Nested polynomial trends for the improvement of Gaussian process-based predictors

G. Perrin [a,*], C. Soize [b], S. Marque-Pucheu [a,c], J. Garnier [c]

[a] *CEA/DAM/DIF, F-91297, Arpajon, France*
[b] *Université Paris-Est, MSME UMR 8208 CNRS, Marne-la-Vallée, France*
[c] *Laboratoire de Probabilités et Modèles Aléatoires, Laboratoire Jacques-Louis Lions, Université Paris Diderot, 75205 Paris Cedex 13, France*

A R T I C L E   I N F O

A B S T R A C T

The role of simulation keeps increasing for the sensitivity analysis and the uncertainty quantification of complex systems. Such numerical procedures are generally based on the processing of a huge amount of code evaluations. When the computational cost associated with one particular evaluation of the code is high, such direct approaches based on the computer code only, are not affordable. Surrogate models have therefore to be introduced to interpolate the information given by a fixed set of code evaluations to the whole input space. When confronted to deterministic mappings, the Gaussian process regression (GPR), or kriging, presents a good compromise between complexity, efficiency and error control. Such a method considers the quantity of interest of the system as a particular realization of a Gaussian stochastic process, whose mean and covariance functions have to be identified from the available code evaluations. In this context, this work proposes an innovative parametrization of this mean function, which is based on the composition of two polynomials. This approach is particularly relevant for the approximation of strongly non linear quantities of interest from very little information. After presenting the theoretical basis of this method, this work compares its efficiency to alternative approaches on a series of examples.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The numerical cost of many codes to simulate complex physical systems is still very high. To perform sensitivity analyses, uncertainty quantification or reliability studies, these computer models have therefore to be replaced by surrogate models, that is to say by fast and inexpensive mathematical functions. Within the computational science community, when the maximal available information is a finite set of code evaluations, the most widely used surrogate models are the generalized polynomial chaos expansion (PCE) [11,10,32,7,21,1,27] and the Gaussian process regression (GPR), or kriging (see [30,24,34]).

On the one hand, the main idea of PCE is to expand the code output, which is denoted by g in the following, onto an appropriate basis made of orthonormal multivariate polynomials, which are related to the distribution of the code input variables. As the number of unknown expansion coefficients usually grows exponentially with the number of input parameters, the relevance of these approaches strongly depends on their ability to select the most relevant basis functions. To this end, several penalization techniques, such as the ℓ_1-minimization [33,15] and the least Angle Regression (LAR) methods [14,9,5], have been introduced to select polynomial basis sets that lead to more accurate PCE than would have been

obtained if the basis is *a priori* fixed. Taking advantage of the tensor-product structure of the multivariate polynomial basis, separated representations, such as low-rank approximations [23,19], have alternatively been proposed to develop surrogate models with polynomial functions in highly-compressed formats.

On the other hand, the GPR is based on the assumption that the code output is a particular realization of a Gaussian stochastic process, Y. This hypothesis, which was first introduced in time series analysis [26] and in optimization [20], is widely used as it allows dealing with the conditional probability and expectation, while leading to very interesting results in terms of computer code prediction. Hence, contrary to the PCE, the GPR is not associated with an *a priori* projection basis, but requires the introduction of the mean and the covariance functions of Y. In practice, we observe that the role of the mean function of Y on the prediction decreases when the number of code evaluations increases. This explains that in applications where many code evaluations are available, good GPR-based surrogate models can be obtained using constant or linear trends for the mean function. On the contrary, when the number of code evaluations is low compared to the complexity of g, it can be very useful to optimize it. In that case, searching the mean function of Y as a well-chosen sum of polynomial functions can indeed strongly improve the relevance of the associated GPR. In particular, the authors refer to [16] and [18] for an illustration of the interest of using variable selection techniques to optimize this polynomial representation of the mean function of Y.

Following on these works, the idea of this paper is to propose an alternative parametrization of the mean function of Y, which is particularly adapted to the case when the number of code evaluations is small compared to the complexity of g. Instead of searching sparse polynomial approximations, we look for high dimensional polynomial approximations that are characterized by a small number of parameters. In other words, if we want to model a complex code response with a very limited number of code evaluations, we believe that it can be more efficient to use complex but approximated models than simple but fully optimized models. We thus propose to consider the composition of two polynomials for the mean function of Y. Indeed, the composition of two polynomial functions is still a polynomial function, but of much higher order. In particular, such a formalism can be used to model separately a transformation of each code input and the dependence structure between them.

The main difficulty concerning this specific representation is the identification of the parameters of the two combined polynomials. Indeed, by composing two polynomial functions that are linear with respect to their parameters, we get a strongly non-linear representation, which is likely to be very sensitive to small changes in the parameters values. In addition, distinct values for these parameters can lead to the same nested representation, which does not help for the identification. To avoid such redundancies, minimal nested parametrizations are introduced, and we show to what extent integrating this nested structure in the Gaussian process formalism can increase the robustness of the results, make easier the error control, and limit as much as possible over-fitting.

The outline of this work is as follows. First, Section 2 presents the theoretical framework for the definition of a Gaussian-process regression with a linear polynomial trend. Then, the nested polynomial trends we propose are detailed in Section 3. At last, the efficiency of the method is illustrated on a series of analytic examples in Section 4.

2. Gaussian process predictors

2.1. General framework

For $d \geq 1$, let $L^2(\mathcal{D}_d, \mathbb{R})$ be the space of square integrable functions on any compact subset $\mathcal{D}_d$ of $\mathbb{R}^d$, with values in $\mathbb{R}$, equipped with the inner product $(\cdot, \cdot)_{L^2}$, and the associated norm $\|\cdot\|_{L^2}$, such that for all u and v in $L^2(\mathcal{D}_d, \mathbb{R})$,

$$(u, v)_{L^2} := \int_{\mathcal{D}_d} u(\boldsymbol{x})v(\boldsymbol{x})d\boldsymbol{x}, \qquad \|u\|_{L^2}^2 := (u, u)_{L^2}. \tag{1}$$

Let $\mathcal{S}$ be a physical system, whose response depends on a d-dimensional input vector $\boldsymbol{x} = (x_1, \ldots, x_d)$, and whose performance can be evaluated from the computation of a quantity of interest, $g(\boldsymbol{x})$. Function g is a deterministic mapping that is assumed to be an element of $L^2(\mathcal{D}_d, \mathbb{R})$. In this work, it is supposed that the maximal available information about g is a set of N code evaluations at the points $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$ in $\mathcal{D}_d$. Given this information, we are interested in the identification of the *best* predictor $g^\star$ of g, in the sense that:

$$\forall \widehat{g} \in L^2(\mathcal{D}_d, \mathbb{R}), \qquad \left\| g - g^\star \right\|_{L^2}^2 \leq \left\| g - \widehat{g} \right\|_{L^2}^2. \tag{2}$$

In that context, the Gaussian process regression (GPR), or kriging, plays a major role [30,24,31,34]. It is indeed able to provide a prediction of $g(\boldsymbol{x})$, which is optimal in the class of the linear predictors of g, and whose precision can be *a posteriori* quantified. Such a method considers function g as a sample path of a real-valued Gaussian stochastic process Y, which is defined on the probability space $(\Omega, \mathcal{T}, \mathbb{P})$. Let μ and C be respectively the mean and the covariance functions of Y:

$$Y \sim \text{GP}(\mu, C). \tag{3}$$

We can introduce $\mathcal{F}_N$ the σ-algebra generated by the available information about g,

$$\boldsymbol{y} = \left(y^{(1)} = g(\boldsymbol{x}^{(1)}), \dots, y^{(N)} = g(\boldsymbol{x}^{(N)}) \right), \tag{4}$$

such that $\mathbb{P}(\,\cdot\,|\,\mathcal{F}_N)$ and $\mathbb{E}[\,\cdot\,|\,\mathcal{F}_N]$ denote the conditional probability and conditional mathematical expectation respectively.

Therefore, gathering in the vector $\boldsymbol{\mu}$ and in the matrix $\boldsymbol{C}$ the evaluations of μ and C at the available points, such that:

$$\begin{cases} \boldsymbol{\mu} := \left(\mu(\boldsymbol{x}^{(1)}), \dots, \mu(\boldsymbol{x}^{(N)}) \right), \\ \boldsymbol{C}_{ij} := C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}), \quad 1 \leq i, j \leq N, \end{cases} \tag{5}$$

it can be shown [25] that if matrix $\boldsymbol{C}$ is invertible, then:

$$Y \,|\, \mathcal{F}_N \sim \mathrm{GP}(\mu_N, C_N), \tag{6}$$

where, for all $\boldsymbol{x}, \boldsymbol{x}'$ in $\mathcal{D}_d$:

$$\begin{cases} \mu_N(\boldsymbol{x}) := \mu(\boldsymbol{x}) + \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{C}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}), \\ C_N(\boldsymbol{x}, \boldsymbol{x}') := C(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{r}(\boldsymbol{x}'), \\ \boldsymbol{r}(\boldsymbol{x}) := \left(C(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \dots, C(\boldsymbol{x}, \boldsymbol{x}^{(N)}) \right). \end{cases} \tag{7}$$

Under this formalism, the best prediction of g in an unobserved point $\boldsymbol{x}$ is given by the mean value of $(Y(\boldsymbol{x}) \,|\, \mathcal{F}_N)$, $\mu_N(\boldsymbol{x})$, whereas $C_N(\boldsymbol{x}, \boldsymbol{x})$ quantifies the trust we can put in that prediction.

In practice, it appears that $\boldsymbol{C}$ may not be invertible due to numerical reasons. This can generally be overcome by adding a small nugget to the covariance matrix and optimizing with respect to it too (see [12]).

2.2. Choice of the covariance function

Without information about the regularity of g, function C is generally chosen in general parametric families. In this work, function C is supposed to be an element of the Matern-5/2 class, such that for all $\boldsymbol{x}, \boldsymbol{x}'$ in $\mathcal{D}_d$:

$$C(\boldsymbol{x}, \boldsymbol{x}') := \sigma^2 \prod_{i=1}^{d} (1 + \sqrt{5}h_i + 5h_i^2/3) \exp(-\sqrt{5}h_i), \qquad h_i = |x_i - x_i'|/\ell_i. \tag{8}$$

Hence, covariance function C is characterized by a vector of hyper-parameters, $\boldsymbol{\Theta} := (\sigma, \ell_1, \dots, \ell_d)$, whose values also have to be conditioned by $\mathcal{F}_N$. More details about other usual parametric expressions for C can be found in [31]. A *full Bayesian* approach would then require the introduction of a prior distribution for this vector, and the use of sampling techniques (such as Monte Carlo Markov Chains [29]) to approximate the posterior distribution of $(Y \,|\, \mathcal{F}_N)$ [13,17,4]. In this work, we will adopt an alternative approach, which consists in conditioning all the results by the maximum likelihood estimate of the covariance parameters. This method, which is generally called *plug-in* approach, has been used in many previous papers for the definition of Gaussian process-based predictors, as it presents a good compromise between complexity, efficiency, and errors control [3,2]. In that case, explicit formula can be derived to evaluate the relevance of the GPR-based metamodel from a cross validation procedure [8].

2.3. Choice of the mean function

In the same manner than for the covariance function, the mean function of Y is supposed to be parametrized by a M-dimensional vector $\boldsymbol{\beta}$. In the general case, the computation of $\mathbb{E}[Y(\boldsymbol{x}) \,|\, \mathcal{F}_N)]$ is not direct, but if:

- covariance function C is known,
- μ is linear with respect to $\boldsymbol{\beta}$, that is to say it exists a M-dimensional vector-valued function $\boldsymbol{f}$ such that $\mu(\boldsymbol{x}) = \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\beta} \rangle$, where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^M$,
- $\boldsymbol{\beta}$ is uniformly distributed on $\mathbb{R}^M$ (improper prior distribution),

then (see [31] for further details):

$$Y \,|\, \mathcal{F}_N \sim \mathrm{GP}(\mu_{\mathrm{UK}}, C_{\mathrm{UK}}), \tag{9}$$

$$\begin{cases} \mu_{\mathrm{UK}}(\boldsymbol{x}) := \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\beta}^\star \rangle + \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{C}^{-1} (\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}^\star), \\ C_{\mathrm{UK}}(\boldsymbol{x}, \boldsymbol{x}') := C_N(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{u}(\boldsymbol{x})^T (\boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{F})^{-1} \boldsymbol{u}(\boldsymbol{x}'), \\ \boldsymbol{\beta}^\star := (\boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{y}, \\ \boldsymbol{u}(\boldsymbol{x}) := \boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}), \\ \boldsymbol{F}_{nm} := f_m(\boldsymbol{x}^{(n)}), \quad 1 \leq m \leq M, \ 1 \leq n \leq N, \end{cases} \tag{10}$$

where "UK" stands for Universal Kriging, and corresponds to the former plug-in approach, and where the term $\boldsymbol{u}(\boldsymbol{x})^T(\boldsymbol{F}^T\boldsymbol{C}^{-1}\boldsymbol{F})^{-1}\boldsymbol{u}(\boldsymbol{x}')$ can be interpreted as the prediction uncertainty that is due to the estimation of $\boldsymbol{\beta}$. Under these assumptions, the best prediction of $g(\boldsymbol{x})$ is now given by μ_{UK}. The last thing that can be done to minimize $\|g - \mu_{\text{UK}}\|_{L^2}$ is working on the choice of $\boldsymbol{f}$.

Without information about g, polynomials are generally chosen for $\boldsymbol{f}$. Indeed, the set $\{m_{\boldsymbol{\alpha}}, \ \boldsymbol{\alpha} \in \mathbb{N}^d\}$, with

$$m_{\boldsymbol{\alpha}}(\boldsymbol{x}) := x_1^{\alpha_1} \times \cdots \times x_d^{\alpha_d}, \quad \boldsymbol{x} \in \mathcal{D}_d, \tag{11}$$

defines a basis of $L^2(\mathcal{D}_d, \mathbb{R})$. For a given value of M, characterizing $\boldsymbol{f}$ amounts at identifying the best M-dimensional subset of $\{m_{\boldsymbol{\alpha}}, \ \boldsymbol{\alpha} \in \mathbb{N}^d\}$ to minimize $\|g - \mu_{\text{UK}}\|_{L^2}$.

In practice, this optimization problem over a very vast space is replaced by an optimization over a finite dimensional subset of $\{m_{\boldsymbol{\alpha}}, \ \boldsymbol{\alpha} \in \mathbb{N}^d\}$. Different truncation schemes have been proposed to choose such a relevant subset, which are mostly based on the assumption that the most influential elements of $\{m_{\boldsymbol{\alpha}}, \ \boldsymbol{\alpha} \in \mathbb{N}^d\}$ correspond to the elements of lowest total polynomial order. Denoting by r the maximal polynomial order of the projection basis, we can introduce:

$$\mathcal{P}(r, d) := \left\{ m_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^d, \ \sum_{i=1}^{d} |\alpha_i| \leq r \right\}. \tag{12}$$

By construction, it can be noticed that the cardinal of $\mathcal{P}(r, d)$, $\mathcal{C}(r, d)$, increases exponentially with respect to r and d:

$$\mathcal{C}(r, d) = (d + r)!/(d! \times r!). \tag{13}$$

For $M \leq \mathcal{C}(r, d)$, vector $\boldsymbol{f}$ can finally be searched using a penalization technique, such as the Least Angle Regression (LAR) method [14,9,5], which allows disregarding insignificant terms. Such an approach will be referred as "LAR + UK" approach in the following.

3. Nested polynomial trends for Gaussian process predictors

As presented in Introduction, we are interested in identifying the best predictor of g in any unobserved point $\boldsymbol{x}$ in $\mathcal{D}_d$, when the maximal information is a fixed number of code evaluations. Instead of considering sparse representations for the parametrization of the mean function in the GPR formalism, this section proposes to focus on nested polynomial representations. First, the notations and the motivations for this new parametrization are presented. Then, it is explained why and how it is integrated in the GPR formalism. Finally, a method to *a posteriori* evaluate the projection error is introduced.

3.1. Nested polynomial representations

Using the notations given by Eqs. (12) and (13), for p, q, u in $\mathbb{N}^*$, let $\boldsymbol{m}^{(p,u)}$ and $\boldsymbol{m}^{(q,d)}$ be the vector-valued functions that gather all the elements of $\mathcal{P}(p, u)$ and $\mathcal{P}(q, d)$ respectively, and let $\mathcal{C}(p, u)$ and $\mathcal{C}(q, d)$ be their respective dimensions. The elements of these two vectors are supposed to be sorted in an increasing total polynomial order. In particular, it comes:

$$m_1^{(p,u)} = m_1^{(q,d)} = 1. \tag{14}$$

Hence, for all $(u \times \mathcal{C}(q, d))$-dimensional matrix $\boldsymbol{A}$ and all $\mathcal{C}(p, u)$-dimensional vector $\boldsymbol{b}$, the mapping

$$\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{m}^{(q,d)}(\boldsymbol{x}) \tag{15}$$

is a function with values in $\mathbb{R}^u$, and the mapping

$$\boldsymbol{x} \mapsto \left\langle \boldsymbol{m}^{(p,u)}(\boldsymbol{A}\boldsymbol{m}^{(q,d)}(\boldsymbol{x})), \boldsymbol{b} \right\rangle \tag{16}$$

defines a nested polynomial representation. For $d > 1$, such a representation allows us to model separately the dependency structure between the different input parameters, which is characterized by p and u, and the individual actions of each input parameter, which are characterized by the polynomial order q (considering different values of q for each input could eventually be done to optimize such a two-scale modeling). Hence, analyzing the optimal values of p, u and q can bring information about the structure of g. For instance, if $p = 1$ and $u = d$, then g is just an additive model, up to a transformation of its input parameters. In the same manner, a value of q strictly greater than 1 tends to say that the relation between $\boldsymbol{x}$ and g is multi-scale.

An other interesting property of this nested structure comes from the fact that, for all $\boldsymbol{x}$ in $\mathbb{R}^d$:

$$\begin{aligned}
\left\langle \boldsymbol{m}^{(p,u)}(\boldsymbol{A}\boldsymbol{m}^{(q,d)}(\boldsymbol{x})), \boldsymbol{b} \right\rangle &= \sum_{0 \leq |\alpha_1| + \cdots + |\alpha_u| \leq p} b_{(\alpha_1, \ldots, \alpha_u)} \times \prod_{i=1}^{u} \left(\sum_{k=1}^{\mathcal{C}(q,d)} \boldsymbol{A}_{ik} m_k^{(q,d)}(\boldsymbol{x}) \right)^{\alpha_i}, \\
&= \sum_{0 \leq |\widetilde{\alpha}_1| + \cdots + |\widetilde{\alpha}_d| \leq p \times q} x_1^{\widetilde{\alpha}_1} \times \cdots \times x_d^{\widetilde{\alpha}_d} \ \widetilde{c}_{\widetilde{\boldsymbol{\alpha}}}(\boldsymbol{A}, \boldsymbol{b}; u),
\end{aligned} \tag{17}$$

Table 1
Comparison between the dimension of the projection set, $\mathcal{C}(p \times q, d)$, and the number of independent parameters to characterize the associated projection coefficients in the proposed nested approach, #Coeff$(d, p, q, u) = \mathcal{C}(p, u) + (\mathcal{C}(q, d) - 1) - u$, for $q = p = 3$, $d \in \{1, 2, 5, 10, 20\}$ and $u \in \{1, d\}$.

Values of d	$\mathcal{C}(p \times q, d)$	#Coeff$(d, p, q, u = 1)$	#Coeff$(d, p, q, u = d)$
1	10	6	6
2	55	12	17
5	2002	58	106
10	92378	288	561
20	10015005	1773	3521

where $\widetilde{c}_{\widetilde{\alpha}}(A, b; u)$ is the projection coefficient of $\langle m^{(p,u)}(Am^{(q,d)}(x)), b \rangle$ on $x_1^{\widetilde{\alpha}_1} \times \cdots \times x_d^{\widetilde{\alpha}_d}$. Hence, function $x \mapsto \langle m^{(p,u)}(Am^{(q,d)}(x)), b \rangle$ is in Span $\{\mathcal{P}(p \times q, d)\}$, while being characterized by $\mathcal{C}(p, u) + u \times \mathcal{C}(q, d)$ parameters. Thus, by choosing u such that the ratio $(\mathcal{C}(p, u) + u \times \mathcal{C}(q, d))/\mathcal{C}(p \times q, d)$ is small, it is possible to parametrize polynomial families with very high cardinality, with only a reduced number of parameters. Such a parametrization is however redundant, in the sense that several distinct values of A and b lead to the same nested representations. From Eq. (17), it can be seen that some of these redundancies can be avoided by imposing that:

$$\begin{cases} A_{i1} = 0, \\ \sum_{k=1}^{\mathcal{C}(q,d)} A_{ik}^2 = 1, & 1 \le i \le u. \end{cases} \tag{18}$$

For fixed values of p and q, it is clear that ratio $(\mathcal{C}(p, u) + u \times \mathcal{C}(q, d))/\mathcal{C}(p \times q, d)$ is minimal when $u = 1$. However, considering higher values of u strongly increases the flexibility of the nested representation to approximate function g. In this work, as a compromise between flexibility and minimal parametrization, for all $2 \le k \le \mathcal{C}(q, d)$, we thus propose to fix to zero all the components of $(A_{1k}, \ldots, A_{uk})$ but one. This means that each component of vector $m^{(q,d)}(x)$ is used only once in the construction of $Am^{(q,d)}(x)$, and that only #Coeff$(d, p, q, u) = \mathcal{C}(p, u) + (\mathcal{C}(q, d) - 1) - u$ independent parameters have to be fixed to span a $\mathcal{C}(p \times q, d)$-dimensional projection set. As it can be seen in Table 1 and as it will be shown in Section 4, this assumption is indeed very attractive in terms of dimension reduction while being particularly interesting for the modeling of complex phenomena with very limited information.

To simplify the notations of the next sections, these $\mathcal{C}(q, d) - 1$ non-zero coefficients of A are supposed to be gathered in a vector a, and we introduce the vector $P^{(q,d)}(x)$ such that for all $x \in \mathcal{D}_d$:

$$P^{(q,d)}(x)a := Am^{(q,d)}(x). \tag{19}$$

For given values of a and b, we then denote by $\mu(\cdot; a, b)$ the following nested representation:

$$\mu(x; a, b) := \langle m^{(p,u)}(P^{(q,d)}(x)a), b \rangle, \quad x \in \mathcal{D}_d. \tag{20}$$

Finally, for given values of u, p, q, the most appropriate nested representation to approximate function g is given by $\mu(\cdot; a^*, b^*)$, where (a^*, b^*) is the solution of the following optimization problem:

$$(a^*, b^*) := \arg \min_{(a,b) \in \mathcal{S}^*} \| g - \mu(\cdot; a, b) \|_{L^2}^2, \tag{21}$$

where the admissible searching set, $\mathcal{S}^*$, is a subset of $\mathbb{R}^{\mathcal{C}(q,d)-1} \times \mathbb{R}^{\mathcal{C}(p,u)}$ that takes into account the constraints on a defined by Eqs. (18) and (19).

There are three main difficulties concerning the optimization problem defined by Eq. (21). First, as the maximal information about g is a N-dimensional set of evaluations, for given values of a and b, the norm $\| g - \mu(\cdot; a, b) \|_{L^2}^2$ has to be approximated. If the evaluation points $\{x^{(1)}, \ldots, x^{(N)}\}$ are (more or less) uniformly distributed on $\mathcal{D}_d$, a (rather) good estimation of this norm is given by its least squares approximation,

$$\frac{1}{N} \sum_{n=1}^{N} \left(g(x^{(n)}) - \mu(x^{(n)}; a, b) \right)^2 = \frac{1}{N} \| y - M(a)b \|^2, \tag{22}$$

where $\|\cdot\|$ is the classical Euclidean norm, vector y is defined by Eq. (4), and $M(a)$ is a $(N \times \mathcal{C}(p, u))$-dimensional matrix such that:

$$(M(a))_{nk} = m_k^{(p,u)}(P^{(q,d)}(x^{(n)})a), \quad 1 \le n \le N, \ 1 \le k \le \mathcal{C}(p, u). \tag{23}$$

Noticing that for all $(\boldsymbol{a}, \boldsymbol{b})$ in $\mathcal{S}^*$,

$$\left\| \boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \left(\boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{M}(\boldsymbol{a}) \right)^{-1} \boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{y} \right\|^2 \leq \| \boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b} \|^2, \tag{24}$$

the solutions, $\boldsymbol{a}^*$ and $\boldsymbol{b}^*$, of the minimization problem defined by Eq. (21) can respectively be approximated by the vectors $\boldsymbol{a}^{LS}$ and $\boldsymbol{b}^{LS}(\boldsymbol{a}^{LS})$, with:

$$\begin{cases} \boldsymbol{a}^{LS} = \arg \min_{\boldsymbol{a} \in \mathcal{S}_{\boldsymbol{a}}^*} \left\| \boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b}^{LS}(\boldsymbol{a}) \right\|^2, \\ \boldsymbol{b}^{LS}(\boldsymbol{a}) = \left(\boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{M}(\boldsymbol{a}) \right)^{-1} \boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{y}, \end{cases} \tag{25}$$

where $\mathcal{S}_{\boldsymbol{a}}^*$ is a subset of $\mathbb{R}^{\mathcal{C}(q,d)-1}$ that also takes into account the constraints on $\boldsymbol{a}$ defined by Eqs. (18) and (19).

The second difficulty comes from the fact that the minimization of $\boldsymbol{a} \mapsto \left\| \boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b}^{LS}(\boldsymbol{a}) \right\|^2$ can be complex. This is due to the fact that this mapping is strongly non-linear, leading to a strongly non-convex problem. For high values of p, q and u, even if non-convex optimization algorithms such as simulated annealing or simplex algorithms [6] are used, there is no guarantee that the global minimization can be found in a reasonable computational time.

At last, there is a risk that $\left\| \boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}^{LS}) \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS}) \right\|^2 / N$ strongly underestimates $\left\| g - \mu(\cdot; \boldsymbol{a}^{LS}, \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS})) \right\|_{L^2}^2$, as the same information is used twice: once for the optimization and once for the error estimation. To avoid such an over-fitting, classical Leave-One-Out (LOO) techniques (see [22,5,28]) have to be introduced to get a relevant approximation of $\left\| g - \mu(\cdot; \boldsymbol{a}^{LS}, \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS})) \right\|_{L^2}^2$.

3.2. Coupling nested representations and Gaussian processes

Once vector $\boldsymbol{a}^{LS}$ has been identified from the solving of Eq. (25), the notion of confidence intervals for the prediction of $g(\boldsymbol{x})$ in an unobserved point $\boldsymbol{x}$ can be found back by assuming that g is a particular realization of a Gaussian stochastic process, whose statistical properties are given by:

$$Y \sim \mathrm{GP}(\mu(\boldsymbol{a}^{LS}, \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS})), C(\boldsymbol{\Theta}^*)), \tag{26}$$

where $\boldsymbol{\Theta}^*$ gathers the $d+1$ parameters of the Matern-5/2 covariance C defined by Eq. (8), which are solution of the following log-likelihood maximization problem:

$$\boldsymbol{\Theta}^* = \arg \max_{\boldsymbol{\Theta} \in]0, +\infty[^{d+1}} -\frac{1}{2} \left\{ \begin{array}{l} N \log(2\pi) + \log(\det(\boldsymbol{C}(\boldsymbol{\Theta}))) \\ + (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}^{LS}) \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS}))^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}^{LS}) \boldsymbol{b}^{LS}(\boldsymbol{a}^{LS})) \end{array} \right\}. \tag{27}$$

Such a naive coupling is nevertheless sub-optimal, as the values of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$ are optimized separately: the nested structure does not take advantage of the Bayesian formalism, and reciprocally. Instead of such a two-steps approach, we propose in this work to directly adopt a Bayesian formalism for the estimation of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$. In the plug-in formalism, this means that the statistical properties of Y are now given by:

$$Y \sim \mathrm{GP}(\mu(\boldsymbol{a}^*, \boldsymbol{b}^*)), C(\boldsymbol{\Theta}^*)), \tag{28}$$

where $(\boldsymbol{a}^*, \boldsymbol{b}^*, \boldsymbol{\Theta}^*)$ is the solution of the following log-likelihood maximization problem:

$$(\boldsymbol{a}^\star, \boldsymbol{b}^\star, \boldsymbol{\Theta}^\star) = \arg \max_{(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta}) \in \mathcal{S}^{\mathrm{adm}}} -\frac{1}{2} \left\{ \begin{array}{l} N \log(2\pi) + \log(\det(\boldsymbol{C}(\boldsymbol{\Theta}))) \\ + (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b}) \end{array} \right\}, \tag{29}$$

where the admissible searching set, $\mathcal{S}^{\mathrm{adm}}$, is a subset of $\mathbb{R}^{\mathcal{C}(q,d)-1} \times \mathbb{R}^{\mathcal{C}(p,u)} \times \mathbb{R}^{d+1}$ but is not trivial, as it first takes into account the constraints on $\boldsymbol{a}$ defined by Eqs. (18) and (19), but also guarantees that $\boldsymbol{C}(\boldsymbol{\Theta})$ and $\boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} \boldsymbol{M}(\boldsymbol{a})$ are invertible.

For all $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta})$ belonging to the admissible set, $\mathcal{S}^{\mathrm{adm}}$, denoting by L the function such that

$$L(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta}) = \log(\det(\boldsymbol{C}(\boldsymbol{\Theta}))) + (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} (\boldsymbol{y} - \boldsymbol{M}(\boldsymbol{a}) \boldsymbol{b}), \tag{30}$$

it is interesting to notice that, in the same manner than in Section 3.1,

$$L(\boldsymbol{a}, \boldsymbol{b}^{LS}(\boldsymbol{a}, \boldsymbol{\Theta}), \boldsymbol{\Theta}) \leq L(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta}), \tag{31}$$

$$\boldsymbol{b}^{LS}(\boldsymbol{a}, \boldsymbol{\Theta}) := \left(\boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} \boldsymbol{M}(\boldsymbol{a}) \right)^{-1} \boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{C}(\boldsymbol{\Theta}) \boldsymbol{y}. \tag{32}$$

It comes:

$$
\begin{cases}
(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) = \arg \min_{(\boldsymbol{a}, \boldsymbol{\Theta})} \mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}), \\
\boldsymbol{b}^{\star} = \left(\boldsymbol{M}(\boldsymbol{a}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \boldsymbol{M}(\boldsymbol{a}^{\star}) \right)^{-1} \boldsymbol{M}(\boldsymbol{a}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star}) \boldsymbol{y},
\end{cases}
\tag{33}
$$

where:

$$
\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}) := L(\boldsymbol{a}, \boldsymbol{b}^{\mathrm{LS}}(\boldsymbol{a}, \boldsymbol{\Theta}), \boldsymbol{\Theta}).
\tag{34}
$$

Function $(\boldsymbol{a}, \boldsymbol{\Theta}) \mapsto \mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta})$ being strongly non-regular and non-convex, it is proposed to work iteratively on the values of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$. Two reasons motivate this separation. First, the actions of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$ on $\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta})$ being very different, dividing the optimization problem tends to regularize the mappings on which the minimization is carried out. Second, by reducing each searching set, each minimization is made easier. Therefore, for a given convergence tolerance ε, Algorithm 1 is introduced for the minimization of $\mathcal{L}$. The convergence of such an iterative algorithm to the global minimum of $\mathcal{L}$ is of course not guaranteed, but it appeared on a series of numerical examples that it allowed us to identify good approximations of $(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})$ at a reasonable computational cost. As the minimization problem defined by Eq. (33) is not convex, better approximations of $\boldsymbol{a}^{\star}$ can be obtained by repeating several times Algorithm 1, with random initialization vectors $\boldsymbol{a}_0$ in $\mathcal{S}_{\boldsymbol{a}}^{*}$.

---

1 Initialization: $L_1 = 0$, $L_2 = +\infty$, $\boldsymbol{a}^{*} = \boldsymbol{a}_0 \in \mathcal{S}_{\boldsymbol{a}}^{*}$;
2 while $|L_2 - L_1| > \varepsilon$ **do**
3 $\quad L_1 = L_2$;
4 $\quad \boldsymbol{\Theta}^{*} = \arg \max_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{a}^{*}, \boldsymbol{\Theta})$;
5 $\quad \boldsymbol{a}^{*} = \arg \max_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}^{*})$;
6 $\quad L_2 = \min(L_2, \mathcal{L}(\boldsymbol{a}^{*}, \boldsymbol{\Theta}^{*}))$;
7 end
8 $\boldsymbol{a}^{\star} \approx \boldsymbol{a}^{*}$, $\boldsymbol{\Theta}^{\star} \approx \boldsymbol{\Theta}^{*}$.

Algorithm 1: Iterative minimization of function $\mathcal{L}$.

3.3. Linearization of the nested polynomial trend

Even for small values of p, q and u, the quantity $\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta})$ is sensitive to small changes in the values of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$, which makes the solving of the optimization problem defined by Eq. (33) difficult. In that context, it can be interesting to linearize the nested polynomial trend around the solutions given by Algorithm 1, $\boldsymbol{a}^{\star}$ and $\boldsymbol{b}^{\star}$, and then work on the compensations $(\boldsymbol{a} - \boldsymbol{a}^{\star})$ and $(\boldsymbol{b} - \boldsymbol{b}^{\star})$ that could make the prediction of function g better. In the vicinity of $\boldsymbol{a}^{\star}$ and $\boldsymbol{b}^{\star}$, for all $\boldsymbol{x}$ in $\mathcal{D}_d$, it comes:

$$
\mu(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{b}) \approx \left\langle \left(\boldsymbol{h}^{(1)}(\boldsymbol{x}; \boldsymbol{a}^{\star}, \boldsymbol{b}^{\star}), \boldsymbol{h}^{(2)}(\boldsymbol{x}; \boldsymbol{a}^{\star}) \right), (\boldsymbol{a} - \boldsymbol{a}^{\star}, \boldsymbol{b}) \right\rangle,
\tag{35}
$$

$$
\boldsymbol{h}^{(1)}(\boldsymbol{x}; \boldsymbol{a}^{\star}, \boldsymbol{b}^{\star}) = \boldsymbol{P}^{(q,d)}(\boldsymbol{x})^T \boldsymbol{D}(\boldsymbol{P}^{(q,d)}(\boldsymbol{x}) \boldsymbol{a}^{\star})^T \boldsymbol{b}^{\star},
\tag{36}
$$

$$
\boldsymbol{h}^{(2)}(\boldsymbol{x}; \boldsymbol{a}^{\star}) = \boldsymbol{m}^{(p,u)}(\boldsymbol{P}^{(q,d)}(\boldsymbol{x}) \boldsymbol{a}^{\star}),
\tag{37}
$$

$$
(\boldsymbol{D}(\boldsymbol{z}))_{kj} := \frac{\partial m_k^{(p,u)}}{\partial z_j}(\boldsymbol{z}), \quad 1 \le j \le u, \quad 1 \le k \le \mathcal{C}(p, u), \quad \boldsymbol{z} \in \mathbb{R}^u.
\tag{38}
$$

Now, let us denote by $\boldsymbol{\beta} := (\boldsymbol{a} - \boldsymbol{a}^{\star}, \boldsymbol{b})$ the new vector of parameters we need to determine, and by $\boldsymbol{f} := \left(\boldsymbol{h}^{(1)}(\cdot; \boldsymbol{a}^{\star}, \boldsymbol{b}^{\star}), \boldsymbol{h}^{(2)}(\cdot; \boldsymbol{a}^{\star}) \right)$ the new set of projection functions. Conditioned by the values of $\boldsymbol{a}^{\star}$, $\boldsymbol{b}^{\star}$ and $\boldsymbol{\Theta}^{\star}$, the formalism introduced in Section 2.3 is found back:

$$
Y \sim \mathrm{GP}(\langle \boldsymbol{f}, \boldsymbol{\beta} \rangle, C),
\tag{39}
$$

such that the distribution of $(Y \mid \mathcal{F}_N)$ can be calculated analytically. Its mean value can directly be used to predict the values of g, and its covariance function can allow us to quantify the confidence we can put in these predictions.

We underline at least two advantages for the linearization. First, the distribution of $(Y \mid \mathcal{F}_N)$ will be less dependent on the convergence properties of Algorithm 1, which are not easy to control. Secondly, as the covariance function of $(Y \mid \mathcal{F}_N)$ integrates the uncertainty associated with the least squares estimation of $\boldsymbol{\beta}$, that is to say the uncertainty associated with the estimation of $\boldsymbol{a}$ and $\boldsymbol{b}$ in the vicinity of $\boldsymbol{a}^{\star}$ and $\boldsymbol{b}^{\star}$, the confidence intervals associated with these predictions are expected to be more adapted.

3.4. Error evaluation

According to the previous Sections and to Eq. (10), for given values of truncation parameters p, q and u, we propose to use the deterministic function $\widehat{g}^{\text{nest}}(\boldsymbol{x})$, such that:

$$\widehat{g}^{\text{nest}}(\boldsymbol{x}) = \left\langle \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}), \boldsymbol{\beta}^{\star}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \right\rangle + \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\Theta}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \left(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \boldsymbol{\beta}^{\star}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \right), \tag{40}$$

$$\boldsymbol{\beta}^{\star}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) := (\boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}))^{-1} \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \boldsymbol{y}, \tag{41}$$

to predict the value of $g(\boldsymbol{x})$ for all $\boldsymbol{x}$ in $\mathcal{D}_d$, where:

- vectors $\boldsymbol{a}^{\star}$ and $\boldsymbol{\Theta}^{\star}$ are the solutions of the optimization problem given by Eq. (33), under the additional condition that the matrix $\boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})$ is invertible,
- vector $\boldsymbol{y}$ is defined by Eq. (4),
- the function $\boldsymbol{x} \mapsto \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})$ gathers the most influential terms of the vector-valued function $(\boldsymbol{h}^{(1)}(\cdot; \boldsymbol{a}^{\star}, \boldsymbol{b}^{\text{LS}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})),$ $\boldsymbol{h}^{(2)}(\cdot; \boldsymbol{a}^{\star}))$, which have been identified from a LAR procedure,
- $\boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) := [\boldsymbol{f}(\boldsymbol{x}^{(1)}; \boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \cdots \boldsymbol{f}(\boldsymbol{x}^{(N)}; \boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})]$ is the matrix that gathers the evaluations of $\boldsymbol{f}(\cdot; \boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})$ at the available code evaluations,
- and for all $1 \leq n, m \leq N$, $\boldsymbol{C}(\boldsymbol{\Theta}^{\star})_{nm} = C(\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(m)})$ and $r_n(\boldsymbol{x}; \boldsymbol{\Theta}^{\star}) = C(\boldsymbol{x}, \boldsymbol{x}^{(n)})$, with C the Matern-5/2 covariance function of parameters $\boldsymbol{\Theta}^{\star}$.

In the same manner than in Section 2, as function g is only known through a limited number of evaluations, classical Leave-One-Out (LOO) techniques have to be introduced to approximate the relevance of such a predictor:

$$\left\| g - \widehat{g}^{\text{nest}} \right\|_{L_2}^2 \approx \epsilon_{\text{LOO}}^2 := \frac{1}{N} \sum_{n=1}^{N} \left(g(\boldsymbol{x}^{(n)}) - \widehat{g}_{-n}^{\text{nest}}(\boldsymbol{x}^{(n)}) \right)^2, \tag{42}$$

where, for all $1 \leq n \leq N$, the function $\widehat{g}_{-n}^{\text{nest}}$ has been constructed in the same manner than $\widehat{g}^{\text{nest}}$, but using the $N - 1$ evaluations of the code in $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n-1)}, \boldsymbol{x}^{(n+1)}, \ldots, \boldsymbol{x}^{(N)}\}$ only.

In order to reduce the computational cost associated with the evaluation of ϵ_{LOO}^2, it is interesting to notice (see [8] for further details) that for all $1 \leq n \leq N$:

$$g(\boldsymbol{x}^{(n)}) - \widehat{g}_{-n}^{\text{nest}}(\boldsymbol{x}^{(n)}) = \frac{(\widehat{\boldsymbol{C}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \boldsymbol{y})_n}{\widehat{\boldsymbol{C}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})_{nn}}, \tag{43}$$

$$\widehat{\boldsymbol{C}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) = \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} - \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \widehat{\boldsymbol{F}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1}, \tag{44}$$

$$\widehat{\boldsymbol{F}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) := \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})(\boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})^T \boldsymbol{C}(\boldsymbol{\Theta}^{\star})^{-1} \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}))^{-1} \boldsymbol{F}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})^T. \tag{45}$$

LOO error ϵ_{LOO}^2 can then be approximated by:

$$\epsilon_{\text{LOO}}^2 \approx \widehat{\epsilon}_{\text{LOO}}^2 := \frac{1}{N} \sum_{n=1}^{N} \widehat{e}_n^2, \quad \widehat{e}_n^2 := \left\{ \frac{(\widehat{\boldsymbol{C}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star}) \boldsymbol{y})_n}{\widehat{\boldsymbol{C}}(\boldsymbol{a}^{\star}, \boldsymbol{\Theta}^{\star})_{nn}} \right\}^2. \tag{46}$$

Such an approximation is however conditioned by the values of $\boldsymbol{a}^{\star}$ and $\boldsymbol{\Theta}^{\star}$, which are computed using all the code evaluations. In order to be more precise, it can be noticed that for all $\boldsymbol{a}$, $\boldsymbol{\Theta}$, $1 \leq n \leq N$:

$$\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}) = \mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta}) + \frac{(\widetilde{\boldsymbol{C}}(\boldsymbol{a}, \boldsymbol{\Theta}) \boldsymbol{y})_n^2}{\widetilde{\boldsymbol{C}}(\boldsymbol{a}, \boldsymbol{\Theta})_{nn}}, \tag{47}$$

$$\widetilde{\boldsymbol{C}}(\boldsymbol{a}, \boldsymbol{\Theta}) = \boldsymbol{C}(\boldsymbol{\Theta})^{-1} \left\{ \boldsymbol{I} - \boldsymbol{M}(\boldsymbol{a})(\boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} \boldsymbol{M}(\boldsymbol{a}))^{-1} \boldsymbol{M}(\boldsymbol{a})^T \boldsymbol{C}(\boldsymbol{\Theta})^{-1} \right\}, \tag{48}$$

where $\boldsymbol{I}$ is the identity matrix and $\mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta})$ is the evaluation of function $\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta})$ based on the $N - 1$ evaluations of the code in $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n-1)}, \boldsymbol{x}^{(n+1)}, \ldots, \boldsymbol{x}^{(N)}\}$ only. Hence, in the optimization process leading us to the identification of $\boldsymbol{a}^{\star}$ and $\boldsymbol{\Theta}^{\star}$, let $\{(\boldsymbol{a}_i, \boldsymbol{\Theta}_i), 1 \leq i \leq N_{\text{test}}\}$ be the N_{test} values of $\boldsymbol{a}$ and $\boldsymbol{\Theta}$, in which function $\mathcal{L}$ has been evaluated. With very limited additional computational cost, we can then define, for all $1 \leq n \leq N$, the LOO evaluations of $\boldsymbol{a}^{\star}$ and $\boldsymbol{\Theta}^{\star}$, which are denoted by $\boldsymbol{a}_{-n}^{\star}$ and $\boldsymbol{\Theta}_{-n}^{\star}$ respectively, and which are given by:

$$(\boldsymbol{a}_{-n}^{\star}, \boldsymbol{\Theta}_{-n}^{\star}) = \arg \min_{(\boldsymbol{a}, \boldsymbol{\Theta}) \in \{(\boldsymbol{a}_i, \boldsymbol{\Theta}_i), \, 1 \leq i \leq N_{\text{test}}\}} \mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta}). \tag{49}$$

Finally, we can introduce error $\widetilde{\epsilon}_{\text{LOO}}$, such that:

$$\left\| g - \widehat{g}^{\text{nest}} \right\|_{L_2}^2 \approx \widetilde{\epsilon}_{\text{LOO}}^2 := \frac{1}{N} \sum_{n=1}^{N} \widetilde{e}_n^2, \quad \widetilde{e}_n^2 := \left\{ \frac{(\widehat{\boldsymbol{C}}(\boldsymbol{a}_{-n}^{\star}, \boldsymbol{\Theta}_{-n}^{\star}) \boldsymbol{y})_n}{\widehat{\boldsymbol{C}}(\boldsymbol{a}_{-n}^{\star}, \boldsymbol{\Theta}_{-n}^{\star})_{nn}} \right\}^2. \tag{50}$$

3.5. Convergence analysis

All the developments presented in Sections 3.1 and 3.2 are conditioned by the values of three truncation parameters, p, q and u, which have to be identified from a convergence analysis. As presented in Section 3.1, we remind that the roles of p, q and u in the modeling of g are different. Whereas p and u are associated with the modeling of the dependency structure between the input parameters, q is associated with the individual transformation of each input. As a consequence, q is strongly dependent on the dimension of vector $\boldsymbol{a}$, which parametrizes these individual transformations. On the contrary, this dimension of $\boldsymbol{a}$, which is equal to $\mathcal{C}(q, d) - 1 - u$, does not depend on p, but depends only linearly on u. Hence, increasing the values of p and u does not really increase the dimension of the search set for the identification of $\boldsymbol{a}^\star$, but makes the relation between $\boldsymbol{a}$ and $\mathcal{L}(\boldsymbol{a}, \Theta)$ much more complex.

To choose u, p and q, maximal values are *a priori* chosen, which are written $u^{\max}$, $p^{\max}$ and $q^{\max}$ respectively. In this work, as we want to reduce the number of parameters on which the polynomial trend is based, only values of u that are lower than d are considered: $u^{\max} = d$. Finally, the optimal value of (u, p, q) is the one that gives the minimum LOO error after having tested all these combinations of values:

$$(u^\star, p^\star, q^\star) := \arg \min_{1 \leq u \leq d,\ 1 \leq p \leq p^{\max},\ 1 \leq q \leq q^{\max}} \widetilde{\epsilon}^2_{\text{LOO}}(u, p, q), \tag{51}$$

where error $\widetilde{\epsilon}^2_{\text{LOO}}$ is defined by Eq. (46).

4. Applications

To illustrate the advantages of the nested structure presented in Section 3 for the modeling of quantity of interest g, this section introduces a series of analytic examples, which are sorted with respect to the input set dimension, d. In each case, the proposed approach is compared to the "LAR+UK" approach, which has been described in Section 2. For each function g, let $\widehat{g}^{\text{nest}}$ and $\widehat{g}^{\text{LAR+UK}}$ be the best approximations of g we can get from the available information, when considering a nested polynomial trend and a simple polynomial trend, respectively. Let $\varepsilon^2_{\text{NEST}}$ and $\varepsilon^2_{\text{LAR+UK}}$ be the associated normalized errors, such that:

$$\varepsilon^2_{\text{NEST}} = \left\| g - \widehat{g}^{\text{nest}} \right\|^2_{L^2} / \left\| g \right\|^2_{L^2}, \tag{52}$$

$$\varepsilon^2_{\text{LAR+UK}} = \left\| g - \widehat{g}^{\text{LAR+UK}} \right\|^2_{L^2} / \left\| g \right\|^2_{L^2}. \tag{53}$$

When dealing with a simple polynomial trend, it is reminded that the only truncation parameter that needs to be identified is the maximal total polynomial order, which will be denoted in the following by $p^{\text{LAR+UK}}$ for the sake of clarity. On the contrary, three truncation parameters have to be identified for the nested polynomial trends: p, u and q. As a consequence, the required computational time to identify $\widehat{g}^{\text{nest}}$ can be much higher than the one to identify $\widehat{g}^{\text{LAR+UK}}$.

4.1. $d = 1$

In this part, we suppose that $d = 1$, and we fix $\mathcal{D}_d = [-1, 1]$. Three analytic expressions for g are then proposed:

- case 1: $g(x) = P_2 \circ P_1(x)$,
- case 2: $g(x) = \sin((x + 1)^3)$,
- case 3: $g(x) = \sin(20x) \cos(2x)$,

where, for all x in $[-1, 1]$:

$$\begin{cases} P_1(x) = \displaystyle\sum_{i=1}^{5} c_i^{(1)} x^{i-1}, \quad \boldsymbol{c}^{(1)} = \dfrac{(0, -0.03, 0.5, -0.4, -0.5)}{\sqrt{0.03^2 + 0.5^2 + 0.4^2 + 0.5^2}}, \\[4mm] P_2(x) = \displaystyle\sum_{i=1}^{5} c_i^{(2)} x^{i-1}, \quad \boldsymbol{c}^{(2)} = (-0.1, 0.2, 0.7, -0.2, -0.2). \end{cases} \tag{54}$$

For each case, Fig. 1 compares the evolution of errors $\varepsilon^2_{\text{NEST}}$ and $\varepsilon^2_{\text{LAR+UK}}$ with respect to N, the number of available evaluations of g. For each value of N, convergence analyses have been performed for both methods. The maximal values for the truncation parameters associated were fixed such that:

$$0 \leq p^{\text{LAR+UK}} \leq 20, \quad 0 \leq p, q \leq 10, \quad u = 1. \tag{55}$$

For the three applications, these convergence analyses lead us to relatively high values for these truncation parameters ($p \geq 4$, $q \geq 4$). As underlined in Section 3.1, this can be explained by the ability of the proposed nested structure to
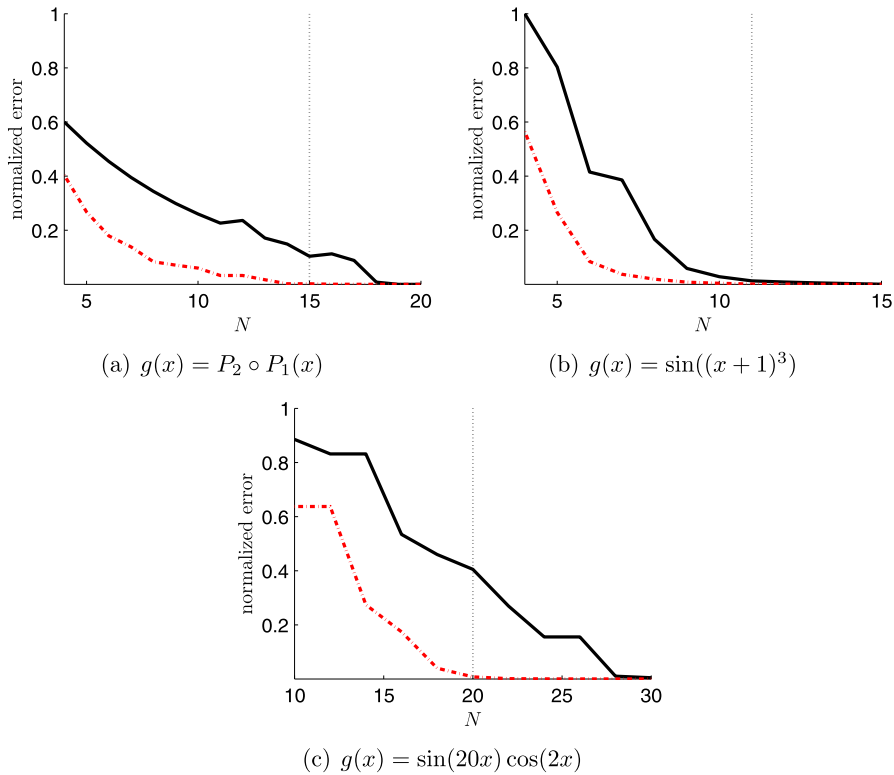
(a) $g(x) = P_2 \circ P_1(x)$

(b) $g(x) = \sin((x+1)^3)$

(c) $g(x) = \sin(20x)\cos(2x)$

Fig. 1. Evolution of the normalized L^2 errors with respect to N, the number of code evaluations. To be more representative, for each value of N, the LAR + UK and the proposed approaches have been repeated 10 times on randomly chosen learning sets. The curves correspond to the mean value of the errors associated with these 10 repetitions. Solid black line: evolution of the error associated with the LAR + UK approach, $\varepsilon^2_{\text{LAR+UK}}$. Red dotted line: evolution of the error associated with the proposed approach, $\varepsilon^2_{\text{NEST}}$. The vertical bar indicates moreover the value of N on which the results of Fig. 2 are focused. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

parametrize polynomial families with very high cardinality with only few parameters. This is particularly efficient when N is small compared to the number of oscillations of g.

In addition, Fig. 2 compares the two approaches in term of prediction for given values of N. In these figures we notice that the proposed method is particularly adapted to the cases when g presents a nested structure or when it is oscillating. This is particularly true when N is small compared to the complexity of g.

4.2. $d > 1$

The idea of this section is to show that the tendencies that were noticed in the one-dimensional cases are found back when considering multidimensional input spaces. To this end, let us consider the three following expressions of g, and the associated maximal values for the convergence analyses:

- Case 1: $d = 2$, $0 \le p^{\text{LAR+UK}} \le 20$, $0 \le p \le 6$, $0 \le q \le 10$, $1 \le u \le d$.

$$g : \begin{cases} [-1,1]^2 \to & [-1,1] \\ \boldsymbol{x} \mapsto g^{2D}(\boldsymbol{x}) = (1 - x_1^2)\cos(7x_1) \times (1 - x_2^2)\sin(5x_2) \end{cases}. \tag{56}$$

- Case 2 (the Ishigami function): $d = 3$, $0 \le p^{\text{LAR+UK}} \le 20$, $0 \le p \le 3$, $0 \le q \le 10$, $1 \le u \le d$.

$$g : \begin{cases} [-\pi,\pi]^3 \to & \mathbb{R} \\ \boldsymbol{x} = (x_1, x_2, x_3) \mapsto g^{3D}(\boldsymbol{x}) = \sin(x_1) + 7\sin(x_2)^2 + 0.1x_3^4 \sin(x_1) \end{cases}. \tag{57}$$

- Case 3: $d = 6$, $0 \le p^{\text{LAR+UK}} \le 10$, $0 \le p \le 3$, $0 \le q \le 10$, $1 \le u \le d$.

$$g : \begin{cases} [-1,1]^6 \to & \mathbb{R} \\ \boldsymbol{x} \mapsto g^{6D}(\boldsymbol{x}) = g^{(1)} \circ \boldsymbol{g}^{(2)}(\boldsymbol{x}), \end{cases} \tag{58}$$

$$g^{(1)}(\boldsymbol{z}) = 0.1 \cos\left(\sum_{i=1}^{6} z_i\right) + \sum_{i=1}^{6} z_i^2, \quad \boldsymbol{z} \in \mathbb{R}^6, \tag{59}$$
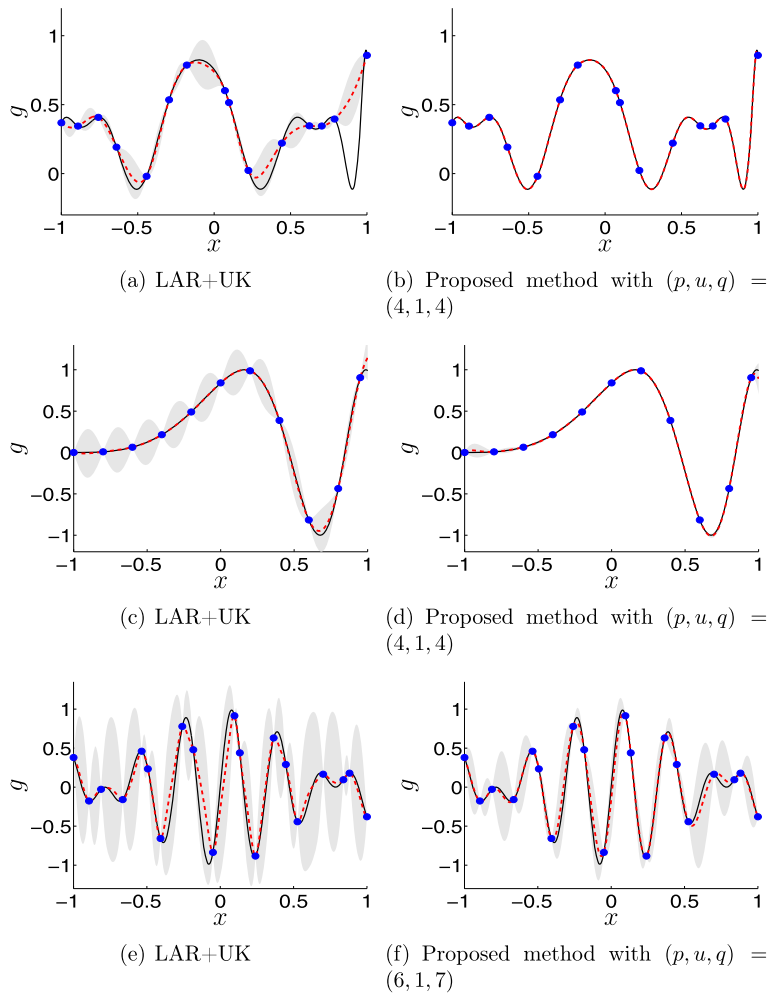
Fig. 2. Efficiency of the proposed method to predict in an unobserved point the value of $g(x) = P_2 \circ P_1(x)$ with $N = 15$ (first row), $g(x) = \sin((x+1)^3)$ with $N = 11$ (second row) and $g(x) = \sin(20x)\cos(2x)$ with $N = 20$ (third row). In each figure, the black solid line is the evolution of the quantity of interest, g, with respect to x, the blue points are the positions of the available observations of g, the red dotted line is the prediction of g based on an optimized LAR + UK approach (left column) or based on the proposed approach associated with optimized values of p, u and q (right column). The grey areas correspond to the 95% confidence region for the prediction. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

$$\boldsymbol{g}^{(2)}(\boldsymbol{x}) = (\cos(\pi x_1 + 1), \cos(\pi x_2 + 2), \dots, \cos(\pi x_6 + 6)). \tag{60}$$

In the same manner than in Section 4.1, Fig. 3 compares the evolution of errors $\varepsilon^2_{\text{NEST}}$ and $\varepsilon^2_{\text{LAR+UK}}$ with respect to N. As for the one-dimensional cases, it can be noticed in these figures that, for the considered examples, introducing a nested structure for the polynomial trend can allow us to make the L^2 error decrease by several orders of magnitude, especially when N is low. Moreover, these figures emphasize the interest of optimizing the values of truncation parameter u when dealing with multidimensional input spaces.

As explained in Section 3.1, the values of p, q and u that were obtained from the convergence analyses can give many information about the structure of the quantity of interest. For the first example, the values $p = 2$, $u = 2$ and $q > 2$ were most of the time chosen, which is coherent with the fact that $g^{2D}(x_1, x_2)$ is just the product of two functions that depend on x_1 and x_2 only. Hence, a particular attention has to be paid to the modeling of each input, rather than on the modeling of the dependence structure.

In the same manner, for the second example, most of the convergence analyses lead us to $u = 3$ and $p < q$, which also shows that the modeling of each input seems to be more important than the characterization of the relation between these modified inputs.

At last, for the third quantity of interest, which is a highly oscillating function in dimension $d = 6$, the convergence analyses seemed to encourage the values of p and q that lead to the highest product $p \times q$ (before over-fitting). This means that, for this example, it is interesting to approximate quantity of interest g by a complex polynomial representation that is characterized by a small number of parameters.
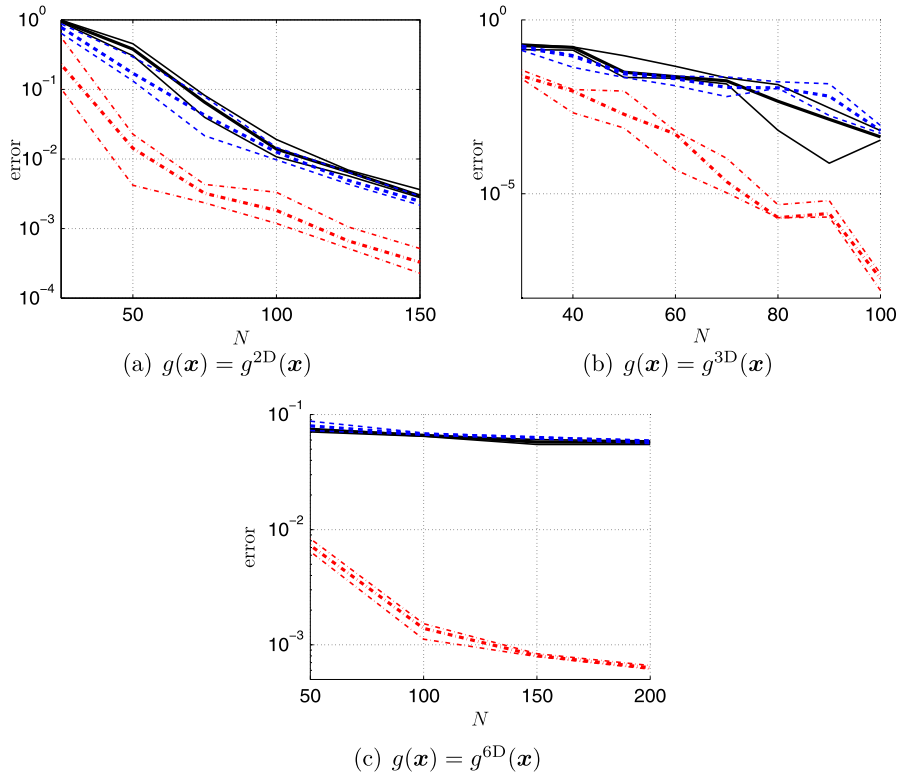
(a) $g(\boldsymbol{x}) = g^{2D}(\boldsymbol{x})$

(b) $g(\boldsymbol{x}) = g^{3D}(\boldsymbol{x})$

(c) $g(\boldsymbol{x}) = g^{6D}(\boldsymbol{x})$

Fig. 3. Evolution of the normalized L^2 errors with respect to N, the number of code evaluations. To be more representative, for each value of N, the LAR + UK and the proposed approaches have been repeated 10 times on randomly chosen learning sets. The curves correspond to the values of the 25% (thin line), the 50% (thick line) and the 75% (thin line) quantiles of the errors associated with these 10 repetitions. Solid black line: evolution of the error associated with the LAR + UK approach, $\varepsilon^2_{\text{LAR+UK}}$. Blue dotted line: evolution of the error associated with the proposed approach, $\varepsilon^2_{\text{NEST}}$, with $u = 1$. Red dashed line: evolution of the error associated with the proposed approach, $\varepsilon^2_{\text{NEST}}$, with $1 \leq u \leq d$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

4.3. Relevance of the LOO error

As presented in Section 3, when the maximal information about g is a set of code evaluations, error $\left\| g - \widehat{g}^{\text{nest}} \right\|_{L^2}$ can be evaluated by its LOO approximation, ε_{LOO}. In order to reduce the computational cost associated with the evaluation of ε_{LOO}, two alternative estimations of error $\left\| g - \widehat{g}^{\text{nest}} \right\|_{L^2}$, $\widehat{\varepsilon}_{\text{LOO}}$ and $\widetilde{\varepsilon}_{\text{LOO}}$, have been proposed. In order to underline the relevance of these two LOO errors, Fig. 4 compares these three errors in the case when $N = 100$ and g is the Ishigami function, for which expression is given by Eq. (57) (the same kinds of results would have been obtained for other values of N and other expressions of g). In this figure, it can thus be noticed that both approximations $\widehat{\varepsilon}_{\text{LOO}}$ and $\widetilde{\varepsilon}_{\text{LOO}}$ are very close to $\left\| g - \widehat{g}^{\text{nest}} \right\|_{L^2}$. In general, approximation $\widetilde{\varepsilon}_{\text{LOO}}$ is more conservative, in the sense that there are less chances that it underestimates $\left\| g - \widehat{g}^{\text{nest}} \right\|_{L^2}$. However, as explained in Section 3, introducing a linearization around $\boldsymbol{a}^\star$ reduces the risk of being too dependent on $\boldsymbol{a}^\star$, which explains the fact that only small differences can be noticed between $\widehat{\varepsilon}_{\text{LOO}}$ and $\widetilde{\varepsilon}_{\text{LOO}}$.

5. Conclusions

One of the main objectives of this paper is to propose an alternative parametrization of the polynomial trends for the Gaussian process regression. This parametrization, which is based on the composition of two polynomials, allows us to span high dimensional polynomial spaces with a reduced number of parameters. Hence, it has been shown on a series of examples that this approach can be very useful, especially when confronted to the modeling of complex functions with very little information.

However, identifying relevant values for these parameters is not easy. In this work, these parameters are identified from a two-steps approach. First, their maximum-likelihood estimates are searched from the solving of a non-convex optimization problem. An iterative algorithm has been proposed to approximate the solutions of this problem. Then, a linearization around these values is carried out, in order to find back the usual formalism of GPR, and to minimize the sensitivity of the results to these values.

In spite of all these adaptations, when the input dimension becomes high ($d > 10$), and when a lot of code evaluations are available ($N > 100d$), it appears that the value of q is often equal to 1. Such a value for q corresponds to the "LAR + UK"
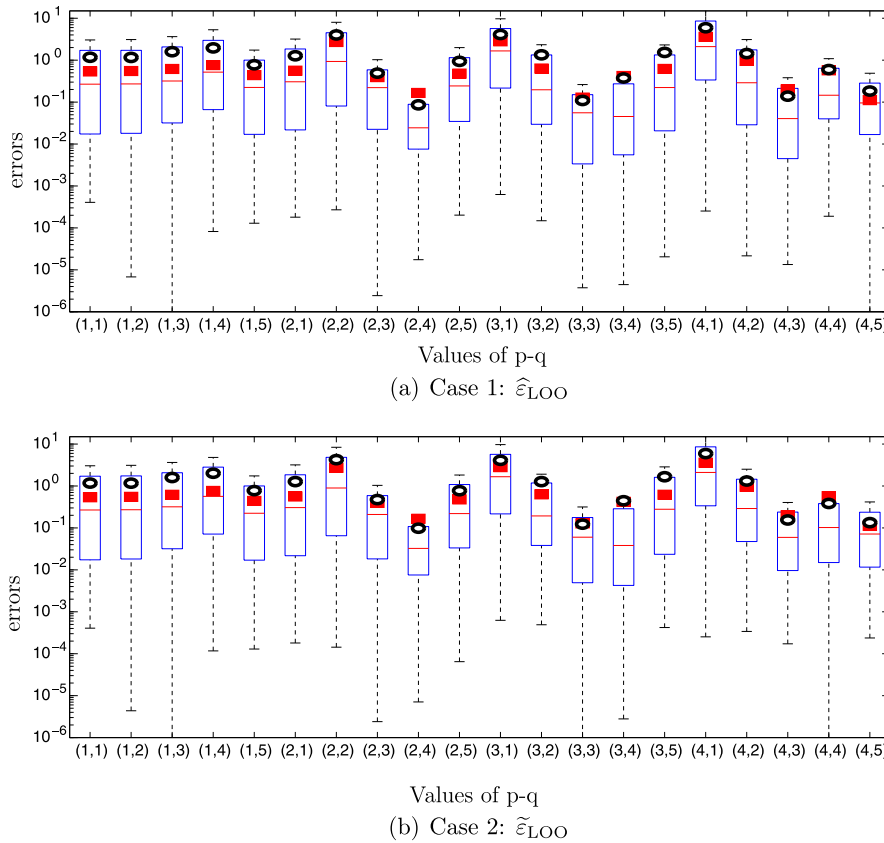
(a) Case 1: $\widehat{\varepsilon}_{\mathrm{LOO}}$

(b) Case 2: $\widetilde{\varepsilon}_{\mathrm{LOO}}$

Fig. 4. Comparisons between error $\left\|g - \widehat{g}^{\mathrm{nest}}\right\|_{L^2}$ and its LOO approximations $\widehat{\varepsilon}_{\mathrm{LOO}}$ and $\widetilde{\varepsilon}_{\mathrm{LOO}}$ for the modeling of the Ishigami function from $N = 100$ code evaluations, for $u = d$, $1 \le p \le 4$ and $1 \le q \le 5$. Red squares: the true values of $\left\|g - \widehat{g}^{\mathrm{nest}}\right\|_{L^2}$. Black circles: the approximated values. In each case, the box-plots correspond to the distributions of $(\widehat{e}_n^2, \ 1 \le n \le N)$ and $(\widetilde{e}_n^2, \ 1 \le n \le N)$, whose expressions are given by Eqs. (46) and (50).

configuration, which would mean that, in that case, the nested structure is not necessary. This can be due to the fact that the considered quantity of interest does not present a nested structure, or to the fact that the numerical complexity of the optimization problems associated with the nested representation is too high. Increasing the robustness of the proposed iterative algorithm, as well as proposing more efficient methods to solve the introduced optimization problems are thus possible extensions of the present work.

Trying to increase the sparsity of the proposed nested representation could also be a good idea, especially to enable the proposed method to deal with systems with higher values of d. Coupling the proposed nested representation to dedicated penalization techniques seems promising for future work.

References

[1] M. Arnst, R. Ghanem, C. Soize, Identification of bayesian posteriors for coefficients of chaos expansions, J. Comput. Phys. 229 (9) (2010) 3134–3154.
[2] J. Bect, D. Ginsbourger, L. Li, V. Picheny, E. Vasquez, Sequential design of computer experiments for the estimation of a probability of failure, Stat. Comput. 22 (2012).
[3] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan, J.M. McFarland, Efficient global reliability analysis for non-linear implicit performance functions, AIAA J. 46 (10) (2008).
[4] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable gaussian process: towards an efficient, fully bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013).
[5] G. Blatman, B. Sudret, Adaptative sparse polynomial chaos expansion based on least angle regression, J. Comput. Phys. 230 (2011).
[6] R. Brent, Algorithms for Minimization Without Derivatives, Prentice-Hall, Englewood Cliffs N.J., 1973.
[7] S. Das, R. Ghanem, S. Finette, Polynomial chaos representation of spatio-temporal random field from experimental measurements, J. Comput. Phys. 228 (2009) 8726–8751.
[8] O. Dubrule, Cross validation of kriging in a unique neighborhood, Math. Geol. 15 (6) (1983) 687–699.
[9] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) 407–499.
[10] R. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, rev. ed., Dover Publications, New York, 2003.
[11] R. Ghanem, P.D. Spanos, Polynomial chaos in stochastic finite elements, J. Appl. Mech. 57 (1990) 197–202.
[12] R.B. Gramacy, H.K.H. Lee, Cases for the nugget in modeling computer experiments, Stat. Comput. 22 (2012) 713–722.
[13] M.S. Handcock, M.L. Stein, A bayesian analysis of kriging, Technometrics 35 (1993) 403–4010.
[14] T. Hastie, R. Tibshirani, Friedman, Elements of Statistical Learning, Springer, New York, 2002.

[15] J.D. Jakeman, M.S. Eldred, K. Sargsyan, Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection, J. Comput. Phys. 289 (2015) 18–34.
[16] V. Roshan Joseph, Ying Hung, Agus Sudjianto, Blind kriging: a new method for developing metamodels, J. Mech. Des. 130 (3) (2008) 031102.
[17] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc. 63 (2001) 425–464.
[18] P. Kersaudy, B. Sudret, N. Varsier, O. Picon, A new surrogate modeling technique combining kriging and polynomial chaos expansions – application to uncertainty analysis in computational dosimetry, J. Comput. Phys. 286 (2015) 103–117.
[19] Katerina Konakli, Bruno Sudret, Polynomial meta-models with canonical low-rank approximations: numerical insights and comparison to sparse polynomial chaos expansions, J. Comput. Phys. 321 (2016) 1144–1169.
[20] H.J. Kushner, A new method of locating the maximal point of an arbitrary multipeak curve in the presence of noise, J. Basic Eng. 86 (1964) 97–106.
[21] O.P. Le Maître, O.M. Knio, Spectral Methods for Uncertainty Quantification, Springer, 2010.
[22] R.G. Miller, The jackknife – a review, Biometrika 61 (1974) 1–15.
[23] A. Nouy, Proper generalized decomposition and separated representations for the numerical solution of high dimensional stochastic problems, Arch. Comput. Methods Eng. 17 (2010) 403–434.
[24] J. Oakley, A. O'Hagan, Bayesian inference for the uncertainty distribution of computer model outputs, Biometrika 89 (4) (2002) 769–784.
[25] A. O'Hagan, Curvefitting and optimal design for prediction, J. R. Stat. Soc., Ser. B, Methodol. 40 (1) (1978) 1–42.
[26] E. Parzen, An approach to time series analysis, Ann. Math. Stat. 32 (1962) 951–989.
[27] G. Perrin, C. Soize, D. Duhamel, C. Funfschilling, Identification of polynomial chaos representations in high dimension from a set of realizations, SIAM J. Sci. Comput. 34 (6) (2012) 2917–2945.
[28] G. Perrin, C. Soize, D. Duhamel, C. Funfschilling, A posteriori error and optimal reduced basis for stochastic processes defined by a finite set of realizations, SIAM/ASA J. Uncertain. Quantificat. 2 (2014) 745–762.
[29] R.T. Rubinstein, D.P. Kroese, Simulation and the Monte Carlo Method, John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.
[30] J. Sacks, W. Welch, T. Mitchell, H. Wynn, Design and analysis of computer experiments, Stat. Sci. 4 (1989) 409–435.
[31] T.J. Santner, B.J. Williams, W.I. Notz, The Design and Analysis of Computer Experiments, Springer, New York, 2003.
[32] C. Soize, R. Ghanem, Physical systems with random uncertainties: chaos representations with arbitrary probability measure, SIAM J. Sci. Comput. 26 (2004) 395–410.
[33] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc., Ser. B, Stat. Methodol. 58 (1) (1989) 267–288.
[34] C.K. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, vol. 2, MIT Press, Boston, 2006.