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

Guillaume Perrin, Christian Soize, Josselin Garnier, Marque-Pucheu Sophie

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*

Abstract

The role of simulation has kept 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 can be 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-based 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, which mean and covariance functions have to be identified from the available code evaluations. In this context, this work proposes an innovative parameterization 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.

Key words:
Computer experiments, Gaussian Processes, nested polynomial trend, Bayesian framework

Email addresses: guillaume.perrin2@cea.fr (G. Perrin)

1. Introduction

In spite of always increasing computational resources, the numerical cost of many codes to simulate complex mechanical 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 mathematical functions that are cheap to evaluate. To be more precise, 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 $(\cdot, \cdot)$, and the associated norm $\|\cdot\|_{L^2}$, such that for all u and v in $L^2(\mathcal{D}_d, \mathbb{R})$,

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

Let $\mathcal{S}$ be a physical system, which response depends on a d-dimensional input vector $\boldsymbol{x} = (x_1, \ldots, x_d)$, and which 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 $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ 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:

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

In that context, the Gaussian process regression (GPR) method, or kriging, plays a major role [19, 15, 20, 21]. 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 for which precision can be *a posteriori* quantified. Such a method considers performance function $g := \{g(\boldsymbol{x}),\ \boldsymbol{x} \in \mathcal{D}_d\}$ as a sample path of a real-valued Gaussian stochastic process $Y := \{Y(\boldsymbol{x}, \omega),\ \boldsymbol{x} \in \mathcal{D}_d,\ \omega \in \Omega\}$, 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:

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

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

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

such that $\mathbb{P}(\ \cdot\ \mid \mathcal{F}_N)$ and $\mathbb{E}[\ \cdot\ \mid \mathcal{F}_N]$ will be used to denote the conditional probability and conditional mathematical expectation respectively. The mean function of Y is then supposed to be parameterized by a chosen M-dimensional vector of functions in $L^2(\mathcal{D}_d, \mathbb{R})$, $\boldsymbol{f} = (f_1, \ldots, f_M)$, and a M-dimensional real vector to be determined, $\boldsymbol{\beta}$, such that:

$$\mu := \langle \boldsymbol{f}, \boldsymbol{\beta} \rangle, \quad Y \mid \boldsymbol{\beta} \ \sim\ \mathrm{GP}(\langle \boldsymbol{f}, \boldsymbol{\beta} \rangle, C), \tag{5}$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^M$. This hypothesis, which was first introduced in time series analysis [17] and in optimization [12], is widely used in computational sciences, as it allows dealing with the conditional probability and expectation, while leading to very interesting results in terms of computer code prediction. Indeed, gathering in the matrices $[F]$ and $[C]$ the evaluations of $\boldsymbol{f}$ and C at the available points $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$,

$$\begin{cases} [F] := [\boldsymbol{f}(\boldsymbol{x}^{(1)}) \ \cdots \ \boldsymbol{f}(\boldsymbol{x}^{(N)})]^T, \\ [C]_{ij} := C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}), \quad 1 \leq i, j \leq N, \end{cases} \tag{6}$$

it can be shown [16] that if the matrix $[C]$ is invertible, then:

$$Y \mid \boldsymbol{\beta}, \mathcal{F}_N \ \sim\ \mathrm{GP}(\mu_N, C_N), \tag{7}$$

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

$$\begin{cases} \mu_N(\boldsymbol{x}) := \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\beta} \rangle + \boldsymbol{r}(\boldsymbol{x})^T [C]^{-1} \left(\mathbb{Y} - [F]\boldsymbol{\beta}\right), \\ C_N(\boldsymbol{x}, \boldsymbol{x}') := C(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{r}(\boldsymbol{x})^T [C]^{-1} \boldsymbol{r}(\boldsymbol{x}'), \\ \boldsymbol{r}(\boldsymbol{x}) := \left(C(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \ldots, C(\boldsymbol{x}, \boldsymbol{x}^{(N)})\right). \end{cases} \tag{8}$$

In addition, if matrix $[F]^T [C]^{-1} [F]$ is also invertible, and if we suppose that the vector $\boldsymbol{\beta}$, which is *a priori* unknown, can be modeled by a random vector that is uniformly distributed on $\mathbb{R}^M$ (improper prior distribution), it comes:

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

$$\begin{cases} \mu_{\text{UK}}(\boldsymbol{x}) := \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\beta}^\star \rangle + \boldsymbol{r}(\boldsymbol{x})^T [C]^{-1} \left(\mathbb{Y} - [F] \boldsymbol{\beta}^\star \right), \\ C_{\text{UK}}(\boldsymbol{x}, \boldsymbol{x}') := C_N(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{u}(\boldsymbol{x})^T ([F]^T [C]^{-1} [F])^{-1} \boldsymbol{u}(\boldsymbol{x}'), \\ \boldsymbol{\beta}^\star := ([F]^T [C]^{-1} [F])^{-1} [F]^T [C]^{-1} \mathbb{Y}, \\ \boldsymbol{u}(\boldsymbol{x}) := [F]^T [C]^{-1} \boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}). \end{cases} \tag{10}$$

Under this formalism, the best prediction of g in a non-computed point $\boldsymbol{x}$ is given by the mean value of $(Y(\boldsymbol{x}) \mid \mathcal{F}_N)$, $\mu_{\text{UK}}(\boldsymbol{x})$, whereas $C_{\text{UK}}(\boldsymbol{x}, \boldsymbol{x})$ quantifies the trust we can put in that prediction.

Therefore, the relevance of $\mu_{\text{UK}}(\boldsymbol{x})$ to predict $g(\boldsymbol{x})$ is conditioned by the choice of function C and vector $\boldsymbol{f}$. Without information about the regularity of g, function C is generally chosen as an element of the Matern-5/2 class, such that for all $\boldsymbol{x}, \boldsymbol{x}'$ in $\mathcal{D}_d$:

$$C(\boldsymbol{x}, \boldsymbol{x}') := \sigma^2 \prod_{i=1}^{d} (1 + \sqrt{5} h_i + 5 h_i^2 / 3) \exp(-\sqrt{5} h_i), \quad h_i = |x_i - x_i'| / \ell_i. \tag{11}$$

In this case, covariance function C is characterized by a vector of hyper-parameters, $\boldsymbol{\Theta} = (\sigma, \ell_1, \ldots, \ell_d)$, which values have also to be conditioned by $\mathcal{F}_N$ and $\boldsymbol{f}$. 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 [18]) to approximate the posterior distribution of $(Y \mid \mathcal{F}_N)$ [8, 10, 3]. In this work, we will adopt an alternative approach, which consists in conditioning all the former results by the maximum likelihood estimate of $\boldsymbol{\Theta}$. This method, which is generally called *plug-in* approach, has indeed 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 [2, 1]. Finally, given these hypotheses, the only thing that can be done to minimize $\| g - \mu_{\text{UK}} \|_{L^2}$ is working on the choice of vector $\boldsymbol{f}$. Once again, without information about g, polynomials are generally chosen for $\boldsymbol{f}$. Indeed, the set $\left\{ m_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathbb{N}^d \right\}$, with

$$m_{\boldsymbol{\alpha}}(\boldsymbol{x}) := x_1^{\alpha_1} \times \cdots \times x_d^{\alpha_d}, \quad \boldsymbol{x} \in \mathcal{D}_d, \tag{12}$$

defines a basis of $L^2(\mathcal{D}_d, \mathbb{R})$. For a given value of M, characterizing $\boldsymbol{f}$ amounts therefore at identifying the best M-dimensional subset of $\left\{ m_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathbb{N}^d \right\}$ 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 $\left\{m_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathbb{N}^d\right\}$. Different truncation schemes have thus been proposed to choose such a relevant subset, which are mostly based on the assumption that the most influential elements of $\left\{m_{\boldsymbol{\alpha}},\ \boldsymbol{\alpha} \in \mathbb{N}^d\right\}$ correspond to the elements of lowest total polynomial order. Denoting by r the maximal polynomial order of the projection basis, we can introduce:

$$\mathcal{P}(r,d) := \left\{ m_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^d,\ \sum_{i=1}^{d} |\alpha_i| \leq r \right\}. \tag{13}$$

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:

$$\mathcal{C}(r,d) = (d+r)!/(d! \times r!). \tag{14}$$

For $M \leq \mathcal{C}(r,d)$, vector $\boldsymbol{f}$ is then generally searched using a penalization technique, such as the Least Angle Regression (LAR) method [9, 7, 4], which allows disregarding insignificant terms. Such an approach will be referred as "LAR+UK" approach in the following, where "UK" stands for universal kriging, and corresponds to the former plug-in approach (more details about the combination of the universal kriging and the Least Angle Regression can be found in [11]).

However, when N, the number of code evaluations, is low compared to the complexity of g, such approaches are limited by the fact that only low values of M, the dimension of the projection family, can be considered to avoid extra-fitting. In order to be able to deal with higher values of M, without increasing the number of unknown parameters to be identified, this work proposes a new parameterization of the polynomial trend, which is based on a nested structure. This paper shows therefore an alternative approach to construct relevant predictors for complex systems, when only very limited information is available.

The outline of this work is as follows. First, Section 2 presents the theoretical framework we propose for the definition of a Gaussian-process regression with a nested polynomial trend. The practical implementation of this

method is then detailed in Section 3. At last, the efficiency of the method is illustrated on a series of analytic examples in Section 4.

2. Nested polynomial trend

2.1. General framework

As presented in Introduction, in this work, we are interested in identifying the best predictor of g in any non-computed point $\boldsymbol{x}$ in $\mathcal{D}_d$, when the maximal information is a fixed number of code evaluations. To this end, for p, q, u in $\mathbb{N}^*$, let $[a]$ and $\boldsymbol{b}$ be respectively a $(u \times \mathcal{C}(q,d))$-dimensional matrix and a $\mathcal{C}(p,u)$-dimensional vector, such that the mean value of the stochastic process Y associated with performance g, μ, can be parameterized by:

$$\mu(\boldsymbol{x}; [a], \boldsymbol{b}) := \left\langle \boldsymbol{m}^{(p,u)}([a]\boldsymbol{m}^{(q,d)}(\boldsymbol{x})), \boldsymbol{b} \right\rangle, \quad \boldsymbol{x} \in \mathcal{D}_d, \tag{15}$$

where numbers $\mathcal{C}(p,u)$ and $\mathcal{C}(q,d)$ are defined by Eq. (14), and where $\boldsymbol{m}^{(p,u)}$ and $\boldsymbol{m}^{(q,d)}$ are the vector-valued functions that gather all the elements of $\mathcal{P}(p,u)$ and $\mathcal{P}(q,d)$ respectively. The elements of these two vectors are supposed to be sorted in an increasing total polynomial order, such that:

$$m_1^{(p,u)} \;=\; m_1^{(q,d)} = 1. \tag{16}$$

By construction, it can be noticed that:

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

such that, for all $u \geq 1$, function $\boldsymbol{x} \mapsto \mu(\boldsymbol{x}; [a], \boldsymbol{b})$ is in Span $\{\mathcal{P}(p \times q, d)\}$, while being characterized by $\mathcal{C}(p,u) + u \times \mathcal{C}(q,d)$ parameters. In order to focus on the minimal parameterization of this nested structure, the two following constraints are moreover introduced:

$$\begin{cases} [a]_{i1} = 0, \\ \displaystyle\sum_{k=1}^{\mathcal{C}(q,d)} [a]_{ik}^2 = 1, \end{cases} \quad 1 \leq i \leq u, \tag{18}$$

and for $2 \leq k \leq \mathcal{C}(q,d)$, at most one component $([a]_{1k}, \ldots, [a]_{uk})$ is supposed to be non-zero. These $\mathcal{C}(q,d) - 1$ non-zero coefficients of $[a]$ can then be gathered in a vector $\boldsymbol{a}$, such that:

$$[a]\boldsymbol{m}^{(q,d)}(\boldsymbol{x}) = [P^{(q,d)}(\boldsymbol{x})]\boldsymbol{a}. \tag{19}$$

A Bayesian formalism is then adopted to identify $[a]$ and $\boldsymbol{b}$. Function g is supposed to be a particular realization of the Gaussian stochastic process Y, which statistical properties are given by:

$$Y \mid \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta} \ \sim \ \mathrm{GP}(\mu(\boldsymbol{a}, \boldsymbol{b}), C(\boldsymbol{\Theta})), \tag{20}$$

$$\mu(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{b}) := \left\langle \boldsymbol{m}^{(p,u)}([P^{(q,d)}(\boldsymbol{x})]\boldsymbol{a}), \boldsymbol{b} \right\rangle, \quad \boldsymbol{x} \in \mathcal{D}_d, \tag{21}$$

where $\boldsymbol{\Theta}$ gathers the $d+1$ parameters of the Matern-5/2 covariance C defined by Eq. (11).

2.2. Linearization of the nested polynomial trend

The proposed nested polynomial trend is however strongly non linear with respect to $\boldsymbol{a}$. This prevents us from using the convenient formula given by Eq. (10). In order to circumvent this problem, let $(\boldsymbol{a}^\star, \boldsymbol{b}^\star, \boldsymbol{\Theta}^\star)$ be the solution of the following log-likelihood maximization problem:

$$(\boldsymbol{a}^\star, \boldsymbol{b}^\star, \boldsymbol{\Theta}^\star) = \arg \max_{(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\Theta}) \in \mathcal{S}^{\mathrm{adm}}} -\frac{1}{2} \left\{ \begin{array}{l} N \log(2\pi) + \log(\det([C(\boldsymbol{\Theta})])) \\ + (\mathbb{Y} - \boldsymbol{M}(\boldsymbol{a}, \boldsymbol{b}))^T [C(\boldsymbol{\Theta})]^{-1} (\mathbb{Y} - \boldsymbol{M}(\boldsymbol{a}, \boldsymbol{b})) \end{array} \right\}, \tag{22}$$

$$\boldsymbol{M}(\boldsymbol{a}, \boldsymbol{b}) := \big(\mu(\boldsymbol{x}^{(1)}; \boldsymbol{a}, \boldsymbol{b}), \ldots, \mu(\boldsymbol{x}^{(N)}; \boldsymbol{a}, \boldsymbol{b})\big) = [\mathbb{M}(\boldsymbol{a})]\boldsymbol{b}, \tag{23}$$

$$[\mathbb{M}(\boldsymbol{a})] := \left[\boldsymbol{m}^{(p,u)}([P^{(q,d)}(\boldsymbol{x}^{(1)})]\boldsymbol{a}) \ \cdots \ \boldsymbol{m}^{(p,u)}([P^{(q,d)}(\boldsymbol{x}^{(N)})]\boldsymbol{a})\right]^T, \tag{24}$$

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 $[C(\boldsymbol{\Theta})]$ and $[\mathbb{M}(\boldsymbol{a})]^T [C(\boldsymbol{\Theta})]^{-1} [\mathbb{M}(\boldsymbol{a})]$ are invertible.

By construction, function $\boldsymbol{b} \mapsto \mu(\cdot;\cdot,\boldsymbol{b})$ is linear, such that the linearization of mean function $\mu(\boldsymbol{x};\boldsymbol{a},\boldsymbol{b})$ in the vicinity of $\boldsymbol{a}^\star$ and $\boldsymbol{b}^\star$ is:

$$\mu(\boldsymbol{x};\boldsymbol{a},\boldsymbol{b}) \approx \left\langle \left(\boldsymbol{h}^{(1)}(\boldsymbol{x};\boldsymbol{a}^\star,\boldsymbol{b}^\star), \boldsymbol{h}^{(2)}(\boldsymbol{x};\boldsymbol{a}^\star)\right), (\boldsymbol{a}-\boldsymbol{a}^\star,\boldsymbol{b})\right\rangle, \tag{25}$$

$$\boldsymbol{h}^{(1)}(\boldsymbol{x};\boldsymbol{a}^\star,\boldsymbol{b}^\star) = [P^{(q,d)}(\boldsymbol{x})]^T [D([P^{(q,d)}(\boldsymbol{x})]\boldsymbol{a}^\star)]^T \boldsymbol{b}^\star, \tag{26}$$

$$\boldsymbol{h}^{(2)}(\boldsymbol{x};\boldsymbol{a}^\star) = \boldsymbol{m}^{(p,u)}([P^{(q,d)}(\boldsymbol{x})]\boldsymbol{a}^\star), \tag{27}$$

$$[D(\boldsymbol{z})] := \left[\frac{\partial \boldsymbol{m}^{(p,u)}}{\partial \boldsymbol{z}}(\boldsymbol{z})\right], \quad \boldsymbol{z} \in \mathbb{R}^u, \tag{28}$$

$$\left[\frac{\partial \boldsymbol{m}^{(p,u)}}{\partial \boldsymbol{z}}(\boldsymbol{z})\right]_{kj} := \frac{\partial m_k^{(p,u)}}{\partial z_j}(\boldsymbol{z}), \quad 1 \leq j \leq u, \quad 1 \leq k \leq \mathcal{C}(p,u), \ \boldsymbol{z} \in \mathbb{R}^u. \tag{29}$$

Let us now denote by $\boldsymbol{\beta} := (\boldsymbol{a}-\boldsymbol{a}^\star, \boldsymbol{b})$ the new vector of parameters to be determined, 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. Hence, conditioned by the values of $\boldsymbol{a}^\star$, $\boldsymbol{b}^\star$ and $\boldsymbol{\Theta}^\star$, the formalism introduced in Eq. (5) is found back:

$$Y \mid \boldsymbol{\beta} \ \sim \ \mathrm{GP}(\langle \boldsymbol{f}, \boldsymbol{\beta} \rangle, C), \tag{30}$$

such that the mean value of $(Y \mid \mathcal{F}_N)$ can be calculated analytically to compute the predictor of g. At last, to avoid extra-fitting, classical penalization techniques can also be used to consider only the most influential components of $\boldsymbol{f}$ in the modeling of g.

Remark on the linearization.

As it will be shown in Section 3, the maximization problem defined by Eq. (22) is not easy. Hence, by introducing vector $\boldsymbol{\beta}$, one additional interest of the proposed linearization is to make the final predictor be less sensitive to the solutions of this problem.

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

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 $u \in \{1, d\}$.

2.3. Comments on the proposed parameterization

Proposing such a nested parameterization of the mean function of Y is motivated by two main reasons.

- First, for $d > 1$, it 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 polynomial order q (considering different values of q for each input could eventually be done to optimize such a two-scales modeling). Hence, analyzing the optimal values of p, u and q can give us 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 multiscale.

- Second, this approach is very attractive in terms of dimension reduction, as it can be seen in Table 1. Indeed, 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 will be seen in Section 4, this is particularly interesting for the modeling of complex phenomena with very limited information.

3. Practical implementation

As presented in Section 2, for given values of $\boldsymbol{x}$ and $\boldsymbol{b}$, function $\boldsymbol{a} \mapsto \mu(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{b})$ is strongly non linear. Hence, one key step of the proposed for-

mulation is the solving of the optimization problem given by Eq. (22). In addition, when trying to define optimized predictors with finite information, we have to be careful to avoid extra-fitting. Methods to quickly evaluate error $\|g - \widehat{g}\|_{L^2}$ for any $\widehat{g}$ in $L^2(\mathcal{D}_d, \mathbb{R})$ are therefore needed. This section is therefore divided in two main sections, which respectively deal with these two issues.

3.1. Maximization of the likelihood

In this section, for given values of N, u, p and q, we are interested in identifying the solution $(\boldsymbol{a}^\star, \boldsymbol{b}^\star, \boldsymbol{\Theta}^\star)$ of the log-likelihood maximization problem given by Eq. (22). To this end, denoting by L the function such that for all $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta})$ belonging to the admissible set $\mathcal{S}^{\text{adm}}$,

$$L(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta}) = \log(\det([C(\boldsymbol{\Theta})])) + (\mathbb{Y} - \boldsymbol{M}(\boldsymbol{a}, \boldsymbol{b}))^T [C(\boldsymbol{\Theta})]^{-1} (\mathbb{Y} - \boldsymbol{M}(\boldsymbol{a}, \boldsymbol{b})), \tag{31}$$

it is interesting to notice that for all $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta})$ in $\mathcal{S}^{\text{adm}}$,

$$L(\boldsymbol{a}, \boldsymbol{b}^{\text{LS}}(\boldsymbol{a}, \boldsymbol{\Theta}), \boldsymbol{\Theta}) \leq L(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\Theta}), \tag{32}$$

where:

$$\boldsymbol{b}^{\text{LS}}(\boldsymbol{a}, \boldsymbol{\Theta}) = \left([\mathbb{M}(\boldsymbol{a})]^T [C(\boldsymbol{\Theta})]^{-1} [\mathbb{M}(\boldsymbol{a})]\right)^{-1} [\mathbb{M}(\boldsymbol{a})]^T [C(\boldsymbol{\Theta})] \mathbb{Y}, \tag{33}$$

and matrix $[\mathbb{M}(\boldsymbol{a})]$ is given by Eq. (24). It comes:

$$\begin{cases} (\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star) = \arg\min_{(\boldsymbol{a}, \boldsymbol{\Theta})} \mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}), \\ \boldsymbol{b}^\star = \left([\mathbb{M}(\boldsymbol{a}^\star)]^T [C(\boldsymbol{\Theta}^\star)]^{-1} [\mathbb{M}(\boldsymbol{a}^\star)]\right)^{-1} [\mathbb{M}(\boldsymbol{a}^\star)]^T [C(\boldsymbol{\Theta}^\star)] \mathbb{Y}, \end{cases} \tag{34}$$

$$\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}) := L(\boldsymbol{a}, \boldsymbol{b}^{\text{LS}}(\boldsymbol{a}, \boldsymbol{\Theta}), \boldsymbol{\Theta}). \tag{35}$$

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.

Initialization: $L_1 = 0$, $L_2 = +\infty$, $\boldsymbol{a}^* = (1, \ldots, 1)/\left\|(1, \ldots, 1)\right\|$;
while $|L_2 - L_1| > \varepsilon$ **do**
  $L_1 = L_2$;
  $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{a}^*, \boldsymbol{\Theta})$;
  $\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} \mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}^*)$;
  $L_2 = \min(L_2, \mathcal{L}(\boldsymbol{a}^*, \boldsymbol{\Theta}^*))$;
end
$\boldsymbol{a}^\star \approx \boldsymbol{a}^*$, $\boldsymbol{\Theta}^\star \approx \boldsymbol{\Theta}^*$.

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

3.2. Error evaluation

According to Section 2 and 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:

$$\begin{aligned}\widehat{g}^{\text{nest}}(\boldsymbol{x}) = &\langle \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{a}^\star, \boldsymbol{\Theta}^\star), \boldsymbol{\beta}^\star(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)\rangle \\ &+ \boldsymbol{r}(\boldsymbol{x}; \boldsymbol{\Theta}^\star)^T [C(\boldsymbol{\Theta}^\star)]^{-1} \left(\mathbb{Y} - [F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]\boldsymbol{\beta}^\star(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)\right),\end{aligned} \tag{36}$$

$$\boldsymbol{\beta}^\star(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star) := ([F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]^T [C(\boldsymbol{\Theta}^\star)]^{-1} [F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)])^{-1} [F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]^T [C(\boldsymbol{\Theta}^\star)]^{-1} \mathbb{Y}, \tag{37}$$

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. (34), under the additional condition that the matrix $[F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]^T [C(\boldsymbol{\Theta}^\star)]^{-1} [F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]$ is invertible,
- vector $\mathbb{Y}$ 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 $\left(\boldsymbol{h}^{(1)}(\cdot; \boldsymbol{a}^\star, \boldsymbol{b}^{\text{LS}}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)), \boldsymbol{h}^{(2)}(\cdot; \boldsymbol{a}^\star)\right)$, which have been identified from a LAR procedure,

- $[F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)] := [\boldsymbol{f}(\boldsymbol{x}^{(1)}; \boldsymbol{a}^\star, \boldsymbol{\Theta}^\star) \ \cdots \ \boldsymbol{f}(\boldsymbol{x}^{(N)}; \boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]$ gathers the evaluations of $\boldsymbol{f}(\cdot; \boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)$ at the available code evaluations,
- and for all $1 \leq n, m \leq N$, $[C(\boldsymbol{\Theta}^\star)]_{nm} = C(\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(m)})$ and $r_n(\boldsymbol{x}; \boldsymbol{\Theta}^\star) = C(\boldsymbol{x}, \boldsymbol{x}^{(n)})$, with C the Matern-5/2 covariance function of parameters $\boldsymbol{\Theta}^\star$.

As presented in Section 1, the relevance of such a predictor is assessed from the computation of the L^2 error $\|g - \widehat{g}^{\text{nest}}\|_{L_2}$. Function g being only known through a limited number of evaluations, classical Leave-One-Out (LOO) techniques [13, 4] can therefore be introduced to approximate such a norm:

$$\left\|g - \widehat{g}^{\text{nest}}\right\|_{L_2}^2 \approx \epsilon_{\text{LOO}}^2 := \frac{1}{N} \sum_{n=1}^{N} \left(g(\boldsymbol{x}^{(n)}) - \widehat{g}_{-n}^{\text{nest}}(\boldsymbol{x}^{(n)})\right)^2, \tag{38}$$

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 only using the $N-1$ evaluations of the code in

$$\mathbb{X}^{(-n)} := \begin{cases} \left\{\boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\right\} \text{ if } n = 1, \\ \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N-1)}\right\} \text{ if } n = N, \\ \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n-1)}, \boldsymbol{x}^{(n+1)}, \ldots, \boldsymbol{x}^{(N)}\right\} \text{ otherwise.} \end{cases} \tag{39}$$

In order to reduce the computational cost associated with the evaluation of ϵ_{LOO}^2, it is interesting to notice (see [6] for further details) that for all $1 \leq n \leq N$:

$$g(\boldsymbol{x}^{(n)}) - \widehat{g}_{-n}^{\text{nest}}(\boldsymbol{x}^{(n)}) = \frac{([\widehat{C}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]\mathbb{Y})_n}{[\widehat{C}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]_{nn}}, \tag{40}$$

$$[\widehat{C}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)] = [C(\boldsymbol{\Theta}^\star)]^{-1} - [C(\boldsymbol{\Theta}^\star)]^{-1}[\mathbb{F}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)][C(\boldsymbol{\Theta}^\star)]^{-1}, \tag{41}$$

$$[\mathbb{F}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)] := [F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]([F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]^T[C(\boldsymbol{\Theta}^\star)]^{-1}[F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)])^{-1}[F(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]^T. \tag{42}$$

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

$$\epsilon^2_{\mathrm{LOO}} \approx \widehat{\epsilon}^2_{\mathrm{LOO}} := \frac{1}{N}\sum_{n=1}^{N} \widehat{e}^2_n, \quad \widehat{e}^2_n := \left\{ \frac{([\widehat{C}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]\mathbb{Y})_n}{[\widehat{C}(\boldsymbol{a}^\star, \boldsymbol{\Theta}^\star)]_{nn}} \right\}^2. \tag{43}$$

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 $\boldsymbol{a}$, $\boldsymbol{\Theta}$, $1 \leq n \leq N$:

$$\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta}) = \mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta}) + \frac{([\widetilde{C}(\boldsymbol{a}, \boldsymbol{\Theta})]\mathbb{Y})^2_n}{[\widetilde{C}(\boldsymbol{a}, \boldsymbol{\Theta})]_{nn}}, \tag{44}$$

$$[\widetilde{C}(\boldsymbol{a}, \boldsymbol{\Theta})] = [C(\boldsymbol{\Theta})]^{-1}\left\{[I] - [\mathbb{M}(\boldsymbol{a})]([\mathbb{M}(\boldsymbol{a})]^T[C(\boldsymbol{\Theta})]^{-1}[\mathbb{M}(\boldsymbol{a})])^{-1}[\mathbb{M}(\boldsymbol{a})]^T[C(\boldsymbol{\Theta})]^{-1}\right\}, \tag{45}$$

where $\mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta})$ is the evaluation of function $\mathcal{L}(\boldsymbol{a}, \boldsymbol{\Theta})$ based on the $N-1$ evaluations of the code in $\mathbb{X}^{(-n)}$ only. Hence, in the optimization process leading us to the identification of $\boldsymbol{a}^\star$ and $\boldsymbol{\Theta}^\star$, let $\{(\boldsymbol{a}_i, \boldsymbol{\Theta}_i),\ 1 \leq i \leq N_{\mathrm{test}}\}$ 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 $\boldsymbol{a}^\star_{-n}$ and $\boldsymbol{\Theta}^\star_{-n}$ respectively, and which are given by:

$$(\boldsymbol{a}^\star_{-n}, \boldsymbol{\Theta}^\star_{-n}) = \arg \max_{(\boldsymbol{a},\boldsymbol{\Theta}) \in \{(\boldsymbol{a}_i,\boldsymbol{\Theta}_i),\ 1 \leq i \leq N_{\mathrm{test}}\}} \mathcal{L}_{-n}(\boldsymbol{a}, \boldsymbol{\Theta}). \tag{46}$$

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

$$\left\|g - \widehat{g}^{\mathrm{nest}}\right\|^2_{L_2} \approx \widetilde{\epsilon}^2_{\mathrm{LOO}} := \frac{1}{N}\sum_{n=1}^{N} \widetilde{e}^2_n, \quad \widetilde{e}^2_n := \left\{ \frac{([\widehat{C}(\boldsymbol{a}^\star_{-n}, \boldsymbol{\Theta}^\star_{-n})]\mathbb{Y})_n}{[\widehat{C}(\boldsymbol{a}^\star_{-n}, \boldsymbol{\Theta}^\star_{-n})]_{nn}} \right\}^2. \tag{47}$$

3.3. 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. To do so, maximal values for p, q and u are *a priori* chosen, and the values for these parameters will be chosen in order to minimize error $\widetilde{\epsilon}^2_{\mathrm{LOO}}$. 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.

Remark on the roles of p, q and u in the modeling of g.

As presented in Section 2.3, 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 parameterizes 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, and depends only linearly on u. Hence, increasing the values of p and u does not really increase the search set for the identification of $\boldsymbol{a}^\star$, but makes the relation between $\boldsymbol{a}$ and $\mathcal{L}(\boldsymbol{a},\boldsymbol{\Theta})$ much more complex.

4. Applications

To illustrate the advantages of the nested structure presented in Sections 2 and 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 1. In that prospect, 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 about g, 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:

$$\varepsilon^2_{\text{NEST}} = \left\| g - \widehat{g}^{\text{nest}} \right\|^2_{L^2} / \left\| g \right\|^2_{L^2}, \tag{48}$$

$$\varepsilon^2_{\text{LAR+UK}} = \left\| g - \widehat{g}^{\text{LAR+UK}} \right\|^2_{L^2} / \left\| g \right\|^2_{L^2}. \tag{49}$$

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.

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 $[-1, 1]$:

$$\begin{cases} P_1(x) = \displaystyle\sum_{i=1}^{5} c_i^{(1)} x^{i-1}, & \boldsymbol{c}^{(1)} = \dfrac{(0, -0.03, 0.5, -0.4, -0.5)}{\sqrt{0.03^2 + 0.5^2 + 0.4^2 + 0.5^2}}, \\ P_2(x) = \displaystyle\sum_{i=1}^{5} c_i^{(2)} x^{i-1}, & \boldsymbol{c}^{(2)} = (-0.1, 0.2, 0.7, -0.2, -0.2). \end{cases} \tag{50}$$

For each case, Figure 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:

$$0 \leq p^{\text{LAR+UK}} \leq 20, \quad 0 \leq p, q \leq 10, \quad u = 1. \tag{51}$$

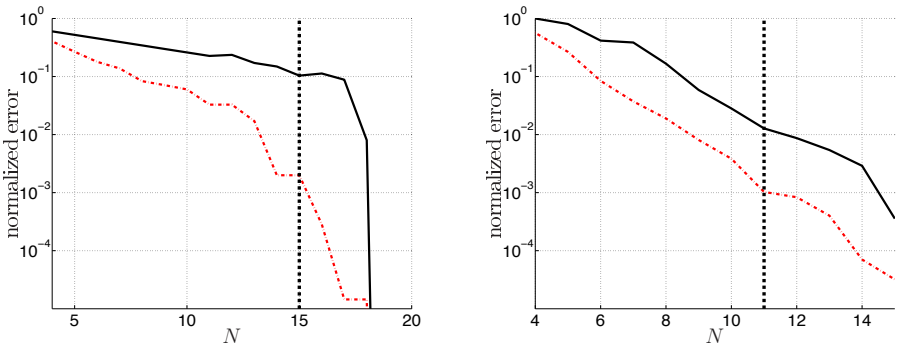
In addition, Figure 2 compares the two approaches in term of prediction for given values of N. In these figures we notice that the proposed method is particularly adapted to the cases when g presents a nested structure or when it is oscillating. This is particularly true when N is small compared to the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

4.2. d>1

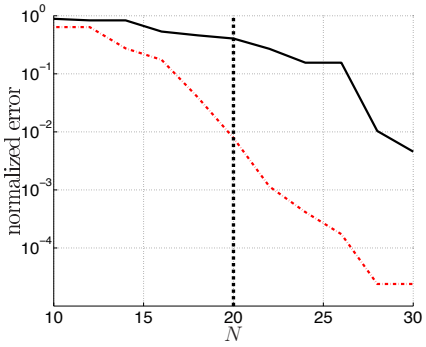
The idea of this section is to show that the tendencies that were noticed in the one-dimensional cases are found back when considering multidimensional input spaces. To this end, let us consider the three following expressions of g, and the associated maximal values for the convergence analyses:

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

$$g : \begin{cases} [-1, 1]^2 & \to & [-1, 1] \\ \boldsymbol{x} & \mapsto & g^{\text{2D}}(\boldsymbol{x}) = (1 - x_1^2)\cos(7x_1) \times (1 - x_2^2)\sin(5x_2) \end{cases}. \tag{52}$$

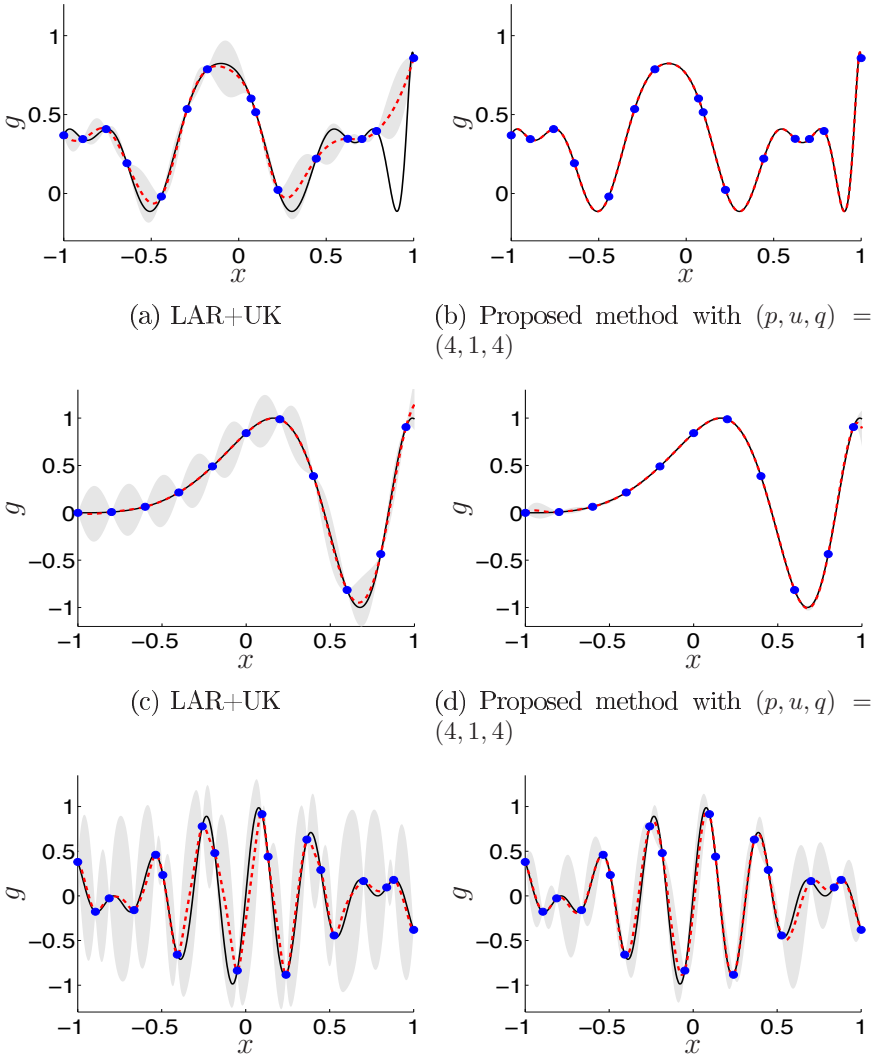


(a) $g(x) = P_2 \circ P_1(x)$ (b) $g(x) = \sin((x+1)^3)$



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

Figure 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 Figure 2 are focused.



(a) LAR+UK

(b) Proposed method with $(p, u, q) = (4, 1, 4)$

(c) LAR+UK

(d) Proposed method with $(p, u, q) = (4, 1, 4)$

(e) LAR+UK

(f) Proposed method with $(p, u, q) = (6, 1, 7)$

Figure 2: Efficiency of the proposed method to predict in a non-computed 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.

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

$$g : \left\{ \begin{array}{ccc} [-\pi, \pi]^3 & \to & \mathbb{R} \\ \boldsymbol{x} = (x_1, x_2, x_3) & \mapsto & g^{\mathrm{3D}}(\boldsymbol{x}) = \sin(x_1) + 7\sin(x_2)^2 + 0.1x_3^4 \sin(x_1) \end{array} \right. . \tag{53}$$

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

$$g : \left\{ \begin{array}{ccc} [-1, 1]^6 & \to & \mathbb{R} \\ \boldsymbol{x} & \mapsto & g^{\mathrm{6D}}(\boldsymbol{x}) = g^{(1)} \circ \boldsymbol{g}^{(2)}(\boldsymbol{x}), \end{array} \right. \tag{54}$$

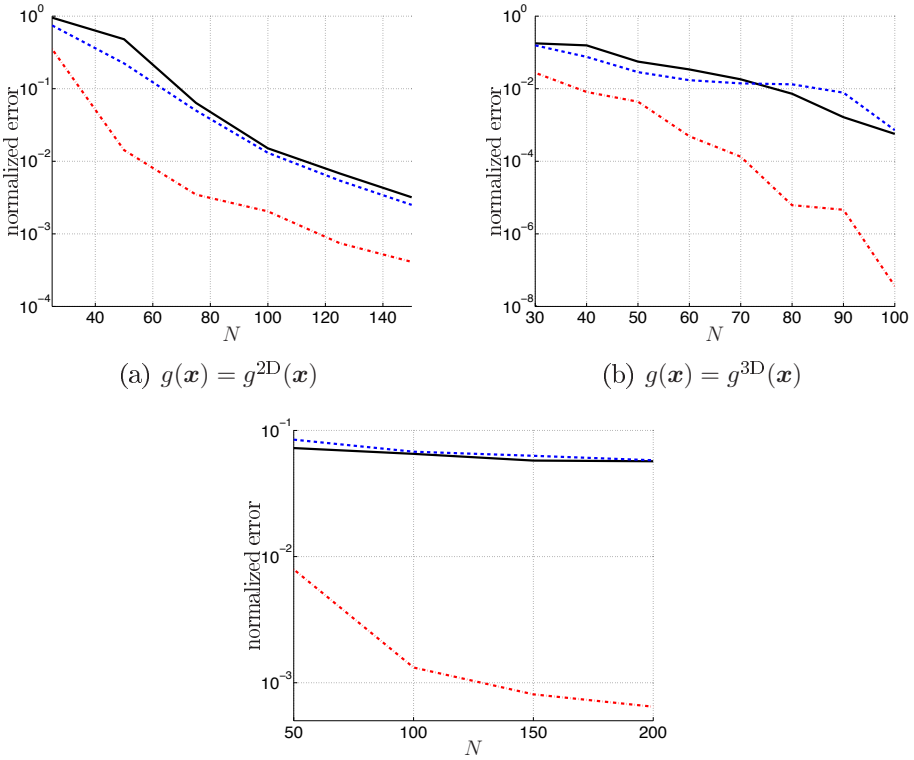
$$g^{(1)}(\boldsymbol{z}) = 0.1 \cos\left(\sum_{i=1}^{6} z_i \right) + \sum_{i=1}^{6} z_i^2, \quad \boldsymbol{z} \in \mathbb{R}^6, \tag{55}$$

$$\boldsymbol{g}^{(2)}(\boldsymbol{x}) = \left(\cos(\pi x_1 + 1), \cos(\pi x_2 + 2), \ldots, \cos(\pi x_6 + 6) \right). \tag{56}$$

In the same manner than in Section 4.1, Figure 3 compares the evolution of errors $\varepsilon^2_{\mathrm{NEST}}$ and $\varepsilon^2_{\mathrm{LAR+UK}}$ 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.

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 $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$ can be evaluated by its LOO approximation, $\varepsilon_{\mathrm{LOO}}$. In order to reduce the computational cost associated with the evaluation of $\varepsilon_{\mathrm{LOO}}$, two alternative estimations of error $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$, $\widehat{\varepsilon}_{\mathrm{LOO}}$ and $\widetilde{\varepsilon}_{\mathrm{LOO}}$, have been proposed. In order to underline the relevance of these two LOO errors, Figure 4 compares these three errors in the case when $N = 100$ and g is the Ishigami function, for which expression is given by Eq. (52) (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 $\widehat{\varepsilon}_{\mathrm{LOO}}$ and $\widetilde{\varepsilon}_{\mathrm{LOO}}$ are very close to $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$. In



(a) $g(\boldsymbol{x}) = g^{2\mathrm{D}}(\boldsymbol{x})$ (b) $g(\boldsymbol{x}) = g^{3\mathrm{D}}(\boldsymbol{x})$

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

Figure 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 mean value of the errors associated with these 10 repetitions. Solid black line: evolution of the error associated with the LAR+UK approach, $\varepsilon^2_{\mathrm{LAR+UK}}$. Blue dotted line: evolution of the error associated with the proposed approach, $\varepsilon^2_{\mathrm{NEST}}$, with $u = 1$. Red dashed line: evolution of the error associated with the proposed approach, $\varepsilon^2_{\mathrm{NEST}}$, with $1 \leq u \leq d$.

general, approximation $\widetilde{\varepsilon}_{\mathrm{LOO}}$ is more conservative, in the sense that there are less chances that it underestimates $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$. However, as explained in the final remark of Section 2.2, 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 $\widehat{\varepsilon}_{\mathrm{LOO}}$ and $\widetilde{\varepsilon}_{\mathrm{LOO}}$.

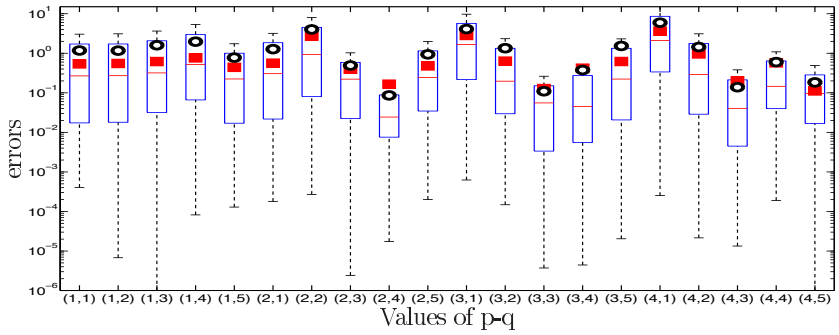
5. Conclusions

One of the main objectives of this paper is to propose an alternative parameterization of the polynomial trends for the Gaussian process-based regression. This parameterization, 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.

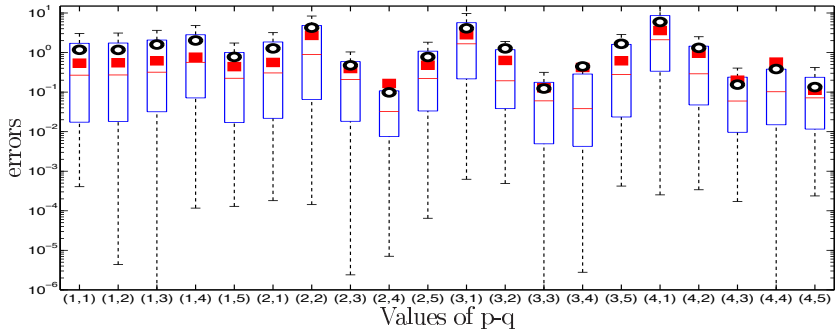
However, identifying relevant values for these parameters is not easy. In this work, these parameters are identified from a two-steps approach. First, their maximum-likelihood estimates are searched from the solving of a non-convex optimization problem. An iterative algorithm has been proposed to approximate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

When the number of code evaluations becomes high, it appears that the proposed approach and the "LAR+UK" approach give similar results (the "LAR+UK" approach being a particular case of the proposed approach). This can be due to the fact that the nested structure can be not necessary when a lot of information about the code is available, or to numerical difficulties in the parameters identification. Increasing the robustness of the proposed iterative algorithm, as well as proposing more efficient methods to solve the introduced optimization problem are possible extensions of the present work.

Furthermore, trying to increase the sparsity of the proposed nested representation could be a good idea, especially to enable the proposed method to deal with systems with high values of d. In that prospect, coupling the proposed approach to low rank approximations [14, 5] seems promising for future work.



(a) Case 1: $\widehat{\varepsilon}_{\mathrm{LOO}}$



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

Figure 4: Comparisons between error $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$ 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 $u = d$, $1 \leq p \leq 4$ and $1 \leq q \leq 5$. Red squares: the true values of $\|g - \widehat{g}^{\mathrm{nest}}\|_{L^2}$. Black circles : the approximated values. In each case, the boxplots correspond to the distributions of $(\widehat{e}_n^2,\ 1 \leq n \leq N)$ and $(\widetilde{e}_n^2,\ 1 \leq n \leq N)$, which expressions are given by Eqs. (42) and (46).