Constrained Gaussian process regression: an adaptive approach for the estimation of hyperparameters and the verification of constraints with high probability

Guillaume Perrin, S. da Veiga

Constrained Gaussian process regression: an adaptive approach for the estimation of hyperparameters and the verification of constraints with high probability

G. Perrin[a], S. Da Veiga[b]

[a] *COSYS, Université Gustave Eiffel, 77420 Champs-sur-Marne, France*
[b] *Safran Tech, Safran SA, Magny-Les-Hameaux, France*

Abstract

This paper focuses on the Gaussian Process regression (GPR) of non-linear functions subject to multiple linear constraints, such as boundedness, monotonicity or convexity. It presents an algorithm allowing to optimize, in a concerted way, the statistical moments of the Gaussian process used for the regression, and the position of a reduced number of points where the constraints are required to hold, such that the constraints are verified in the whole input space, with high probability, at a reasonable computational cost. After having presented the theoretical bases and the numerical implementation of this algorithm, this paper illustrates its efficiency though the analysis of several test functions of increasing dimensions.

Keywords: Gaussian process, machine learning, uncertainty quantification, linear constraints, physics-constrained machine learning

1. Introduction

The conception and the certification of complex systems using simulation are generally based on the evaluation of computer codes in a very high number of input points. In this work, we focus on the analysis of one of these systems, whose properties can be characterized by a vector of d_x continuous parameters, $\boldsymbol{x} = (x_1, \ldots, x_{d_x}) \in \mathbb{X} \subset \mathbb{R}^{d_x}$, and we denote by y the measurable function defined on $\mathbb{X}$ that is used to monitor the good functioning of this system. Function y is considered as the output of a computationally expensive deterministic "black box", in the sense that for every $\boldsymbol{x}$ in $\mathbb{X}$, $y(\boldsymbol{x})$ is unique and it can be calculated using a time consuming computer code.
As each evaluation of y is time consuming, the fine exploration of input space $\mathbb{X}$ cannot be done using y directly, but it is necessary to associate a surrogate model to it, as it is done in [16]. Among these surrogate modeling techniques, the Gaussian process regression (GPR), or kriging, plays a central role, which is due in particular to its capacity to associate in a very natural way a confidence to the predictions it returns [19, 11, 20, 7]. In a classical way, the

Email address: guillaume.perrin@univ-eiffel.fr (G. Perrin)

construction of a GPR model relies on the evaluation of y at a well-chosen set of points in $\mathbb{X}$. Nevertheless, it often happens that the modeler manipulating the code also wants to include one or more a priori knowledge about the behavior of the system in the construction of the GPR. For example, such constraints can be associated with underlying physical phenomena when considering engineering applications.

The motivations for taking these constraints into account are numerous: improvement of prediction capacities, reduction of uncertainties, better explainability of the results, and so on. Although not suitable for taking into account all types of constraints, the GPR formalism offers a very attractive framework for taking into account linear constraints on y, i.e. constraints that can be written in the form $a(\boldsymbol{x}) \leq \mathcal{L}y(\boldsymbol{x}) \leq b(\boldsymbol{x})$, with a, b two given functions and $\mathcal{L}$ a linear operator. This includes boundedness, monotonicity or convexity constraints, but also constraints based on integral operators and partial differential equations. Indeed, if function y is modeled by a Gaussian process, $\mathcal{L}y$ is also Gaussian, and its statistical moments can be explicitly derived from the statistical moments of y.

Several methods for imposing linear constraints on GPs can therefore be found in the literature (see [21] for a survey). Among them, several works strive to ensure the respect of constraints at all points of $\mathbb{X}$ [14, 13]. These approaches are based on a finite dimensional Gaussian approximation associated with a structured discretisation of the input space. These methods show interesting results for examples in 1D and 2D, while being too time consuming for applications in higher dimensions. In order to tackle problems of larger dimensions ($d_x > 2$), it was proposed in [18, 22, 24, 1, 23] to restrict the verification of constraints to a finite set of input points, often called virtual observations. In that case, the respect of the constraints on $\mathbb{X}$ is strongly dependent on the position of these virtual observations and their number. Criteria were therefore defined to optimize their positions and number and make the constraints verified in $\mathbb{X}$ with high probability at a reasonable computational cost (i.e. without having to densely fill $\mathbb{X}$ with virtual observations).

This paper is a continuation of these works, with two directions of improvement. First, it proposes two new criteria for the positioning of virtual observations, for a more global and faster respect of the constraints according to the number of observation points. It then focuses on the consideration of the constraints for the identification of the statistical properties of the Gaussian process used for the regression. Few works have actually addressed this problem which still remains relatively open. Left as a working perspective for most of the previously listed works, the choice of the mean function and especially of the covariance function of this Gaussian process plays however a very important role for the respect of the constraints. Indeed, as we will show in the application part of this paper, the choice of too small correlation lengths may result in the need to introduce a very large number of virtual observations for the respect of constraints with sufficient high probability. Conversely, choosing correlation lengths that are too large may make it almost impossible (in probability) to verify constraints on subsets of $\mathbb{X}$. In other words, the choice of these statistical parameters is intimately linked to the respect of the constraints, and thus strongly depends on the positions of the virtual observations. As the choice of the positions of the virtual observations is itself strongly dependent on the choice of these statistical parameters, it is then proposed

in this paper to identify in a concerted way these statistical parameters and these virtual observations.
The outline of this work is as follows. Section 2 introduces the general framework for carrying out a Gaussian process regression in the presence of inequality constraints. The criteria we propose for the selection of the virtual observations are then described in Section 3. Section 4 deals with the identification of the statistical moments of the Gaussian process used for the regression, and Section 5 describes the algorithm we propose for the verification of constraints with high probability using only a reduced number of virtual observations. Numerical applications are then presented in Section 6, while concluding remarks are given in Section 7.

2. General framework

The formalism of the Gaussian process regression is considered: the quantity of interest y is seen as a sample path of a stochastic process Y defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, which is assumed Gaussian for the sake of tractability: $Y \sim \mathrm{GP}(\mu, C)$, where μ is the mean function and C is the covariance function of Y. Let $\mathcal{X}(N) := \left\{(\boldsymbol{x}^{(n)}, y(\boldsymbol{x}^{(n)})),\ 1 \leq n \leq N\right\}$ be a N-dimensional Design of Experiments (DoE). By conditioning Y by the responses of y in $\mathcal{X}(N)$, we obtain another Gaussian process, which is noted $Y_N := Y \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}) \sim \mathrm{GP}(\mu_N, C_N)$, whose mean and covariance functions can be explicitly derived (see [19, 20] for further details about the expressions):

$$\mu_N(\boldsymbol{x}) = \mu(\boldsymbol{x}) + C(\boldsymbol{x}, \boldsymbol{X})C(\boldsymbol{X}, \boldsymbol{X})^{-1}(\mu(\boldsymbol{X}) - y(\boldsymbol{X})), \quad \boldsymbol{x} \in \mathbb{X}, \tag{1}$$

$$C_N(\boldsymbol{x}, \boldsymbol{x}') = C(\boldsymbol{x}, \boldsymbol{x}') - C(\boldsymbol{x}, \boldsymbol{X})C(\boldsymbol{X}, \boldsymbol{X})^{-1}C(\boldsymbol{X}, \boldsymbol{x}'), \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}. \tag{2}$$

In the former expressions, $\boldsymbol{X} := [\boldsymbol{x}^{(1)} \ \cdots \ \boldsymbol{x}^{(N)}]^T$ is the $(N \times d_x)$-dimensional matrix that gathers the inputs points of $\mathcal{X}(N)$, and for each function f and g defined on $\mathbb{X}$ and $\mathbb{X} \times \mathbb{X}$ respectively, the following notation is adopted

$$(f(\boldsymbol{X}))_n = f(\boldsymbol{x}^{(n)}), \quad (g(\boldsymbol{X}, \boldsymbol{X}))_{nm} = g(\boldsymbol{x}_n, \boldsymbol{x}_m), \quad 1 \leq n, m \leq N. \tag{3}$$

As the time needed to evaluate y in each points of $\mathcal{X}^N$ is supposed to be very high, the value of N is assumed relatively small. Gaussian process Y_N can therefore be used to predict the value of y in any non-observed point of $\mathbb{X}$. In particular, $\mu_N(\boldsymbol{x})$ is the best linear unbiased predictor (BLUP) of $y(\boldsymbol{x})$, while $C_N(\boldsymbol{x}, \boldsymbol{x})$ quantifies the uncertainty associated with this prediction, in the sense that the smaller it is, the more chance there is for $y(\boldsymbol{x})$ and $\mu_N(\boldsymbol{x})$ to be close.
In addition to the observation points, we assume we have access to prior knowledge on some properties of function y, which can be written under the form of a linear operator $\mathcal{L}$ (adopting the same notations than in [1]). For example, the constraints $0 \leq y$, $\ell_2 \leq \partial y/\partial x_i \leq u_2$, $\partial^2 y/\partial x_i \partial x_j \leq u_3$ can be written:

$$(0, \ell_2(\boldsymbol{x}), -\infty) \leq_c \mathcal{L}y(\boldsymbol{x}) \leq_c (+\infty, u_2(\boldsymbol{x}), u_3(\boldsymbol{x})), \ \boldsymbol{x} \in \mathbb{X}, \tag{4}$$

where $\mathcal{L} : y \mapsto \mathcal{L}y := (y, \partial y/\partial x_i, \partial^2 y/\partial x_i \partial x_j)$, and $\leq_c$ stands for the component by component inequality operator, such that for two d-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a} \leq_c \boldsymbol{b}$ is equivalent to $a_1 \leq b_1, \ldots, a_d \leq b_d$. In particular, this includes boundedness, monotonicity or convexity constraints on y.

In the following, we denote by d_c the number of constraints, such that $\mathcal{L}y$ is a function from $\mathbb{R}^{d_x}$ to $\mathbb{R}^{d_c}$, and by $\boldsymbol{\ell}$ and $\boldsymbol{u}$ the vector-valued functions that characterize the lower and upper bounds for $\mathcal{L}y$ (possibly taking infinite values). The Gaussian distribution being stable by linear operations, $\mathcal{L}Y$ is still a Gaussian process, with:

$$\mathbb{E}\left[\mathcal{L}Y(\boldsymbol{x})\right] = \mathcal{L}\mu(\boldsymbol{x}), \quad \mathrm{Cov}(\mathcal{L}Y(\boldsymbol{x}), \mathcal{L}Y(\boldsymbol{x}')) = \mathcal{L}C(\boldsymbol{x}, \boldsymbol{x}')\mathcal{L}^T. \tag{5}$$

Here, the notations $\mathcal{L}C(\boldsymbol{x}, \boldsymbol{x}')$ and $C(\boldsymbol{x}, \boldsymbol{x}')\mathcal{L}^T$ indicate that operator $\mathcal{L}$ is applied as a function of $\boldsymbol{x}$ and $\boldsymbol{x}'$ respectively, so that $\mathrm{Cov}(\mathcal{L}Y(\boldsymbol{x}), \mathcal{L}Y(\boldsymbol{x}'))$ is a $(d_c \times d_c)$-dimensional matrix.

Integrating these constraints on Y in the former GP formalism, the new process $Y_N^c := Y \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}),\ \boldsymbol{\ell} \leq_c \mathcal{L}Y \leq_c \boldsymbol{u}$ seems particularly attractive for the prediction of y. Manipulating Y_N^c is however difficult, if not impossible. Indeed, it is supposed to take into account an infinite number of constraints, and even if Y, $Y(\boldsymbol{X})$ and $\mathcal{L}Y$ are Gaussian, there is no reason for Y_N^c to be still Gaussian once the inequality constraints are applied.

Different approaches were proposed to get back to a problem integrating a finite number of constraints, and therefore circumvent the first difficulty. On the one hand, it was proposed in [14, 13] to approximate Y by its finite-dimensional projection on a tensorized grid of $\mathbb{X}$. In that case, the projection functions are deterministic, the projection coefficients are modeled by (potentially correlated) Gaussian random variables, and the constraints on the entire domain are translated as constraints on the projection coefficients only. However, due to the tensorized structure of the projection functions, the application of this approach is limited to very small values of d_x (generally less than 2).

On the other hand, it was proposed in [22] (and completed in [23]) to impose constraints only at a finite set of virtual observations. In that case, the constraints are not fulfilled on the entire domain, but only with a more or less high probability depending on the number and positions of these virtual observations.

Let $\boldsymbol{Z} := [\boldsymbol{z}^{(1)}; \cdots; \boldsymbol{z}^{(M)}]$ be the $(M \times d_x)$-dimensional matrix gathering the positions of M virtual observations in $\mathbb{X}$, and $\boldsymbol{\alpha} \in \{1, \ldots, d_c\}^M$ be the M-dimensional vector gathering the indices of the constraints that we want to impose in each element of $\boldsymbol{Z}$. For instance, for $1 \leq m \leq M$ and $1 \leq j \leq d_c$, choosing $\alpha_m = j$ amounts at imposing

$$\ell_j(\boldsymbol{z}^{(m)}) \leq (\mathcal{L}y(\boldsymbol{z}^{(m)}))_j \leq u_j(\boldsymbol{z}^{(m)}).$$

Under that formalism, we denote by

$$Y_{N,M}^c := Y \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}),\ \boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z}), \tag{6}$$

the process we would like to consider to predict the value of y in any non-observed point of $\mathbb{X}$, and by

$$\mathcal{L}Y^c_{N,M} := \mathcal{L}Y \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}),\ \boldsymbol{\ell_\alpha}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \leq_c \boldsymbol{u_\alpha}(\boldsymbol{Z}), \tag{7}$$

the process we need to manipulate to verify that the constraints have been correctly taken into account not only at the points in $\boldsymbol{Z}$, but at any point in $\mathbb{X}$. Here, for all function f and g defined on $\mathbb{X}$ and $\mathbb{X}\times\mathbb{X}$, $\boldsymbol{\ell_\alpha}(\boldsymbol{Z})$, $\boldsymbol{u_\alpha}(\boldsymbol{Z})$, $\mathcal{L}_{\boldsymbol{\alpha}} f(\boldsymbol{Z})$ and $\mathcal{L}_{\boldsymbol{\alpha}} g(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}^T_{\boldsymbol{\alpha}}$ are three M-dimensional vectors and a $(M\times M)$-dimensional matrix respectively such that for all $1 \leq m, m' \leq M$:

$$(\mathcal{L}_{\boldsymbol{\alpha}} f(\boldsymbol{Z}))_m = (\mathcal{L}f)_{\alpha_m}(\boldsymbol{z}^{(m)}),\quad (\mathcal{L}_{\boldsymbol{\alpha}} g(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}^T_{\boldsymbol{\alpha}})_{mm'} = (\mathcal{L}g(\boldsymbol{z}^{(m)},\boldsymbol{z}^{(m')})\mathcal{L}^T)_{\alpha_m\alpha_{m'}}, \tag{8}$$

$$(\boldsymbol{\ell_\alpha}(\boldsymbol{Z}))_m := \ell_{\alpha_m}(\boldsymbol{z}^{(m)}),\quad (\boldsymbol{u_\alpha}(\boldsymbol{Z}))_m := u_{\alpha_m}(\boldsymbol{z}^{(m)}). \tag{9}$$

Given these notations, the constraints' probability function p_c is defined by:

$$p_c(\boldsymbol{x}) := \mathbb{P}\left(\boldsymbol{\ell}(\boldsymbol{x}) \leq_c \mathcal{L}Y^c_{N,M}(\boldsymbol{x}) \leq_c \boldsymbol{u}(\boldsymbol{x})\right),\ \boldsymbol{x}\in\mathbb{X}, \tag{10}$$

and, for each $1 \leq j \leq d_c$, the probability $p^j_c(\boldsymbol{x})$ for $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$ to satisfy the j^{th} constraint is given by:

$$p^j_c(\boldsymbol{x}) := \mathbb{P}\left(\ell_j(\boldsymbol{x}) \leq (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j \leq u_j(\boldsymbol{x})\right). \tag{11}$$

To analyze the statistical properties of $Y^c_{N,M}$ and $\mathcal{L}Y^c_{N,M}$, let us first consider the following random vector

$$\boldsymbol{L}(\boldsymbol{Z}) := \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}), \boldsymbol{\ell_\alpha}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \leq_c \boldsymbol{u_\alpha}(\boldsymbol{Z}). \tag{12}$$

By construction, $\boldsymbol{L}(\boldsymbol{Z})$ follows the truncated normal distribution $\mathcal{TN}(\boldsymbol{\mu_L}, \boldsymbol{C_L}, \boldsymbol{\ell}(\boldsymbol{Z}), \boldsymbol{u}(\boldsymbol{Z}))$, with:

$$\boldsymbol{\mu_L} := \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z}) + \mathcal{L}_{\boldsymbol{\alpha}} C(\boldsymbol{Z},\boldsymbol{X})C(\boldsymbol{X},\boldsymbol{X})^{-1}(y(\boldsymbol{X}) - \mu(\boldsymbol{X})), \tag{13}$$

$$\boldsymbol{C_L} := \mathcal{L}_{\boldsymbol{\alpha}} C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}^T_{\boldsymbol{\alpha}} - \mathcal{L}_{\boldsymbol{\alpha}} C(\boldsymbol{Z},\boldsymbol{X})C(\boldsymbol{X},\boldsymbol{X})^{-1}C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}^T_{\boldsymbol{\alpha}}, \tag{14}$$

in the sense that its probability density function (PDF) $f_{\boldsymbol{L}}$ verifies the following proportionality relation:

$$f_{\boldsymbol{L}}(\boldsymbol{v}) \propto 1_{\boldsymbol{\ell}(\boldsymbol{Z})\leq_c \boldsymbol{v} \leq_c \boldsymbol{u}(\boldsymbol{Z})} \exp\left(-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\mu_L})^T \boldsymbol{C_L}^{-1}(\boldsymbol{v}-\boldsymbol{\mu_L})\right),\quad \boldsymbol{v}\in\mathbb{R}^M. \tag{15}$$

Looking at Propositions 1 and 2 (see AppendixA and AppendixB for the proofs), we therefore notice that this vector $\boldsymbol{L}(\boldsymbol{Z})$ plays a major role in the analysis of processes $Y^c_{N,M}$ and $\mathcal{L}Y^c_{N,M}$, as there are affine transforms between the mean and the variance of $Y^c_{N,M}(\boldsymbol{x})$ and $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$ and the mean vector and the covariance matrix of $\boldsymbol{L}(\boldsymbol{Z})$, but also between the realizations of $\boldsymbol{L}(\boldsymbol{Z})$ and the realizations of $Y^c_{N,M}(\boldsymbol{x})$ and $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$. The fact that we can efficiently generate independent realizations of non-Gaussian processes $Y^c_{N,M}$ and $\mathcal{L}Y^c_{N,M}$ plays indeed a central role in the following developments. On the one hand, this will allow the construction

of empirical prediction intervals for the value of y at any point of $\mathbb{X}$. On the other hand, it will allow to identify the sub-domains of $\mathbb{X}$ where the constraints are most likely to be violated, but also to estimate average values over $\mathbb{X}$ of verifying the constraints.

Proposition 1. *For all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}$, if $\mu_{\boldsymbol{\alpha}}(\boldsymbol{Z})$ and $C_{\boldsymbol{\alpha}}(\boldsymbol{Z})$ correspond to the mean vector and the covariance matrix of $\boldsymbol{L}(\boldsymbol{Z})$, we obtain :*

$$\mathbb{E}\left[Y^c_{N,M}(\boldsymbol{x})\right] = \mu(\boldsymbol{x}) + \boldsymbol{a}_1^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{a}_2^T(\boldsymbol{x})(\mu_{\boldsymbol{\alpha}}(\boldsymbol{Z}) - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})), \tag{16}$$

$$\begin{aligned} Var\left(Y^c_{N,M}(\boldsymbol{x})\right) =& C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) \\ & - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) + \boldsymbol{a}_2^T(\boldsymbol{x})C_{\boldsymbol{\alpha}}(\boldsymbol{Z})\boldsymbol{a}_2(\boldsymbol{x})^, \end{aligned} \tag{17}$$

$$\mathbb{E}\left[\mathcal{L}Y^c_{N,M}(\boldsymbol{x})\right] = \mathcal{L}\mu(\boldsymbol{x}) + \boldsymbol{A}_3^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{A}_4^T(\boldsymbol{x})(\mu_{\boldsymbol{\alpha}}(\boldsymbol{Z}) - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})), \tag{18}$$

$$\begin{aligned} Cov\left(\mathcal{L}Y^c_{N,M}(\boldsymbol{x}), \mathcal{L}Y^c_{N,M}(\boldsymbol{x}')\right) =& \mathcal{L}C(\boldsymbol{x},\boldsymbol{x}')\mathcal{L}^T - \boldsymbol{A}_3^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}')\mathcal{L}^T \\ & - \boldsymbol{A}_4^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}')\mathcal{L}^T + \boldsymbol{A}_4^T(\boldsymbol{x})C_{\boldsymbol{\alpha}}(\boldsymbol{Z})\boldsymbol{A}_4(\boldsymbol{x}')^, \end{aligned} \tag{19}$$

with $\boldsymbol{a}_1(\boldsymbol{x}) \in \mathbb{R}^N$, $\boldsymbol{a}_2(\boldsymbol{x}) \in \mathbb{R}^M$ the two vectors and $\boldsymbol{A}_3(\boldsymbol{x})$, $\boldsymbol{A}_4(\boldsymbol{x})$ the two matrices that verify:

$$\begin{pmatrix} \boldsymbol{a}_1(\boldsymbol{x}) \\ \boldsymbol{a}_2(\boldsymbol{x}) \end{pmatrix} = \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} C(\boldsymbol{X},\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) \end{pmatrix}, \tag{20}$$

$$\begin{pmatrix} \boldsymbol{A}_3(\boldsymbol{x}) \\ \boldsymbol{A}_4(\boldsymbol{x}) \end{pmatrix} = \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} C(\boldsymbol{X},\boldsymbol{x})\mathcal{L} \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x})\mathcal{L} \end{pmatrix}. \tag{21}$$

Proposition 2. *Let $\boldsymbol{l}^{(1)}, \ldots, \boldsymbol{l}^{(Q)}$ be Q independent realizations of $\boldsymbol{L}(\boldsymbol{Z})$, $\xi^{(1)}, \ldots, \xi^{(Q)}$ be Q independent realizations of a centered Gaussian random value of variance equal to 1, and $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(Q)}$ be Q independent realizations of a d_c-dimensional centered Gaussian random vector whose covariance matrix is the identity matrix. Then, for each $\boldsymbol{x} \in \mathbb{X}$ and each $1 \leq q \leq Q$,*

$$\begin{aligned} & \mu(\boldsymbol{x}) + \boldsymbol{a}_1^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{a}_2^T(\boldsymbol{x})(\boldsymbol{l}^{(q)} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})) \\ & \quad + \sqrt{C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x})}\ \xi^{(q)}, \end{aligned} \tag{22}$$

is an independent realization of $Y^c_{N,M}(\boldsymbol{x})$, and

$$\begin{aligned} & \mathcal{L}\mu(\boldsymbol{x}) + \boldsymbol{A}_3(\boldsymbol{x})^T(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{A}_4(\boldsymbol{x})^T(\boldsymbol{l}^{(q)} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})) \\ & \quad + \left(\mathcal{L}C(\boldsymbol{x},\boldsymbol{x})\mathcal{L}^T - \boldsymbol{A}_3(\boldsymbol{x})^T C(\boldsymbol{X},\boldsymbol{x})\mathcal{L}^T - \boldsymbol{A}_4^T\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x})\mathcal{L}^T\right)^{1/2}\ \boldsymbol{\zeta}^{(q)}, \end{aligned} \tag{23}$$

is an independent realization of $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$, where for any symmetric square matrix $\boldsymbol{R}$, $\boldsymbol{R}^{1/2}$ is a matrix such that $\boldsymbol{R}^{1/2}(\boldsymbol{R}^{1/2})^T = \boldsymbol{R}$.

Generating realizations of $\boldsymbol{L}(\boldsymbol{Z})$ (and therefore of $Y^c_{N,M}(\boldsymbol{x})$ and $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$) is however challenging. Of course, using rejection techniques, that is generating samples from the unconstrained normal distribution $\mathcal{N}(\boldsymbol{\mu_L}, \boldsymbol{C_L})$ and keeping the ones that verify the constraints, would be the most natural method to obtain such realizations. But the more constraint points will be considered, and therefore the more M will increase, the lower the acceptance rate is likely to be, and the more inefficient the method will be. In that case, alternative methods have to be employed, such as the method based on the minimax tilting proposed by [4], which proves to be particularly efficient for the generation of realizations of truncated Gaussian vectors with dimensions smaller than 200, with acceptance probabilities up to 10^{-100}. For higher dimensions, methods based on Gibbs sampling [12] could also be used, but the convergence of such methods is likely to require a very significant numerical cost.

Remarks.

- Looking at Proposition 2, it is important to notice that the same Q iid realizations of $\boldsymbol{L}(\boldsymbol{Z})$ can be used to get Q iid realizations of $Y^c_{N,M}(\boldsymbol{x})$ and $\mathcal{L}Y^c_{N,M}(\boldsymbol{x})$ in any value of $\boldsymbol{x}$, and therefore to predict the value of $y(\boldsymbol{x})$ and estimate $p_c(\boldsymbol{x})$ or $p_c^j(\boldsymbol{x})$ in each $\boldsymbol{x}$.
- If $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(P)}$ are $P \geq 1$ elements of $\mathbb{X}$, Proposition 2 is easily generalized to the generation of realizations of the vector $(Y^c_{N,M}(\boldsymbol{x}^{(1)}), \dots, Y^c_{N,M}(\boldsymbol{x}^{(P)}))$.

3. Selection of the virtual observations

Predictor $\mathbb{E}\left[Y^c_{N,M}(\boldsymbol{x})\right]$ of $y(\boldsymbol{x})$ depends on two sets of points: the N observation points of y gathered in $\boldsymbol{X}$, and the M virtual observations associated with the constraints gathered in $\boldsymbol{Z}$. While the observation points are generally imposed, it is possible to choose the number and the position of the virtual observations as they do not require any code evaluations. Different strategies can be proposed to choose these points. Naively, in order to allow a verification of the constraints over the whole input domain, these observation points can be chosen as uniformly as possible in $\mathbb{X}$. For this, one can then rely on several experiment design works [6, 2, 5, 10, 17]. This approach, which will be refered as "space-filling" (SF) approach in the following, is however clearly sub-optimal. Indeed, these points are supposed to allow a maximisation of the constraints' probability function p_c defined by Eq. (10). Thus, adding points where p_c is large is of little interest. However, searching directly for all of these points to maximise the minimal value of p_c over $\mathbb{X}$ is generally far too difficult, and greedy approaches are often preferred. For instance, following [1, 23], given a set of M virtual observations already chosen, the $M+1$ virtual observation $\boldsymbol{x}^\star$ associated with the $j_\star^{\text{th}}$ constraint can be chosen as the point with the best chance of not respecting the constraints, that is the solution of the following optimization problem:

$$(\boldsymbol{x}^\star, j_\star) \in \arg \max_{\boldsymbol{x}\in\mathbb{X},\ 1\leq j\leq d_c} c_1(\boldsymbol{x}, j), \quad c_1(\boldsymbol{x}, j) := 1 - p_c^j(\boldsymbol{x}). \tag{24}$$

However, it can be noticed that such a pointwise strategy does not take into account in its selection criteria the fact that the new evaluation point will bring additional information

in its neighbourhood, nor integrate the fact that a constraint is strongly or slightly not respected. This can result in an unintended accumulation of virtual points in the same area. For example, if we are interested in a function that is increasing with respect to one of its parameters, and that this function is almost constant over a certain interval, whatever the number of virtual observations that we place in this zone, the probability of not respecting the monotonicity constraint will always be close to 50% at any point in this zone. In order to propose a better treatment of the constraints on the whole input domain, we can then propose to focus on the following enrichment criterion :

$$(\boldsymbol{x}^\star, j_\star) \in \arg \max_{\boldsymbol{x}\in\mathbb{X},\ 1\leq j\leq d_c} c_2(\boldsymbol{x}, j), \tag{25}$$

$$c_2(\boldsymbol{x}, j) := \mathbb{E}\left[\left(\ell_j(\boldsymbol{x}) - (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j\right)^+ + \left((\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j - u_j(\boldsymbol{x})\right)^+\right], \tag{26}$$

which can be seen as an adaptation of the well-known expected improvement (EI) selection criterion [25] for the selection of virtual observations. In Eq. (26), for each z in $\mathbb{R}$, $(z)^+ := \max(0, z)$, and we have used the convention $-\infty \times \Phi(-\infty) = -\infty(1 - \Phi(+\infty)) = 0$ to simplify notations. Thus, for two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ such that $p^j_c(\boldsymbol{x})$ and $p^j_c(\boldsymbol{x}')$ are close, the criterion defined by Eq. (26) allows us to favor the point associated with the strongest non-respect of the constraints. Of course, weights that depend on j could be added to the expressions provided in Eqs. (24) and (26) in the event that one constraint is to be favoured over another.

Finally, to better take into account the impact of the addition of a new virtual observation on its neighborhood, the criteria c_1 and c_2 defined by Eqs. (24) and Eqs. (26) can be replaced by the integrated criteria c_1^{int} and c_2^{int} as follows:

$$c_1^{\text{int}}(\boldsymbol{x}, j) := \sum_{j'=1}^{d_c} \int_{\mathbb{X}} c_1^{\boldsymbol{x},j}(\boldsymbol{x}', j')d\boldsymbol{x}', \quad c_2^{\text{int}}(\boldsymbol{x}, j) := \sum_{j'=1}^{d_c} \int_{\mathbb{X}} c_2^{\boldsymbol{x},j}(\boldsymbol{x}', j')d\boldsymbol{x}',$$

where the criteria $c_1^{\boldsymbol{x},j}$ and $c_2^{\boldsymbol{x},j}$ respectively correspond to the criteria c_1 and c_2 assuming that the j^{th} constraint has been imposed at virtual observation $\boldsymbol{x}$. Hence, criteria c_1^{int} or c_2^{int} can be seen as average probabilities of non-respect of the constraints knowing that a new observation has been added in $\boldsymbol{x}$. This explains that these two criteria need now to be minimized, while we wanted to maximise the criteria c_1 and c_2 proposed in Eqs. (24) and (26).

Remarks on the practical solving of the optimization problems. The maximization of criteria c_1 and c_2, and the minimization of criteria c_1^{int} and c_2^{int} have been introduced in their continuous form. In the following, the solving of these problems will however be based on discrete approximations of these optimization problems. Focusing first on criteria c_1 or c_2, the new virtual observation $\boldsymbol{x}^\star$ and the new constraint $j_\star$ will be searched as:

$$(\boldsymbol{x}^\star, j_\star) \in \arg \max_{\boldsymbol{x}\in\mathcal{S}(n),\ 1\leq j\leq d_c} c_k(\boldsymbol{x}, j), \tag{27}$$

where k is equal to 1 or 2, $\mathcal{S}(n)$ gathers $n \gg 1$ points chosen (randomly or not) in $\mathbb{X}$, and where we remind that the sampling procedure of [4] allows us to evaluate c_1 or c_2 in a very high number of points of $\mathbb{X}$ at a reasonable computational cost. Indeed, if $\boldsymbol{l}^{(1)}, \ldots, \boldsymbol{l}^{(Q)}$ denote Q iid realizations of $\boldsymbol{L}(\boldsymbol{Z})$ such that:

$$(\mathcal{L}Y(\boldsymbol{x}))_j \mid \boldsymbol{L}(\boldsymbol{Z}) = \boldsymbol{l}^{(q)}, Y(\boldsymbol{X}) = y(\boldsymbol{X}) \ \sim \ \mathcal{N}\left(m_j^{(q)}(\boldsymbol{x}), (\sigma_j^{(q)}(\boldsymbol{x}))^2\right), \tag{28}$$

we deduce the following empirical estimations for $c_1(\boldsymbol{x}, j)$ and $c_2(\boldsymbol{x}, j)$:

$$\begin{aligned}
1 - c_1(\boldsymbol{x}, j) &= \mathbb{P}\left(\ell_j(\boldsymbol{x}) \leq_c (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j \leq_c u_j(\boldsymbol{x})\right) \\
&= \mathbb{E}\left[\mathbb{P}\left(\ell_j(\boldsymbol{x}) \leq_c (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j \leq_c u_j(\boldsymbol{x}) \mid \boldsymbol{L}(\boldsymbol{Z})\right)\right] \\
&\approx \frac{1}{Q}\sum_{q=1}^{Q} \mathbb{P}\left(\ell_j(\boldsymbol{x}) \leq_c (\mathcal{L}Y(\boldsymbol{x}))_j \leq_c u_j(\boldsymbol{x}) | \boldsymbol{L}(\boldsymbol{Z}) = \boldsymbol{l}^{(q)}, Y(\boldsymbol{X}) = y(\boldsymbol{X})\right) \\
&\approx \frac{1}{Q}\sum_{q=1}^{Q} 1 - \Phi\left(\frac{u_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right) + \Phi\left(\frac{\ell_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right),
\end{aligned} \tag{29}$$

$$\begin{aligned}
c_2(\boldsymbol{x}, j) &= \mathbb{E}\left[\left(\ell_j(\boldsymbol{x}) - (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j\right)^+ + \left((\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j - u_j(\boldsymbol{x})\right)^+\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\ell_j(\boldsymbol{x}) - (\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j\right)^+ + \left((\mathcal{L}Y^c_{N,M}(\boldsymbol{x}))_j - u_j(\boldsymbol{x})\right)^+ \mid \boldsymbol{L}(\boldsymbol{Z})\right]\right] \\
&\approx \frac{1}{Q}\sum_{q=1}^{Q} \sigma_j^{(q)}(\boldsymbol{x})\left(\phi\left(\frac{\ell_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right) + \phi\left(\frac{u_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right)\right) \\
&\quad + (\ell_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x}))\Phi\left(\frac{\ell_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right) + (m_j^{(q)}(\boldsymbol{x}) - u_j(\boldsymbol{x}))\left(1 - \Phi\left(\frac{u_j(\boldsymbol{x}) - m_j^{(q)}(\boldsymbol{x})}{\sigma_j^{(q)}(\boldsymbol{x})}\right)\right),
\end{aligned} \tag{30}$$

where ϕ and Φ are respectively the probability density function (PDF) and the cumulative density function (CDF) of a centered Gaussian random variable of variance 1.
As for the minimization of criteria c_1^{int} and c_2^{int}, the couple $(\boldsymbol{x}^\star, j_\star)$ will be chosen as the solution of:

$$(\boldsymbol{x}^\star, j_\star) \in \arg \min_{\boldsymbol{x} \in \mathcal{S}(n), 1 \leq j \leq d_c} \sum_{1 \leq j' \leq d_c, \boldsymbol{x}' \in \mathcal{S}(n), \ (\boldsymbol{x}', j') \neq (\boldsymbol{x}, j)} c_k^{\boldsymbol{x}, j}(\boldsymbol{x}', j'), \ k = 1 \text{ or } 2. \tag{31}$$

The evaluation of criterion $c_k^{\boldsymbol{x},j}(\boldsymbol{x}', j')$ requires a little more attention than that of criterion $c_k(\boldsymbol{x}, j)$, as it supposes that a $(M+1)^{\text{th}}$ constraint is imposed in $\boldsymbol{x}$. However, denoting by $\widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(n)}$ the elements of $\mathcal{S}(n)$, it is interesting to notice that, once again, the distribution of the random vector $(\mathcal{L}Y^c_{N,M}(\widetilde{\boldsymbol{x}}^{(1)}), \ldots, \mathcal{L}Y^c_{N,M}(\widetilde{\boldsymbol{x}}^{(n)})) \mid \boldsymbol{L}(\boldsymbol{Z})$ is Gaussian, such that once Q iid realizations of $\boldsymbol{L}(\boldsymbol{Z})$ have been generated, the evaluation of $c_k^{\boldsymbol{x},j}(\boldsymbol{x}', j')$ for each $\boldsymbol{x} \neq \boldsymbol{x}'$ will only be based on the generation of one realization of Q independent one-dimensional truncated normal Gaussian random variables, which is relatively easy and quick to do.

4. Estimation of hyperparameters

A key ingredient in the prediction of y in each non-observed value of $\boldsymbol{x}$ is the choice of the mean function μ and probably even more importantly of the covariance function C of Gaussian process Y. For the sake of tractability, parametric representations can be chosen for these two functions. For instance, μ can be written as a weighted sum of chosen functions of $\boldsymbol{x}$,

$$\mu(\boldsymbol{x}) := \sum_{k=1}^{K} \beta_k h_k(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{h}(\boldsymbol{x}), \tag{32}$$

with $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_K)$ and $\boldsymbol{h}(\boldsymbol{x}) := (h_1(\boldsymbol{x}), \ldots, h_K(\boldsymbol{x}))$ a vector of K functions such that $\mathcal{L}h_k$ exists. In the same manner, C can be chosen among a standard parametric class of covariance functions, such as the square exponential or the Matern families (see [20, 19] for more details about these families). Let R be a correlation function defined on $\mathbb{X} \times \mathbb{X}$ such that $\mathcal{L}R(\boldsymbol{x}, \boldsymbol{x}')\mathcal{L}^T$ exists, σ and $\boldsymbol{\theta}$ be associated hyperparameters such that for all $\boldsymbol{x}, \boldsymbol{x}'$:

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}). \tag{33}$$

Under that formalism, the prediction of y requires a prior assessment of $\boldsymbol{\beta}$, σ and $\boldsymbol{\theta}$. One approach to determine $\boldsymbol{\psi} := (\boldsymbol{\beta}, \sigma, \boldsymbol{\theta})$ is to maximize the constrained log-likelihood function $L_{N,M}$ [3],

$$\boldsymbol{\psi}^{\mathrm{MLEc}} := \arg\max_{\boldsymbol{\psi}} L_{N,M}(\boldsymbol{\psi}), \tag{34}$$

where $L_{N,M}(\boldsymbol{\psi})$ corresponds to the evaluation in $y(\boldsymbol{X})$ of the log value of the conditional PDF of $Y(\boldsymbol{X})$ given $\boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z})$. Using the Bayes' theorem, this function can be written under the form:

$$\begin{aligned} L_{N,M}(\boldsymbol{\psi}) = L_N(\boldsymbol{\psi}) &- \log(\mathbb{P}(\boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z})) \\ &+ \log(\mathbb{P}(\boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}))), \end{aligned} \tag{35}$$

where $L_N(\boldsymbol{\psi})$ is the unconstrained log-likelihood. Thus, in addition to the estimation of $L_N(\boldsymbol{\psi})$, the computation of $L_{N,M}(\boldsymbol{\psi})$ for each value of $\boldsymbol{\psi}$ requires to calculate the difference between the prior and posterior, that is integrating the conditioning $Y(\boldsymbol{X}) = y(\boldsymbol{X})$, probabilities of respecting the constraints. However, as it will be shown in the application section, the gain brought by this hyperparameter identification procedure, both in terms of constraints respect and prediction capacity, is often low compared to its numerical cost. This explains that many papers (see for instance [1, 23]) propose to focus on the unconstrained log-likelihood only. Let $\boldsymbol{\psi}^{\mathrm{MLE}}$ be this value of $\boldsymbol{\psi}$ that maximises L_N. In this case, the respect of the constraints relies only on the choice of the M virtual observations.
Nevertheless, as we will also see in the application part, a very large number of virtual observation points may be necessary to compensate for correlation lengths chosen too small. And conversely, if the correlation lengths are too large, the probability of respecting the

constraints can be extremely low, which is likely to make the draw of the realizations of $\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z})$ very unstable numerically. To better integrate the constraints in the identification of $\boldsymbol{\psi}$, another approach inspired by the work achieved in [15] is now proposed. It consists in rewriting the identification problem of $\boldsymbol{\psi}$ in the form of a maximization problem under constraint:

$$\begin{aligned} &\max_{\boldsymbol{\psi}} L_N(\boldsymbol{\psi}) \\ &\quad \text{s.t. } \mathbb{P}(\boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X})) \geq 1-\gamma, \end{aligned} \tag{36}$$

with $0 \leq \gamma \leq 1$ a chosen tolerance. This problem can then be approached by:

$$\begin{aligned} \max_{\boldsymbol{\psi}} L_N(\boldsymbol{\psi}) \;\; &\text{s.t. for all } 1 \leq m \leq M : \\ &\mu_m^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})} + q^* \sqrt{(\boldsymbol{C}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})})_{mm}} \leq (\boldsymbol{u}_{\boldsymbol{\alpha}}(\boldsymbol{Z}))_m \\ &\mu_m^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})} - q^* \sqrt{(\boldsymbol{C}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})})_{mm}} \geq (\boldsymbol{\ell}_{\boldsymbol{\alpha}}(\boldsymbol{Z}))_m \end{aligned} \tag{37}$$

with q^* a chosen constant, and $\boldsymbol{\mu}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})}$ and $\boldsymbol{C}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})}$ the mean vector and covariance matrix of Gaussian random vector $\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})|Y(\boldsymbol{X}) = y(\boldsymbol{X})$. Contrary to the problem given by Eq.(36), the problem introduced in Eq.(37) no longer requires the costly computation of a probability associated with a truncated Gaussian distribution. Indeed, for a given value of $\boldsymbol{\psi}$, the expressions of $L_N(\boldsymbol{\psi})$, $\boldsymbol{\mu}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})}$ and $\boldsymbol{C}^{\mathcal{L}Y_{\boldsymbol{\alpha}}(\boldsymbol{Z})}$ can be explicitly derived, for a total calculation cost close to the evaluation of $L_N(\boldsymbol{\psi})$ only.
In the following, the approximated solution of problem (37) using an Augmented Lagrangian method is denoted by $\boldsymbol{\psi}^{\mathrm{AL}}$ (see [9] for more details about the interest of this method for solving constrained optimization problems).
In the problem (37), a particular attention has to be paid to the role of q^*. The greater q^* is, the greater the weight of the constraints will be in comparison to the likelihood. Choosing a large value for q^* (for instance, choosing $q^* = 2$), and thus forcing a strict respect of the constraints can indeed degrade the predictive capabilities of the model, by strongly underestimating or overestimating the prediction uncertainties. In our opinion, it is necessary to keep in mind that the respect of the constraints relies on two points: the values of the hyperparameters, but also the virtual observations. Thus, it seems to us more judicious to take a value of q^* close to 0. This ensures a reasonable probability of respect of the constraints before applying the constraints in the virtual points, and is likely to lead to a high probability of respect of the constraints once the virtual observation points will be added. In the following, q^* will be chosen equal to 0.1.

5. Joint identification of the hyperparameters and of the virtual observations

Until now, the choice of the M virtual observations gathered in $\boldsymbol{Z}$ and the estimation of the hyperparameters $\boldsymbol{\psi}$ characterizing the mean and covariance function of Y are carried out

independently. In Section 3, we explained how we can choose $\boldsymbol{Z}$ for a fixed value of $\boldsymbol{\psi}$ to impose constraints on Y. And in Section 4, we proposed several methods to choose the value of $\boldsymbol{\psi}$ for a fixed set of virtual observations gathered in $\boldsymbol{Z}$.
In order to maximize the predictive capabilities of the final predictor, and to maximize the probability of respecting the constraints, we now propose to adopt a mixed approach based on an alternation between enrichment of $\boldsymbol{Z}$ and hyperparameter re-estimation. This approach is summarized in Algorithm 1.
Several advantages can be listed for such an approach. First, by progressively increasing the number of constraints, the numerical cost of estimating the hyperparameters under constraints is limited. Secondly, by continuously adapting the hyperparameters to the constraints at the virtual observations, we limit the risk of being confronted with a too low probability of verifying the constraints at these points, which allows an accelerated execution of the generation procedures presented in [4].

Choose thresholds $q^* \in \mathbb{R}$ and $p_c^\star \in]0,1[$, numbers n_p, $n_{\mathcal{L}Y}$ and $M^{\max}$, and parametric representations for the mean function μ and covariance function C ;
Gather N evaluations of y in $y(\boldsymbol{X})$;
Compute the maximum likelihood estimate of the hyperparameters $\boldsymbol{\psi}$;
Let $Y \sim \mathrm{GP}(\mu(\boldsymbol{\psi}), C(\boldsymbol{\psi}))$ be the GPR-based surrogate model associated with y based on $\boldsymbol{\psi}$ and the N evaluations of y ;
Initialize $\boldsymbol{Z} = [\,]$, $\boldsymbol{\alpha}$, $M = 0$;
Compute $\widehat{p}_c$ as the average over n_p points of $\mathbb{X}$ of the empirical estimate of the constraints' probability function p_c based on $n_{\mathcal{L}Y}$ iid realizations of $\mathcal{L}Y^c_{N,M}$;
while $\widehat{p}_c < p_c^\star$ ***and*** $M < M^{max}$ **do**
    Minimize criterion c_k^{int} (for $k = 1$ or 2) ;
    Set $M = M + 1$, add the optimal point to $\boldsymbol{Z}$ and save the associated constraint in $\boldsymbol{\alpha}$;
    Update the value of $\boldsymbol{\psi}$ solving problem (37) ;
    Update the value of $\widehat{p}_c$;
end
Return $Y^c_{N,M}$.

Algorithm 1: Construction of the predictor $Y^c_{N,M}$.

Remark. In Algorithm 1, the stopping criterion is a threshold on the average probability of respecting all the constraints on the input domain. Nevertheless, for high dimensional applications with several constraints, the number of constraint points required can be very large, and the associated generation of realizations of truncated Gaussian vectors can be numerically very or even too difficult, which explains the addition of the second stopping criterion in number of maximum constraint points $M^{\max}$. Although it is reasonable to expect that the number of constraint points needed to ensure $\widehat{p}_c < p_c^\star$ grows with d_x and d_c, the choice of $M^{\max}$ is rather constrained by our ability to correctly estimate the probability of verifying the constraints at the constraint points. Following the recommendations provided in [4] on the stability of its algorithm, $M^{\max}$ can thus be chosen a priori equal to 200,

whatever the value of d_x and d_c.

6. Application

We list at least three objectives for the application section. First, we would like to show the crucial role of hyperparameters on taking into account inequality constraints. It is thus essential to try to integrate inequality constraints as early as the identification phase of the hyperparameters.
In a second step, it seems interesting to insist on the fact that a better consideration of the constraints does not necessarily result in a better prediction, on average, of the function of interest. This is particularly true in areas where the function of interest approaches the thresholds of the constraints. Nevertheless, considering an integrated adaptive approach, where virtual observations and hyperparameters are chosen at the same time and in a sequential way, allows a very good compromise between predictive capabilities and respect of constraints with high probability.
In this paper, we focus on cases where N is small compared to d_x, as these are often configurations where the consideration of potential constraints is particularly sought after to compensate for this small data presence. These are also the configurations where the choice of hyperparameters is the most important.

6.1. Presentation of the test cases

The interest of algorithm 1 for the construction of a Gaussian process predictor under inequality constraints is illustrated on five test cases, whose characteristics are listed in Table 1. None of the introduced examples will actually be costly to evaluate to make possible the performance analysis of the proposed algorithms.
On purpose, one or more inequality constraints can be associated to each function, which are also listed in Table 1. To get sound comparisons between the different ways of integrating constraints, the results presented in the next sections are averaged over 10 repetitions of the whole procedures.
Moreover, for each studied function, a simple linear trend and a tensorized stationary Matern-5/2 kernel are chosen:

$$\mu(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{d_x} \beta_i x_i, \tag{38}$$

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \prod_{i=1}^{d_x} (1 + \sqrt{5}\Delta x_i + \frac{5}{3}\Delta x_i^2) \exp\left(-\sqrt{5}\Delta x_i\right), \ \Delta x_i := \frac{|x_i - x_i'|}{\theta_i}. \tag{39}$$

As a consequence, the vector of hyperparameters $\boldsymbol{\psi} = (\boldsymbol{\beta}, \sigma, \boldsymbol{\theta})$ is constituted of $2(d_x + 1)$ constants to be identified, and the vector $\boldsymbol{\theta}$ gathers the correlation lengths.

Example	Name	d_x	N	constraint operator $\mathcal{L}y$	d_c
1	y_1^{1D}	1	4	dy/dx	1
2	y_2^{1D}	1	10	dy/dx	1
3	y_3^{1D}	1	7	$(y, dy/dx)$	2
4	y^{3D}	3	12	$(y, \partial y/\partial x_1, \partial^2 y/\partial x_2^2, \partial^2 y/\partial x_3^2)$	4
5	y^{5D}	5	35	$(\partial^2 y/\partial x_1^2, \partial^2 y/\partial x_2^2, \partial^2 y/\partial x_3^2, \partial y/\partial x_4, \partial y/\partial x_5)$	5

Table 1: Characteristics of the five analyzed numerical functions (see AppendixC for the expressions of the functions).

6.2. Analysis of the results

The comparison results for the one-dimensional functions are summarized in Figures 1, 2 and 3. Four configurations are compared in these three sets of figure. No constraint is taken into account in the figures labelled (a), but constraints are imposed in M points in the other figures. However, whereas the positions of these constraints are a priori chosen for the figures labelled (b) and (c), the positions of the constraints in the figures labelled (d) are automatically selected using Algorithm 1. Then, the MLE of $\boldsymbol{\psi}$ (i.e. without constraint) is considered in the figures labelled (a) and (b), whereas in the figures labelled (c) and (d), $\boldsymbol{\psi}$ corresponds to the approximated solution using an Augmented Lagragian method of problem (37) associated with the M former points of constraint.

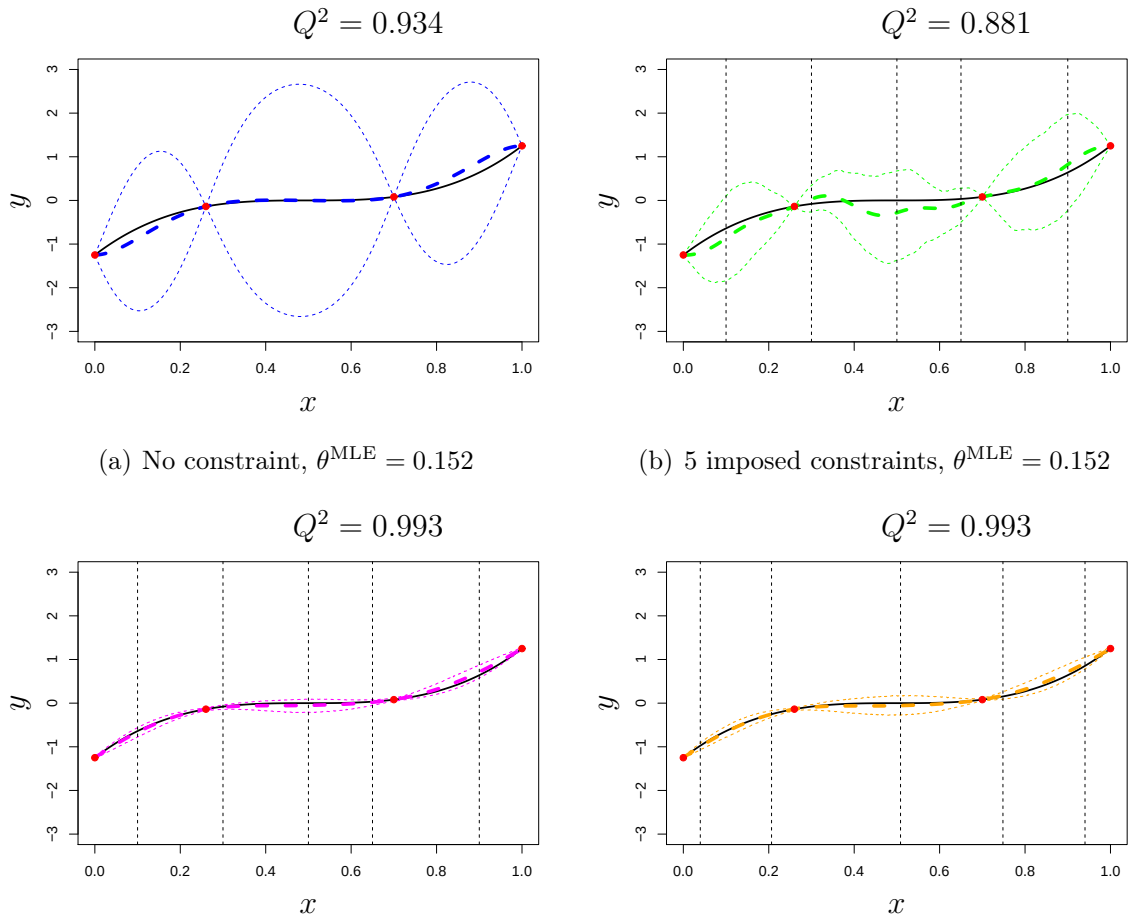
The objective of Figures 1 and 2 is to highlight two typical pathologies that can appear when constraints are not integrated in the hyperparameter selection process. Focusing on Figure 1, we observe that not integrating the constraints can lead to a strong underestimation of the correlation length. This has two direct consequences: a strong overestimation of the confidence intervals, and an over-sensitivity of the prediction mean to the addition of the constraint points. In this example, we also notice that adding the five monotony constraints creates artificial oscillations for the prediction mean around $x = 0.5$, which results in a reduction of the Q^2 value, which is the classical metric of learning performance on test data calculated as one minus the predictive residual error sum of squares (PRESS) divided by the total sum of squares (TSS). As expected, a much longer correlation length is obtained when integrating the constraints, which results in an increased Q^2 value, but also a strong reduction of the confidence intervals around the true function to be predicted. For this example, we also see that the positions of the constraints found by Algorithm 1 allow an interesting compromise between prediction capacity and respect of monotony.

If we now focus on Figure 2, we can see that integrating information on the sign of the derivative can avoid considering too high correlation lengths this time. Imposing the sign of the derivative at three a priori pathological points does not greatly improve the model if we keep $\theta^{\mathrm{MLE}} = 0.183$. We can even say that it degrades it, since it reduces the size of the confidence intervals when they already did not contain the true function. On the contrary, by integrating this information on the first derivative in the choice of the correlation length, we obtain a much more reasonable model, associated with a much lower correlation length. For this model, we retrieve the capacity of Algorithm 1 to correctly position the constraint

points in potentially pathological areas.
The third one-dimensional example deals with bounds and monotony constraints. This example, inspired by the functions studied in [1], serves to illustrate the great interest that there can be in taking into account several kinds of inequality constraints, when they exist, in the construction of Gaussian process predictors. Looking at Figure 3, we note once again that playing on both the location of the constraint points and the estimation of the hyperaparameters allows to reduce the confidence intervals around the true function, and thus to strongly improve the predictive character of the predictor. To go further, Table 2 quantifies the potential gains associated with the different approaches proposed. Seven cases are compared, which are associated with different values of the hyperparameters ($\theta = \theta^{\mathrm{MLE}}$ or $\theta = \theta^{\mathrm{AL}}$) and different selection criteria for the constraint points (c_1, c_2 or c_1^{int}). These sequential selection criteria are also compared to a Space Filling (SF) choice of the constraint points, i.e. the constraint points are distributed as uniformly as possible in the input area (see [8, 5, 17] for further details about the construction of such space filling designs).
The main information to remember from this table is that for one-dimensional applications ($d_x = 1$), for which it is possible to position the constraint points relatively densely in the input space, all the selection criteria have more or less similar performance, both in terms of Q^2 and the probability of respecting the constraints p_c. Nevertheless, it should be noted that for lower values of M, a better overall probability of respecting the constraints is achieved for the approach associated with Algorithm 1, the results of which being placed in the last column on the right.
Differences however appear when the dimension of the entry space increases, which can be seen on Tables 3 and 4, respectively associated with $d_x = 3$ and $d_x = 5$. First, these tables allow us to underline the importance of the choice of the hyperparameters for the good respect of the constraints. For example, for the 3D example, choosing $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{AL}}$ with only $M = 40$ points uniformly chosen in $\mathbb{X}$ (fourth configuration, first line), leads to a p_c value greater than if we choose $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{MLE}}$ with $M = 120$ constraint points (second configuration, third line). The same observations can be made on the 5D example, where taking $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{AL}}$ with $M = 35$ leads to better results than taking $\boldsymbol{\theta} = \boldsymbol{\theta}^{\mathrm{MLE}}$ with $M = 175$. We then notice that for identical hyperparameters, an adaptive selection of constraint points systematically leads to a better respect of the constraints, whatever the chosen criterion. Finally, for these examples, criterion c_2 seems a little more interesting than criterion c_1, but we especially notice that again, the association c_1^{int} plus sequential estimation of $\boldsymbol{\theta}$, which is described in Algorithm 1, allows the fastest convergence of p_c to 1.

7. Conclusions

This paper focuses on the consideration of linear constraints in the Gaussian process regression (GPR) formalism. In particular when few observation points are available, taking into account a priori knowledge about the model in the form of linear constraints on the output of the code can indeed strongly reduce the prediction uncertainties, while improving its explainability.



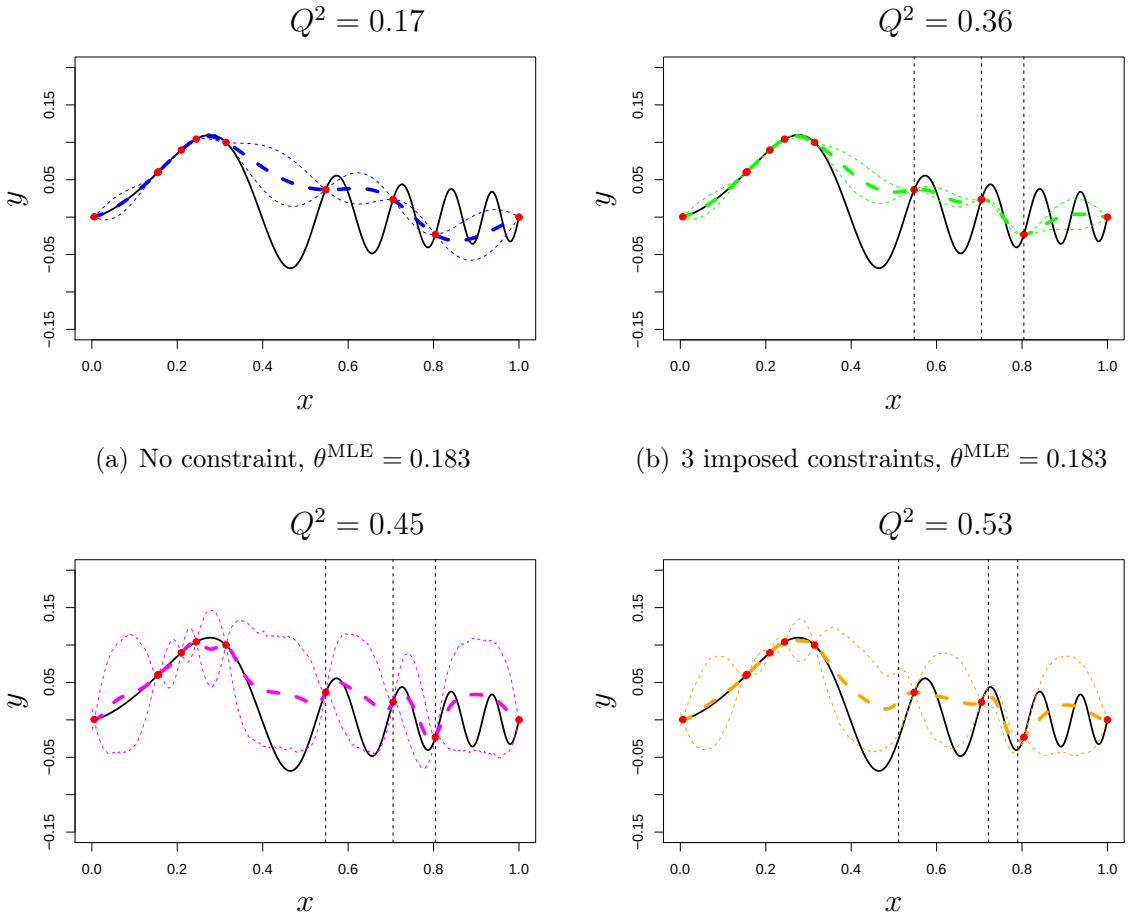
(a) No constraint, $\theta^{\mathrm{MLE}} = 0.152$

(b) 5 imposed constraints, $\theta^{\mathrm{MLE}} = 0.152$

(c) 5 imposed constraints, $\theta^{\mathrm{AL}} = 1.98$

(d) 5 constraints sequentially chosen, $\theta^{\mathrm{AL}} = 1.99$

Figure 1: Impact of taking constraints into account on the performance of the GPR surrogate model in predicting y_1^{1D} function values. The black continuous lines correspond to the true value of y_1^{1D}, the red dots are the observations points for the construction of the GPR, the thick dashed lines are the GPR mean predictions, the thin dashed lines correspond to 95% prediction intervals provided by the GPR, and the vertical dotted lines indicate the positions of the virtual observations where monotonicity constraints are imposed.



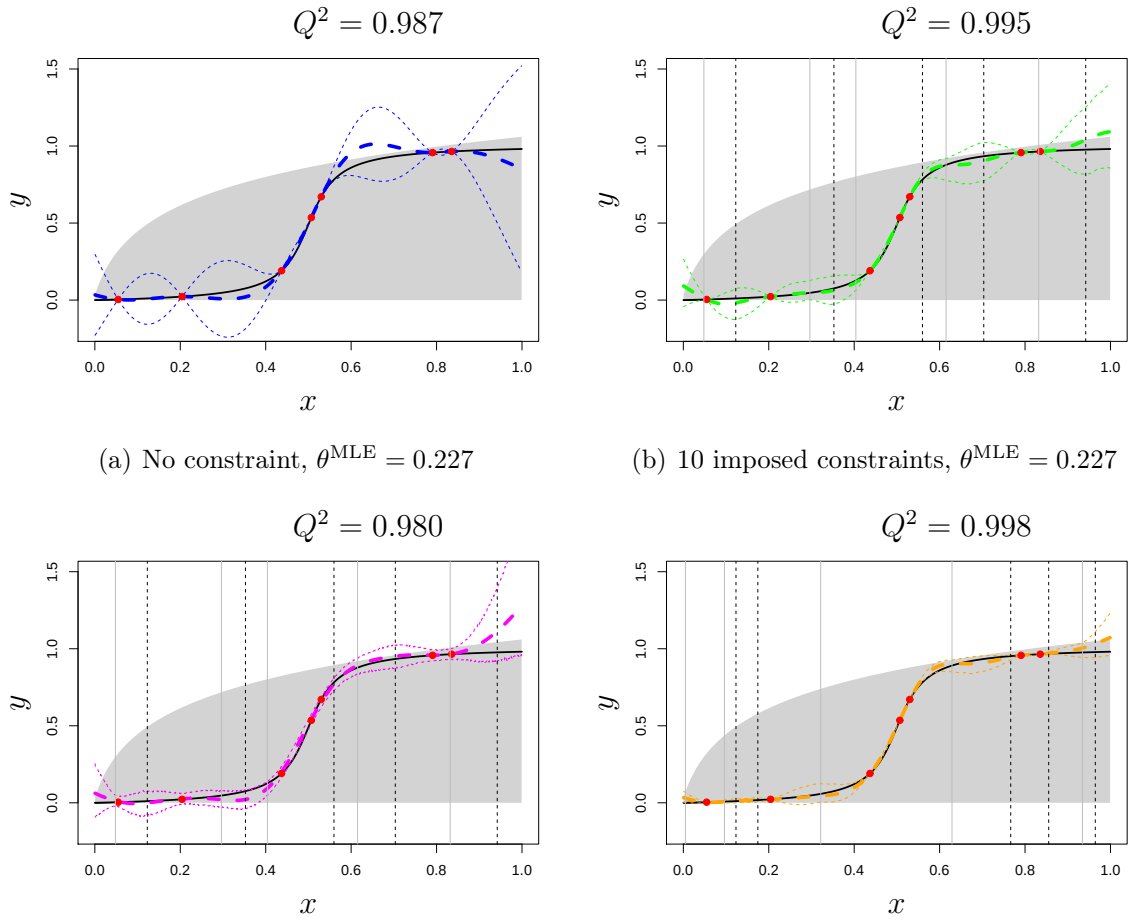
(a) No constraint, $\theta^{\mathrm{MLE}} = 0.183$

(b) 3 imposed constraints, $\theta^{\mathrm{MLE}} = 0.183$

(c) 3 imposed constraints, $\theta^{\mathrm{AL}} = 0.035$

(d) 3 constraints sequentially chosen, $\theta^{\mathrm{AL}} = 0.054$

Figure 2: Impact of taking constraints into account on the performance of the GPR surrogate model in predicting y_2^{1D} function values. The black continuous lines correspond to the true value of y_2^{1D}, the red dots are the observations points for the construction of the GPR, the thick dashed lines are the GPR mean predictions, the thin dashed lines correspond to 95% prediction intervals provided by the GPR, and the vertical dotted lines indicate the positions of the virtual observations where monotonicity constraints are imposed.



(a) No constraint, $\theta^{\mathrm{MLE}} = 0.227$

(b) 10 imposed constraints, $\theta^{\mathrm{MLE}} = 0.227$

(c) 10 imposed constraints, $\theta^{\mathrm{AL}} = 1.34$

(d) 10 constraints sequentially chosen, $\theta^{\mathrm{AL}} = 0.380$

Figure 3: Impact of taking constraints into account on the performance of the GPR surrogate model in predicting y_3^{1D} function values. The black continuous lines correspond to the true value of y_3^{1D}, the red dots are the observations points for the construction of the GPR, the thick dashed lines are the GPR mean predictions and the thin dashed lines correspond to 95% prediction intervals provided by the GPR. The vertical grey solid lines and vertical black dotted lines are the positions of the virtual observations where boundedness and monotonicity constraints are imposed respectively, while the grey areas characterize the admissible areas for the output values.

M	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
10	0.609	0.666	0.666	0.683	0.883	0.881	0.868	**0.903**
20	0.609	0.747	0.742	0.745	0.956	0.947	0.942	**0.960**
30	0.609	0.799	0.795	0.803	0.973	0.976	0.970	**0.977**
40	0.609	0.820	0.817	0.835	0.982	**0.985**	0.982	**0.985**
50	0.609	0.926	0.934	0.945	0.989	**0.990**	0.988	**0.990**
Q^2	0.971	**0.982**	0.981	**0.982**	**0.982**	**0.982**	**0.982**	**0.982**

Table 2: For different selection strategies and for the test function $y_3^{(1D)}$, this table represents the values of the Q^2 criterion, and the evolution with respect to M of the average value over $\mathbb{X}$ of the probability of verifying the constraints, p_c. (1): no constraints applied, $\theta = \theta^{\text{MLE}}$. (2): Space Filling (SF) design, $\theta = \theta^{\text{MLE}}$. (3): SF design, $\theta = \theta^{\text{MLEc}}$. (4): SF design, $\theta = \theta^{\text{AL}}$. (5): sequential design (Seq.D) with c_1, $\theta = \theta^{\text{MLE}}$. (6): Seq.D with c_2, $\theta = \theta^{\text{MLE}}$. (7): Seq.D with c_1^{int}, $\theta = \theta^{\text{MLE}}$. (8): Seq.D with c_1^{int}, sequential estimation of θ with the Augmented Lagrangian-based approach.

M	(1)	(2)	(3)	(4)	(5)	(6)	(7)
40	0.389	0.395	0.463	0.650	0.700	0.747	**0.959**
80	0.389	0.413	0.487	0.665	0.790	0.827	**0.976**
120	0.389	0.574	0.671	0.787	0.842	0.869	**0.986**
160	0.389	0.760	0.846	0.931	0.874	0.896	**0.992**
Q^2	0.930	0.947	0.959	**0.975**	0.947	0.943	0.968

Table 3: For different selection strategies and for the test function $y^{(3D)}$, this table represents the values of the Q^2 criterion, and the evolution with respect to M of the average value over $\mathbb{X}$ of the probability of verifying the constraints, p_c. (1): no constraints applied, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (2): Space Filling (SF) design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (3): SF design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLEc}}$. (4): SF design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{AL}}$. (5): sequential design (Seq.D) with c_1, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (6): Seq.D with c_2, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (7): Seq.D with c_1^{int}, sequential estimation of $\boldsymbol{\theta}$ with the Augmented Lagrangian-based approach.

M	(1)	(2)	(3)	(4)	(5)	(6)	(7)
35	0.190	0.448	0.520	0.727	0.505	0.570	**0.820**
70	0.190	0.583	0.670	0.820	0.561	0.649	**0.887**
105	0.190	0.581	0.670	0.822	0.586	0.700	**0.920**
140	0.190	0.603	0.695	0.860	0.620	0.734	**0.945**
175	0.190	0.603	0.697	0.860	0.636	0.752	**0.957**
Q^2	0.961	0.961	0.974	**0.990**	0.969	0.962	0.987

Table 4: For different selection strategies and for the test function $y^{(5D)}$, this table represents the values of the Q^2 criterion, and the evolution with respect to M of the average value over $\mathbb{X}$ of the probability of verifying the constraints, p_c. (1): no constraints applied, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (2): Space Filling (SF) design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (3): SF design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLEc}}$. (4): SF design, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{AL}}$. (5): sequential design (Seq.D) with c_1, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (6): Seq.D with c_2, $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{MLE}}$. (7): Seq.D with c_1^{int}, sequential estimation of $\boldsymbol{\theta}$ with the Augmented Lagrangian-based approach.

However, for reasons of numerical stability and computational cost, it is often too difficult to impose these constraints at any point of the domain, especially when we are interested in the prediction of functions depending on several parameters.
In order to guarantee, in a reasonable computation time, the respect of these constraints with the highest possible probability, this work has thus proposed two adaptations of previous works: the first one concerning the selection of the reduced number of entry points at which the constraints are imposed, the second one concerning the optimization of the statistical moments of the Gaussian process on which the GPR model is based.
From a theoretical point of view, the proposed method can be applied to the prediction of functions in any dimensions, incorporating any number of linear constraints. But from a practical point of view, since the consideration of constraints is based on local additions of constraint points, it is certain that the larger the dimension of the inputs will be, the more constraint points will have to be added, and the more difficult it will be to obtain a high probability of respecting the constraints. And to deal with higher dimensional problems ($d_x > 10$ for example), other directions will probably have to be explored, which could be the subject of future work.

AppendixA. Proof of Proposition 1

By definition, for all $\boldsymbol{x} \in \mathbb{X}$, we have:

$$\begin{pmatrix} Y(\boldsymbol{x}) \\ \mathcal{L}Y(\boldsymbol{x}) \\ Y(\boldsymbol{X}) \\ \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu(\boldsymbol{x}) \\ \mathcal{L}\mu(\boldsymbol{x}) \\ \mu(\boldsymbol{X}) \\ \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z}) \end{pmatrix}, \begin{bmatrix} C(\boldsymbol{x},\boldsymbol{x}) & C(\boldsymbol{x},\boldsymbol{x})\mathcal{L}^T & C(\boldsymbol{x},\boldsymbol{X}) & C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}C(\boldsymbol{x},\boldsymbol{x}) & \mathcal{L}C(\boldsymbol{x},\boldsymbol{x})\mathcal{L}^T & \mathcal{L}C(\boldsymbol{x},\boldsymbol{X}) & \mathcal{L}C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ C(\boldsymbol{X},\boldsymbol{x}) & C(\boldsymbol{X},\boldsymbol{x})\mathcal{L}^T & C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) & C(\boldsymbol{X},\boldsymbol{x})\mathcal{L}^T & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix} \right). \tag{A.1}$$

For each $\boldsymbol{z} \in \mathbb{R}^M$, if we focus on vectors $(Y(\boldsymbol{x}), Y(\boldsymbol{X}), \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}))$ and $(\mathcal{L}Y(\boldsymbol{x}), Y(\boldsymbol{X}), \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}))$, whose statistical properties can be deduced from Eq. (A.1) by removing the second and the first row respectively, we notice by Gaussian conditioning that $\widetilde{Y}(\boldsymbol{z}) := Y(\boldsymbol{x}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}), \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) = \boldsymbol{z}$ and $\mathcal{L}\widetilde{Y}(\boldsymbol{z}) := \mathcal{L}Y(\boldsymbol{x}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}), \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) = \boldsymbol{z}$ are still Gaussian, and we have:

$$\begin{aligned} \mathbb{E}\left[\widetilde{Y}(\boldsymbol{z})\right] &= \mu(\boldsymbol{x}) + [C(\boldsymbol{x},\boldsymbol{X})\ C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T] \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} y(\boldsymbol{X}) - \mu(\boldsymbol{X}) \\ \boldsymbol{z} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z}) \end{pmatrix} \\ &= \mu(\boldsymbol{x}) + \boldsymbol{a}_1^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{a}_2^T(\boldsymbol{x})(\boldsymbol{z} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})), \end{aligned} \tag{A.2}$$

$$\begin{aligned} \mathrm{Cov}\left(\widetilde{Y}(\boldsymbol{z})\right) &= C(\boldsymbol{x},\boldsymbol{x}) - [C(\boldsymbol{x},\boldsymbol{X})\ C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T] \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} C(\boldsymbol{X},\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) \end{pmatrix} \\ &= C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}), \end{aligned} \tag{A.3}$$

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}\widetilde{Y}(\boldsymbol{z})\right] &= \mathcal{L}\mu(\boldsymbol{x}) + [\mathcal{L}C(\boldsymbol{x},\boldsymbol{X})\ \mathcal{L}C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T] \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} y(\boldsymbol{X}) - \mu(\boldsymbol{X}) \\ \boldsymbol{z} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z}) \end{pmatrix} \\ &= \mathcal{L}\mu(\boldsymbol{x}) + \boldsymbol{A}_3^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{A}_4^T(\boldsymbol{x})(\boldsymbol{z} - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})),\end{aligned} \tag{A.4}$$

$$\begin{aligned}\mathrm{Cov}\left(\mathcal{L}\widetilde{Y}(\boldsymbol{z})\right) &= \mathcal{L}C(\boldsymbol{x},\boldsymbol{x})\mathcal{L}^T - [\mathcal{L}C(\boldsymbol{x},\boldsymbol{X})\ \mathcal{L}C(\boldsymbol{x},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T] \begin{bmatrix} C(\boldsymbol{X},\boldsymbol{X}) & C(\boldsymbol{X},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{X}) & \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{Z})\mathcal{L}_{\boldsymbol{\alpha}}^T \end{bmatrix}^{-1} \begin{pmatrix} C(\boldsymbol{X},\boldsymbol{x})\mathcal{L}^T \\ \mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x})\mathcal{L}^T \end{pmatrix} \\ &= \mathcal{L}C(\boldsymbol{x},\boldsymbol{x})\mathcal{L}^T - \boldsymbol{A}_3^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x})\mathcal{L}^T - \boldsymbol{A}_4^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x})\mathcal{L}^T.\end{aligned} \tag{A.5}$$

We deduce the following :

$$\begin{aligned}&\mathbb{E}\left[Y_{N,M}^c(\boldsymbol{x})\right] = \mathbb{E}\left[Y(\boldsymbol{x}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}),\ \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z})\right] \\ &\quad = \mathbb{E}\left[\mathbb{E}\left[Y(\boldsymbol{x}) \mid \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \\ &\quad = \mu(\boldsymbol{x}) + \boldsymbol{a}_1^T(\boldsymbol{x})(y(\boldsymbol{X}) - \mu(\boldsymbol{X})) + \boldsymbol{a}_2^T(\boldsymbol{x})(\mu_{\boldsymbol{\alpha}}(\boldsymbol{Z}) - \mathcal{L}_{\boldsymbol{\alpha}}\mu(\boldsymbol{Z})),\end{aligned} \tag{A.6}$$

$$\begin{aligned}&\mathbb{E}\left[Y_{N,M}^c(\boldsymbol{x})^2\right] = \mathbb{E}\left[Y(\boldsymbol{x})^2 \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}),\ \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z})\right] \\ &\quad = \mathbb{E}\left[\mathbb{E}\left[Y(\boldsymbol{x})^2 \mid \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \\ &\quad = \mathbb{E}\left[\mathrm{Cov}\left(\widetilde{Y}(\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}))|\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z})\right) + \mathbb{E}\left[\widetilde{Y}(\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}))|\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z})\right]^2 \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \\ &\quad = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) \\ &\qquad + \mathbb{E}\left[\mathbb{E}\left[\widetilde{Y}(\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}))|\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z})\right]^2 \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \\ &\quad = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) \\ &\qquad + \mathbb{E}\left[(\mathbb{E}\left[Y_{N,M}^c(\boldsymbol{x})\right] + \boldsymbol{a}_2^T(\boldsymbol{x})(\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) - \mu_{\alpha}(\boldsymbol{Z})))^2 \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right] \\ &\quad = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) \\ &\qquad + \mathbb{E}\left[Y_{N,M}^c(\boldsymbol{x})^2\right] + \boldsymbol{a}_2^T(\boldsymbol{x})\mathbb{E}\left[(\mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) - \mu_{\alpha}(\boldsymbol{Z}))^2 \mid \boldsymbol{\ell}(\boldsymbol{Z}) \leq_c \mathcal{L}_{\boldsymbol{\alpha}}Y(\boldsymbol{Z}) \leq_c \boldsymbol{u}(\boldsymbol{Z}), Y(\boldsymbol{X}) = y(\boldsymbol{X})\right]\boldsymbol{a}_2(\boldsymbol{x}) \\ &\quad = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) + \boldsymbol{a}_2^T(\boldsymbol{x})C_{\boldsymbol{\alpha}}(\boldsymbol{Z})\boldsymbol{a}_2(\boldsymbol{x}) + \mathbb{E}\left[Y_{N,M}^c(\boldsymbol{x})^2\right],\end{aligned} \tag{A.7}$$

so that:

$$\mathrm{Cov}(Y_{N,M}^c(\boldsymbol{x})) = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{a}_1^T(\boldsymbol{x})C(\boldsymbol{X},\boldsymbol{x}) - \boldsymbol{a}_2^T(\boldsymbol{x})\mathcal{L}_{\boldsymbol{\alpha}}C(\boldsymbol{Z},\boldsymbol{x}) + \boldsymbol{a}_2^T(\boldsymbol{x})C_{\boldsymbol{\alpha}}(\boldsymbol{Z})\boldsymbol{a}_2(\boldsymbol{x}). \tag{A.8}$$

The two other expressions are obtained using the same decompositions of the expectation function.

AppendixB. Proof of Proposition 2

The result of Proposition 2 is a direct consequence of Eqs. (A.2) and (A.3), where we have used the fact that in the normal distribution of $Y(\boldsymbol{x}) \mid Y(\boldsymbol{X}) = y(\boldsymbol{X}), \mathcal{L}_{\boldsymbol{\alpha}} Y(\boldsymbol{Z})$, only the mean depends on $\boldsymbol{L}(\boldsymbol{Z})$.

AppendixC. Expression of the analyzed numerical functions

The three 1D functions $y_1^{1\mathrm{D}}, y_2^{1\mathrm{D}}, y_3^{1\mathrm{D}}$ correspond to :

$$y_1^{1\mathrm{D}} : \left\{ \begin{array}{ccc} [0,1] & \to & \mathbb{R} \\ x & \mapsto & 10(x-0.5)^3, \end{array} \right. \tag{C.1}$$

$$y_2^{1\mathrm{D}} : \left\{ \begin{array}{ccc} [0,1] & \to & \mathbb{R} \\ x & \mapsto & \frac{\sin(10\pi x^{5/2})}{10\pi x}, \end{array} \right. \tag{C.2}$$

$$y_3^{1\mathrm{D}} : \left\{ \begin{array}{ccc} [0,1] & \to & \mathbb{R} \\ x & \mapsto & 1/3(\mathrm{atan}(20x-10) - \mathrm{atan}(-10)), \end{array} \right. \tag{C.3}$$

The 3D function $y^{3\mathrm{D}}$ corresponds to the function:

$$y^{3\mathrm{D}} : \left\{ \begin{array}{ccc} [0,1]^3 & \to & \mathbb{R} \\ \boldsymbol{x} & \mapsto & (x_3 - x_2^2)^2 + (x_2 - x_1^2)^2 + (1-x_2)^2 + (1-x_1)^2 + 3x_1. \end{array} \right. \tag{C.4}$$

The 5D function $y^{5\mathrm{D}}$ corresponds to the function:

$$y^{5\mathrm{D}} : \left\{ \begin{array}{ccc} [0,1]^5 & \to & \mathbb{R} \\ \boldsymbol{x} & \mapsto & 10\sin(\pi x_1 x_2) + 40(x_3 - 0.5)^2(x_4 + 0.25) + 5x_5. \end{array} \right. \tag{C.5}$$

[1] Christian Agrell. Gaussian processes with linear operator inequality constraints. *Journal of Machine Learning Research*, 20:1–36, 2019.

[2] Y. Auffray, P. Barbillon, and J.M. Marin. Maximin design on non hypercube domains and kernel interpolation. *Statistics and Computing*, 22(3):703–712, 2012.

[3] François Bachoc, Agnès Lagnoux, and Andres Lopez Lopera. Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics*, 13:2921–2969, 2019.

[4] Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, 2017.

[5] G. Damblin, M. Couplet, and B. Iooss. Numerical studies of space filling designs: optimization of latin hypercube samples and subprojection properties. *Journal of Simulation*, 7:276–289, 2013.

[6] K.T. Fang. Wrap-aroung l_2-discrepancy of random sampling, latin hypercube and uniform designs. *Journal of Complexity*, 17:608–624, 2001.

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

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

[9] E.G. Golshtein. An iterative linear programming algorithm based on an augmented lagrangian. In Olvi L. Mangasarian, Robert R. Meyer, and Stephen M. Robinson, editors, *Nonlinear Programming 4*, pages 131–146. Academic Press, 1981.

[10] V. R. Joseph, E. Gul, and S. Ba. Maximum projection designs for computer experiments. *Biometrika*, 102(2):371–380, 2015.

[11] M C Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal of the royal statistical society*, 63:425–464, 2001.

[12] J. H. Kotecha and P. M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference - Volume 03*, ICASSP '99, pages 1757–1760, USA, 1999. IEEE Computer Society.

[13] Andres Lopez-Lopera, Francois Bachoc, Nicolas Durrande, and Olivier Roustant. Finite dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

[14] Hassan Maatouk and Xavier Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, 2017.

[15] Andrew Pensoneault, Xiu Yang, and Xueyu Zhu. Nonnegativity-enforced gaussian process regression. *Theoretical and Applied Mechanics Letters*, 10(3):182–187, 2020.

[16] G. Perrin. Point process-based approaches for the reliability analysis of systems modeled by costly simulators. *Reliability Engineering & System Safety*, 214, 2021.

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

[18] Jaakko Riihimki and Aki Vehtari. Gaussian processes with monotonicity information. *Journal of Machine Learning Research - Proceedings Track*, 9:645–652, 2010.

[19] J. Sacks, W. Welch, T. Mitchell, and H. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.

[20] T. J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments*. Springer, New York, 2003.

[21] Laura Swiler, Mamikon Gulian, Ari Frankel, Cosmin Safta, and John Jakeman. A survey of constrained Gaussian process regression: approahces and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2):119–156, 2020.

[22] Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. *Annales de la Faculté des sciences de Toulouse : Mathématiques*, Ser. 6, 21(3):529–555, 2012.

[23] Sébastien Da Veiga and Amandine Marrel. Gaussian process regression with linear inequality constraints. *Reliability Engineering & System Safety*, 195:106732, 2020.

[24] Xiaojing Wang and James O. Berger. Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4:1–25, 2016.

[25] D. Zhan and H. Xing. Expected improvement for expensive optimization: a review. *J Glob Optim*, 78:507–544, 2020.