusions and future work

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

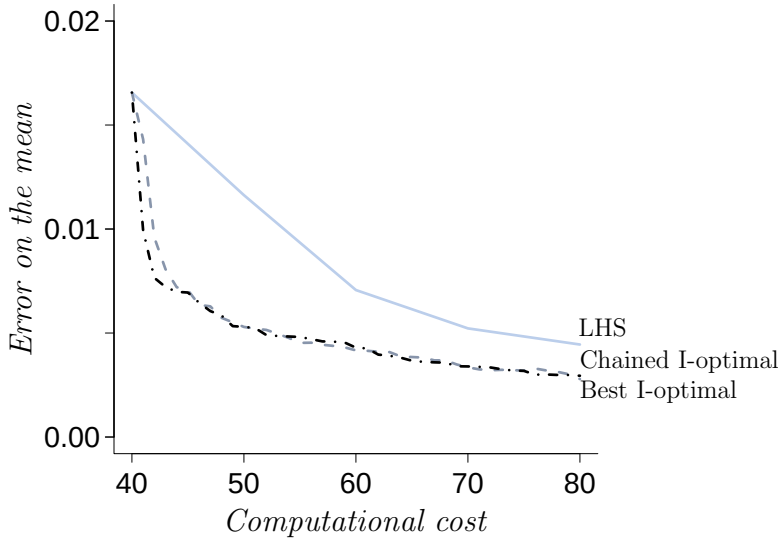
Two methods to evaluate quickly the mean and variance of the nested code predictor have been proposed. The first one, called "analytical" computes the exact value of the two first moments of the predictor. But it cannot be applied to the coupling of more than two codes. The second one, called "linearized", enables to obtain a Gaussian predictor of the two nested codes, 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 to 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 when the two codes cannot be launched separately. The second one, the "best" I-optimal sequential design, allows to choose on which of the two codes

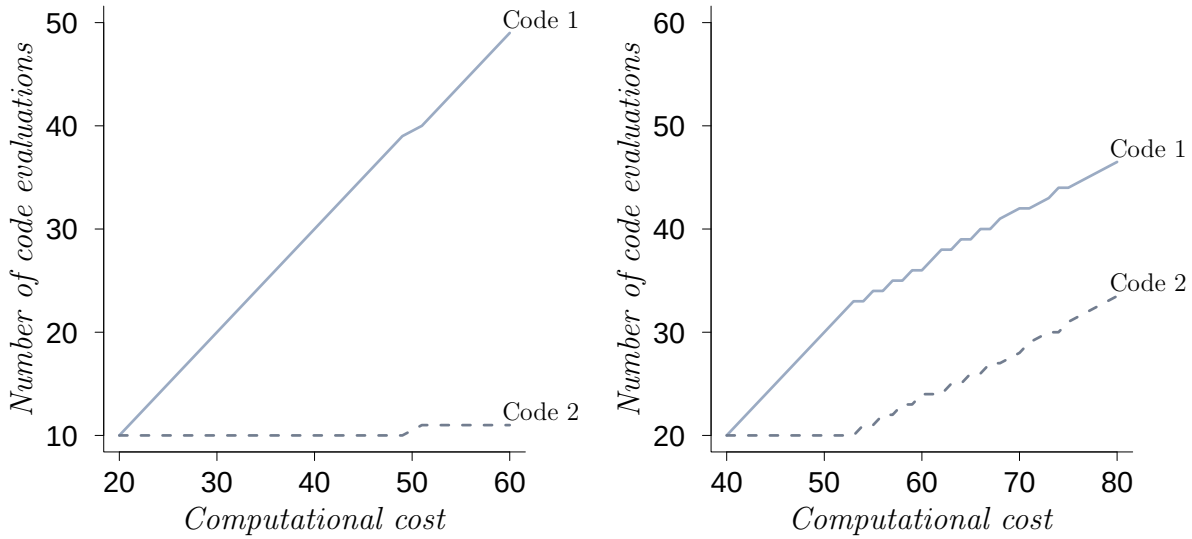



(a) Analytical example




(b) Hydrodynamic example

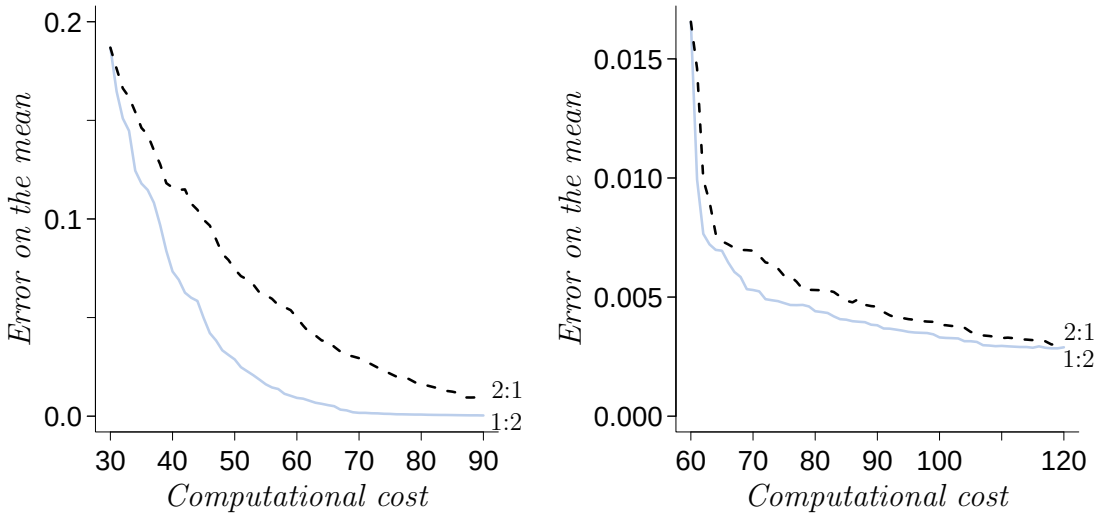
Figure 7: Comparison of the linearized predictor mean precision with the maximin LHS design on $\mathbb{X}_{\text{nest}}$ and the sequential designs applied to the two examples. In the hydrodynamic example, the two curves representing the sequential designs are almost superimposed. The initial designs 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 chained maximin LHS designs 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.




(a) Analytical example

(b) Hydrodynamic example

Figure 8: Comparison of the number of evaluations of each code in 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 9: Performances of the best I-optimal sequential design in terms of prediction mean accuracy with different computational costs for the two codes. 1:2 $\leftrightarrow$ cost 1 for code 1 and 2 for code 2, 2:1 $\leftrightarrow$ cost 2 for code 1 and 1 for code 2. 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.

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 to a space-filling design (maximin LHS). Furthermore, they illustrate the advantage, in terms of prediction mean accuracy, of choosing on which code to add a new observation point compared to simply adding new observation points of the nested code. The results obtained show an amplification of the uncertainties in the chain of codes, leading to the addition of observation points on 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

*Proof of Proposition 2.1*

According to Eq (2.5):

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

where $\xi_1$ and $\xi_2$ are independent according to the independence of the initial processes $\widehat{y}_1$ and $\widehat{y}_2$.
Therefore the process modeling the nested code can be written:

$$\begin{aligned}\widehat{y}_{\text{nest}}^c(\boldsymbol{x}_1,\boldsymbol{x}_2) = &\ \widehat{y}_2^c(\widehat{y}_1^c(\boldsymbol{x}_1),\boldsymbol{x}_2)\\ = &\ \mu_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right)+\sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1,\boldsymbol{x_2}\right)+\sigma_2^c\left(\mu_1^c\left(\boldsymbol{x_1}\right)+\sigma_1^c\left(\boldsymbol{x_1}\right)\xi_1,\boldsymbol{x_2}\right)\xi_2\end{aligned}$$

Given the independence of $\xi_1$ and $\xi_2$ and the fact that $\mathbb{E}\left(\xi_2\right)=0$, it can be inferred that the first moment of $\widehat{y}_{\text{nest}}^c$ can be written:

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

By noting that:

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

- $\xi_1$ and $\xi_2$ are independent,
- $\mathbb{E}\left(\xi_2\right)=0$ and $\mathbb{E}\left(\xi_2^2\right)=1$,

the second moment of $\widehat{y}_{\text{nest}}^c$ can be written:

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

*Proof of Proposition 3.1*

If $x \sim \mathcal{N}(\mu,\sigma^2)$ and $f\left(x,a,b,c\right) = x^c\exp\left(ax+bx^2\right)$ then the mean of $f\left(x,a,b,c\right)$ is defined as:

$$\mathbb{E}\left[f\left(x,a,b,c\right)\right] = \exp\left(-\frac{1}{2\sigma^2}\left(\frac{\left(\sigma^2a+\mu\right)^2}{2\sigma^2b-1}+\mu^2\right)\right)\mathbb{E}\left[x_f^c\right]$$

where $x_f \sim \mathcal{N}\left(\frac{\sigma^2 a + \mu}{1 - 2b\sigma^2}, \frac{\sigma^2}{1 - 2b\sigma^2}\right)$, under the condition that $1 - 2b\sigma^2 > 0$.
Given that the moments of a Gaussian variable can be calculated analytically, $\mathbb{E}\left[x_g^c\right]$ and therefore $\mathbb{E}\left[f\left(x, a, b, c\right)\right]$ can be computed analytically.
So we have shown that if $x \sim \mathcal{N}(\mu, \sigma^2)$, and $f\left(x, a, b, c\right) = x^c \exp\left(ax + bx^2\right)$ then, under the integrability condition $1 - 2b\sigma^2 > 0$, the mean of $f\left(x, a, b, c\right)$ can be calculated analytically.

*First moment*
In the framework of Universal Kriging, the conditional mean function of the process modeling the second code can be written:

$$\begin{aligned}
\mu_2^c\left(\left(\varphi_1, \boldsymbol{x}_2\right)\right) = & \quad \boldsymbol{h_2}\left(\left(\varphi_1, \boldsymbol{x}_2\right)\right)^T \boldsymbol{v}_h + C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\boldsymbol{\varphi}_1^{\text{obs}}, \boldsymbol{X}_2^{\text{obs}}\right)\right) \boldsymbol{v}_c \\
= & \quad \sum_{i=1}^{M_2} \left(\boldsymbol{h_2}\left(\left(\varphi_1, \boldsymbol{x}_2\right)\right)\right)_i \left(\boldsymbol{v}_h\right)_i + \sum_{i=1}^{N_1} C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{(i)}, \boldsymbol{x}_2^{(i)}\right)\right) \left(\boldsymbol{v}_c\right)_i \\
= & \quad (1) + (2)
\end{aligned}$$

where $\boldsymbol{v}_h \in \mathbb{R}^{M_2}$ and $\boldsymbol{v}_c \in \mathbb{R}^{N_1}$ and $\varphi_1 \sim \mathcal{N}\left(\mu_1^c, \left(\sigma_1^c\right)^2\right)$.

According to the assumptions of Proposition 3.1 the mean basis functions $\boldsymbol{h}_2$ can be written:

$$\left(\boldsymbol{h}_2((\varphi_1, \boldsymbol{x}_2))_i = m_i(\boldsymbol{x}_2)\ f\left(\varphi_1, \left(\boldsymbol{\alpha}_i\right)_3, \left(\boldsymbol{\alpha}_i\right)_1, \left(\boldsymbol{\alpha}_i\right)_2\right),\right.$$

with $m_i$ deterministic functions.
In the same way, the covariance function $C_2$ can be written:

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

with $k : x \mapsto \exp\left(-x^2/2\right)$, $n_{\varphi_1}$ and $n_i$ positive integers and $k^{(n)}$ denoting the n-th derivative of function $k$. So, we can written that:

$$C_2\left(\left(\varphi_1, \boldsymbol{x}_2\right), \left(\varphi_1^{'}, \boldsymbol{x}_2^{'}\right)\right) = \sigma_2^2 \sum_{j=1}^{n_{\varphi_1}} a_j f\left(\varphi_1 - \varphi_1', 0, \frac{-1}{2l_1^2}, 2j\right) l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2'\right)\ ,$$

where $l$ is a deterministic function defined according to the previous equation and $a_j$ real numbers.

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

$$(1) = \quad \sum_{i=1}^{M_2} f\left(\varphi_1, \left(\boldsymbol{\alpha}_i\right)_3, \left(\boldsymbol{\alpha}_i\right)_1, \left(\boldsymbol{\alpha}_i\right)_2\right)\ m_i(\boldsymbol{x}_2)\left(\boldsymbol{v}_h\right)_i$$

$$(2) = \sum_{i=1}^{N_1} \sigma_2^2 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right)(\boldsymbol{v}_c)_i \sum_{j=1}^{n_{\varphi_1}} a_j f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2j\right)$$

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

$$\mathbb{E}\left[(1)\right] = \sum_{i=1}^{M_2} \mathbb{E}\left[f\left(\varphi_1, (\boldsymbol{\alpha}_i)_3, (\boldsymbol{\alpha}_i)_1, (\boldsymbol{\alpha}_i)_2\right)\right] m_i(\boldsymbol{x}_2)(\boldsymbol{v}_h)_i$$

$$\mathbb{E}\left[(2)\right] = \sum_{i=1}^{N_1} \sigma_2^2 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right)(\boldsymbol{v}_c)_i \sum_{j=1}^{n_{\varphi_1}} a_j \mathbb{E}\left[f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2j\right)\right]$$

The means $\mathbb{E}\left[(1)\right]$ and $\mathbb{E}\left[(2)\right]$ can therefore 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.

*Second moment*

In the framework of Universal Kriging, it can be written that:

$$\begin{aligned}
\left(\mu_2^c\left((\varphi_1, \boldsymbol{x}_2)\right)\right)^2 + \left(\sigma_2^c\left((\varphi_1, \boldsymbol{x}_2)\right)\right)^2 &= \sigma_2^2 \\
&+ \underbrace{\boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)^T \boldsymbol{A}_h \boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)}_{(1)} \\
+ \underbrace{C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\boldsymbol{\varphi}_1^{\text{obs}}, \boldsymbol{X}_2^{\text{obs}}\right)\right) \boldsymbol{A}_c \, C_2\left(\left(\boldsymbol{\varphi}_1^{\text{obs}}, \boldsymbol{X}_2^{\text{obs}}\right), (\varphi_1, \boldsymbol{x}_2)\right)}_{(2)} \\
+ \underbrace{C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\boldsymbol{\varphi}_1^{\text{obs}}, \boldsymbol{X}_2^{\text{obs}}\right)\right) \boldsymbol{A}_{ch} \, \boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)}_{(3)},
\end{aligned}$$

where $\boldsymbol{A}_h$, $\boldsymbol{A}_c$ and $\boldsymbol{A}_{ch}$ are deterministic real-valued, $M_2 \times M_2$, $N_1 \times N_1$ and $N_1 \times M_2$ dimensional matrices.

According to the assumptions of Proposition 3.1 and the previous equations, the terms (1), (2) and (3) can be rewritten:

$$\begin{aligned}
(1) &= \sum_{i=1}^{M_2}\sum_{j=1}^{M_2} (\boldsymbol{A}_h)_{ij} \left(\boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)\right)_i \left(\boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)\right)_j \\
&= \sum_{i=1}^{M_2}\sum_{j=1}^{M_2} (\boldsymbol{A}_h)_{ij} \, m_i(\boldsymbol{x}_2) \, m_j(\boldsymbol{x}_2) \, f\left(\varphi_1, (\boldsymbol{\alpha}_i)_3, (\boldsymbol{\alpha}_i)_1, (\boldsymbol{\alpha}_i)_2\right) f\left(\varphi_1, (\boldsymbol{\alpha}_j)_3, (\boldsymbol{\alpha}_j)_1, (\boldsymbol{\alpha}_j)_2\right), \\
&= \sum_{i=1}^{M_2}\sum_{j=1}^{M_2} (\boldsymbol{A}_h)_{ij} \, m_i(\boldsymbol{x}_2) \, m_j(\boldsymbol{x}_2) \, f\left(\varphi_1, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_3, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_1, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_2\right),
\end{aligned}$$

$$
\begin{aligned}
(2) = & \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} (\boldsymbol{A}_c)_{ij}\, C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1^{(i)}, \boldsymbol{x}_2^{(i)}\right)\right) C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1^{(j)}, \boldsymbol{x}_2^{(j)}\right)\right) \\
= & \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} (\boldsymbol{A}_c)_{ij}\, \sigma_2^4 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(j)}\right) \\
& \sum_{n=1}^{n_{\varphi_1}} a_n f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2n\right) \sum_{m=1}^{n_{\varphi_1}} a_m f\left(\varphi_1 - \varphi_1^{(j)}, 0, \frac{-1}{2l_1^2}, 2m\right) \\
= & \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} (\boldsymbol{A}_c)_{ij}\, \sigma_2^4 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(j)}\right) \\
& \sum_{n=1}^{n_{\varphi_1}}\sum_{m=1}^{n_{\varphi_1}} a_n a_m f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2n\right) f\left(\varphi_1 - \varphi_1^{(j)}, 0, \frac{-1}{2l_1^2}, 2m\right),
\end{aligned}
$$

$$
\begin{aligned}
(3) = & \sum_{i=1}^{N_1}\sum_{j=1}^{M_2} (\boldsymbol{A}_{ch})_{ij}\, C_2\left((\varphi_1, \boldsymbol{x}_2), \left(\varphi_1^{(i)}, \boldsymbol{x}_2^{(i)}\right)\right) \left(\boldsymbol{h_2}\left((\varphi_1, \boldsymbol{x}_2)\right)\right)_j \\
= & \sum_{i=1}^{N_1}\sum_{j=1}^{M_2} (\boldsymbol{A}_{ch})_{ij}\, \sigma_2^2 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) m_j\left(\boldsymbol{x}_2\right) \\
& \sum_{n=1}^{n_{\varphi_1}} a_n f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2n\right) f\left(\varphi_1, (\boldsymbol{\alpha}_j)_3, (\boldsymbol{\alpha}_j)_1, (\boldsymbol{\alpha}_j)_2\right).
\end{aligned}
$$

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 $a_i$ deterministic real numbers, it can be written:

$$
\mathbb{E}\left[(1)\right] = \sum_{i=1}^{M_2}\sum_{j=1}^{M_2} (\boldsymbol{A}_h)_{ij}\, m_i\left(\boldsymbol{x}_2\right) m_j\left(\boldsymbol{x}_2\right) \mathbb{E}\left[f\left(\varphi_1, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_3, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_1, (\boldsymbol{\alpha}_i + \boldsymbol{\alpha}_j)_2\right)\right],
$$

$$
\begin{aligned}
\mathbb{E}\left[(2)\right] = & \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} (\boldsymbol{A}_c)_{ij}\, \sigma_2^4 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(j)}\right) \\
& \sum_{n=1}^{n_{\varphi_1}}\sum_{m=1}^{n_{\varphi_1}} a_n a_m \mathbb{E}\left[f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2n\right) f\left(\varphi_1 - \varphi_1^{(j)}, 0, \frac{-1}{2l_1^2}, 2m\right)\right],
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}\left[(3)\right] = & \sum_{i=1}^{N_1}\sum_{j=1}^{M_2} (\boldsymbol{A}_{ch})_{ij}\, \sigma_2^2 l\left(\boldsymbol{x}_2 - \boldsymbol{x}_2^{(i)}\right) m_j\left(\boldsymbol{x}_2\right) \\
& \sum_{n=1}^{n_{\varphi_1}} a_n \mathbb{E}\left[f\left(\varphi_1 - \varphi_1^{(i)}, 0, \frac{-1}{2l_1^2}, 2n\right) f\left(\varphi_1, (\boldsymbol{\alpha}_j)_3, (\boldsymbol{\alpha}_j)_1, (\boldsymbol{\alpha}_j)_2\right)\right].
\end{aligned}
$$

The means $\mathbb{E}\left[(1)\right]$, $\mathbb{E}\left[(2)\right]$ and $\mathbb{E}\left[(3)\right]$ can therefore be calculated analytically, and 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.

From the two previous paragraphs and Proposition 1, it can be inferred that if verifying the assumptions of Proposition 3.1, then the first and the second moments of $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ can be calculated analytically.

*Proof of Proposition 3.2*

If $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \widehat{y}^c_2(\widehat{y}^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2)$ where $\widehat{y}^c_i = \mu^c_i + \varepsilon^c_i$, $\varepsilon^c_i \sim \text{GP}\left(0, C^c_i\right)$, $i \in \{1, 2\}$, then if $\varepsilon^c_1$ is small enough, the process $\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ can be linearized:

$$\begin{aligned}\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \mu^c_2(\mu^c_1(\boldsymbol{x}_1) + \varepsilon^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2) + \varepsilon^c_2(\mu^c_1(\boldsymbol{x}_1) + \varepsilon^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2), \\ &\approx \mu^c_2(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2) + \frac{\partial \mu^c_2}{\partial \varphi_1}(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2)\varepsilon^c_1(\boldsymbol{x}_1) + \varepsilon^c_2(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2),\end{aligned}$$

$\varepsilon_1$ and $\varepsilon_2$ being Gaussian processes, the predictor of the nested code can therefore be written as a Gaussian process:

$$\widehat{y}^c_{\text{nest}}(\boldsymbol{x}_1, \boldsymbol{x}_2) \approx \mu^c_2(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2) + \varepsilon^c_{\text{nest}}(\mu^c_1(\boldsymbol{x}_1), \boldsymbol{x}_2),$$

where $\varepsilon^c_{\text{nest}}$ is a centred Gaussian process, whose covariance function, $C^c_{\text{nest}}$, is given by:

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

[1] F Bachoc. *Parametric estimation of covariance function in Gaussian-process based Kriging models. Application to uncertainty quantification for computer experiments*. PhD thesis, Université Paris-Diderot - Paris VII, 2013.

[2] C. T. H. Baker. *The numerical treatment of integral equations*. Clarendon Press, Oxford, 1977.

[3] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22:773–793, 2012.

[4] J. O. Berger, V. De Oliveira, and B. Sansó. Objective bayesian analysis of spatially corellated data. *Journal of the American Statistical Association*, 96(456):1361–1374, 2001.

[5] 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 Journal*, 46:2459–2468, 2008.

[6] 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(4):455–465, 2014.

[7] B. Echard, N. Gayton, and M. Lemaire. AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, 33:145–154, 2011.

[8] K.T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall, Computer Science and Data Analysis Series, London, 2006.

[9] K.T. Fang and D.K. Lin. Uniform experimental designs and their applications in industry. *Handbook of Statistics*, 22:131–178, 2003.

[10] D. Ginsbourger, R. Le Riche, and L. Carraro. *Computational Intelligence in Expensive Optimization Problems*, volume 2 of *Adaptation Learning and Optimization*, chapter Kriging Is Well-Suited to Parallelize Optimization, pages 131–162. Springer Berlin Heidelberg, 2010.

[11] R. Gramacy and H. Lian. Gaussian process single-index models as emulators for computer experiments. *Technometrics*, 54:1:30–41, 2012.

[12] R. B. Gramacy and H. K. H. Lee. Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22:713–722, 2012.

[13] R. Hu and M. Ludkovski. Sequential design for ranking response surfaces. *SIAM/ASA Journal on Uncertainty Quantification*, 5:212–239, 2017.

[14] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are avalaible. *Biometrika*, 87:1–13, 2000.

[15] M. C. Kennedy and A. O'Hagan. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology*, 63(3):425–464, 2001.

[16] Jack P.C. Kleijnen. Regression and kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research*, 256:1–16, 2017.

[17] Stein M.L. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer, New York, 1999.

[18] R. Paulo. Default priors for gaussian processes. *Annals of Statistics*, 33(2):556–582, 2005.

[19] G. Perrin. Active learning surrogate models for the conception of systems with multiple failure modes. *Reliability Engineering and System Safety*, 149:130–136, 2016.

[20] G. Perrin and C. Cannamela. A repulsion-based method for the definition and the enrichment of opotimized space filling designs in constrained input spaces. *Journal de la Société Française de Statistique*, 158(1):37–67, 2017.

[21] G. Perrin, C. Soize, S. Marque-Pucheu, and J. Garnier. Nested polynomial trends for the improvement of gaussian process-based predictors. *Journal of Computational Physics*, 346:389–402, 2017.

[22] C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, Cambridge, 2006.

[23] C. Robert. *The Bayesian Choice.* Springer-Verlag New York, New York, 2007.

[24] J. Sacks, W. Welch, T. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4:409–435, 1989.

[25] T J. Santner, B J. Williams, and W Notz. *The design and analysis of computer experiments.* Springer series in statistics. Springer, New York, 2003.