Gaussian process regression with linear inequality constraints

Sébastien Da Veiga, Amandine Marrel

Gaussian process regression with linear inequality constraints

Sébastien DA VEIGA[1] and Amandine MARREL[2]

April 27, 2017

[1] Safran
Magny-Les-Hameaux, France
sebastien.da-veiga@safran.fr

[2] CEA, DEN, DER, F-13108 Saint-Paul-lez-Durance, France
amandine.marrel@cea.fr

Abstract
The analysis of expensive numerical simulators usually requires metamodelling techniques, among which Gaussian process regression is one of the most popular approaches. Frequently, the code outputs correspond to physical quantities with a behavior which is known a priori: Chemical concentrations lie between 0 and 1, the output is increasing with respect to some parameter, etc. In this paper, we introduce a framework for incorporating any type of linear constraints in Gaussian process modeling, including common bound and monotonicity constraints. This new methodology mainly relies on conditional expectations of the truncated multinormal distribution and a discretization of the input space. When dealing with high-dimensional functions, the discretization suffers from the curse of dimensionality. We thus introduce a sequential sampling strategy where the input space is explored via a criterion which maximizes the probability of respecting the given constraints. To further reduce the computational burden, we also recommend a correlation-free approximation. The proposed approaches are evaluated and compared on several analytical functions with different instances of linear constraints.

Keywords: Computer experiment, Gaussian process, Constrained regression, Sequential sampling.

1 Introduction

In the computer experiments community, surrogate models are essential tools for the understanding of complex phenomena, since modern computer codes usually take several hours or days for a single run. In this setting, uncertainty quantification or sensitivity analysis cannot be performed directly, since they require several thousands of calls to the numerical simulator. The central idea behind metamodeling is to build a proxy (or *surrogate*) for the expensive code from a small number of simulations. Many methods dedicated to sensitivity analysis with proxy models have been proposed in the past few years Oakley and O'Hagan (2004), Marrel et al. (2008), Da Veiga et al. (2009). Standard surrogate models used for computer experiments include linear models, smoothing splines, polynomial chaos expansions, random forests or Gaussian process (GP) regression, among others.

Very often, if not always, such computer codes aim at simulating real physical phenomena through the resolution of equations (ordinary or partial differential equations, simplified analytical formula, ...). But

most importantly, these models usually encompass symmetries, constraints on the sign of output variables, monotonicity with respect to some input variables or other constraints which are due to the very nature of the physics under consideration. From a practical point of view, it would be highly desirable to build a surrogate model which respects the same constraints: not only the predictions would comply with the expected physical behavior, but also the practitioners who developed the numerical simulator would have more confidence in the proxy. On the "statistical" side, of course we expect that integrating the constraints in the surrogate would lead to better prediction accuracy and robustness. Despite the extensive use of meta-models and the potential gain, only a few works tackle constraint incorporation in general regression models. Monotonicity is easily reachable in linear models through non-negative least squares. For nonlinear regression, monotonicity constraints were investigated by Hall and Huang (2001), Ramsay and Silverman (2005) and Bigot and Gadat (2010) in a one-dimensional setting, and Dette and Scheder (2006) and Racine et al. (2009) in a general kernel regression framework with no dimensionality restriction. Focusing on Gaussian Process (GP) regression, bound constraints have been studied by Abrahamsen and Benth (2001), Yoo and Kyriakidis (2006) and Michalak (2008) while monotonicity was examined by Kleijnen and van Beers (2010), Riihimäki and Vehtari (2010) and Maatouk and Bay (2014). Most approaches incorporate these constraints with specific tools and do not generalize easily.

In a previous work, we proposed a framework for linear constraints and GP regression based on conditional expectations Da Veiga and Marrel (2012). By making use of moment approximations for the truncated multivariate normal distribution, we introduced a GP surrogate capable of accounting for bounds and monotonicity constraints. The results on low-dimensional test examples were highly promising, but a straightforward implementation on examples with a moderate number of regressors is problematic. Indeed, our surrogate approximates a conditional expectation by discretizing the constraint on a finite number of points, which implies that the curse of dimensionality applies to the estimation of the mean of a very high-dimensional truncated multivariate normal vector. Our objective here is twofold. First, we generalize our surrogate to other types of frequent linear constraints appearing in the study of vector fields. This includes for example divergence or curl constraints and generalizes the work of Scheuerer and Schlather (2012). Second, we develop a sequential strategy for an efficient span of the constraint space which makes it possible to tackle problems with up to 10 or 15 dimensions.

The structure of the paper is as follows. In Section 2, we first recap the standard GP formulation, the conditional expectation framework for building a constrained surrogate and the corresponding algorithm. It is also illustrated on several common physical constraints. In Section 3, we introduce different new sequential strategies for spanning the constraint space. Several numerical experiments are conducted to compare these strategies on analytical test functions.

2 Accounting for constraints in Gaussian process regression

In this section, we first briefly introduce the Gaussian process modeling framework. We then detail the theoretical setting for incorporating constraints. In what follows, we assume that the complex computer code is represented by a function $g : \mathbb{R}^D \to \mathbb{R}$ which is assumed to be continuous. For a given value of the vector of inputs $\mathbf{x} = \left(x^1, \ldots, x^D\right) \in \mathbb{R}^D$, a simulation run of the code yields a real value $y = g(\mathbf{x})$, which corresponds to the output of interest. In practice, one evaluation of the function g can take several hours or even days. As a result, we make use here of response surface methods. The idea is the following. For a given n-size sample of input vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we compute the corresponding outputs $y_1, \ldots, y_n$. The goal is to build an approximating model of g using the n-sample $(\mathbf{x}_i, y_i)_{i=1,\ldots,n}$. We let $X_s = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T\right]^T$ and $Y_s = [y_1, \ldots, y_n]^T$ denote the matrix of sample input values and the vector of responses, respectively.

2.1 Standard Gaussian process modeling

The GP modeling introduced by Sacks et al. (1989) considers the deterministic response $y = g(\mathbf{x})$ as a realization of a random field $Y(\mathbf{x})$ given by the following decomposition:

$$Y(\mathbf{x}) = g_0(\mathbf{x}) + U(\mathbf{x})$$

where $g_0(\mathbf{x})$ is the mean function (*e.g.* a polynomial) and $U(\mathbf{x})$ is a stationary centered Gaussian field with covariance function $C(\mathbf{x}, \mathbf{x}') = \sigma^2 R(\mathbf{x}, \mathbf{x}')$, σ^2 and R being the variance and correlation function respectively. Note that stationarity implies that its covariance function $C(\mathbf{x}, \mathbf{x}')$ can be written as $C(\boldsymbol{\tau}) = \sigma^2 R(\boldsymbol{\tau})$ with $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$. In this setting, the conditional distribution of the response at a new location $\mathbf{x}^*$ is a Gaussian distribution with moments given by

$$\mathbb{E}\left(Y(\mathbf{x}^*)|Y(X_s) = Y_s\right) = g_0(\mathbf{x}^*) + k(\mathbf{x}^*)^T \Sigma_S^{-1} \left(Y_s - G_{0,s}\right) \tag{1}$$

$$\mathrm{Var}\left(Y(\mathbf{x}^*)|Y(X_s) = Y_s\right) = \sigma^2 - k(\mathbf{x}^*)^T \Sigma_S^{-1} k(\mathbf{x}^*) \tag{2}$$

where $G_{0,s} = [g_0(\mathbf{x}_1), \ldots, g_0(\mathbf{x}_n)]^T$ is the vector of the mean function at sample locations, $k(\mathbf{x}^*)$ is the covariance vector between $\mathbf{x}$ and sample locations X_s and Σ_s is the covariance matrix at sample locations. The conditional mean (1) serves as the predictor at location $\mathbf{x}$, and the prediction variance is given by (2). In practice, the mean function $g_0(\mathbf{x})$ has a parametric form $g_0(\mathbf{x}) = \sum_{j=1}^{J} \beta_j g_j(\mathbf{x}) = G(\mathbf{x})\beta$ where functions $G(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_J(\mathbf{x})]$ are known and $\beta = [\beta_1, \ldots, \beta_J]^T$ are regression parameters to be estimated. Moreover, R is chosen among a class of standard correlation functions (Gaussian, Matérn, ...) given up to some unknown hyperparameters ψ, corresponding to correlation lengths for example, see Rasmussen and Williams (2006). R is then denoted by R_ψ and $R_{\psi,s}$ is the correlation matrix at sample locations. As a result, in order to use the conditional expectation as a predictor, these parameters need to be estimated. Maximum likelihood estimators are usually preferred. For example, provided that ψ is known, regression parameters are obtained with the generalized least square estimator

$$\hat{\beta} = (G_s^T R_{\psi,S}^{-1} G_s)^{-1} G_s^T R_{\psi,s}^{-1} Y_s$$

and the maximum likelihood estimator of σ^2 is

$$\widehat{\sigma^2} = \frac{1}{n}(Y_s - G_s\hat{\beta})^T R_{\psi,s}^{-1}(Y_s - G_s\hat{\beta}).$$

In addition, the estimation of hyperparameters consists in solving the following minimization problem

$$\widehat{\psi} = \arg\min_{\psi} \widehat{\sigma^2}\, |R_{\psi,s}|^{\frac{1}{n}}$$

with $|A|$ denoting the determinant of matrix A. Consequently, the conditional field $\tilde{Y}(\mathbf{x}^*) = [Y(\mathbf{x}^*)|Y(X_s) = Y_s]$, given the estimated parameters, is a Gaussian field with mean $\tilde{\mu}(\mathbf{x}^*) = \mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)\right)$ given by

$$\tilde{\mu}(\mathbf{x}^*) = G(\mathbf{x}^*)\hat{\beta} + k(\mathbf{x}^*)^T \Sigma_s^{-1} \left(Y_s - G_s\hat{\beta}\right) \tag{3}$$

and covariance function equal to

$$\tilde{C}(\mathbf{x}, \mathbf{x}') = C(\mathbf{x}, \mathbf{x}') - k(\mathbf{x})^T \Sigma_s^{-1} k(\mathbf{x}') \tag{4}$$

Σ_s, k and C depending on the estimated parameters. Note that the covariance function has an additional component if the variance estimation on $\hat{\beta}$ is accounted for (see Santner et al. (2003)), but this case will not be considered here.

2.2 Constrained Gaussian process regression

In contrast to previous work on constrained kriging (Yoo and Kyriakidis (2006), Kleijnen and van Beers (2010), Riihimäki and Vehtari (2010)), we propose to keep the conditional expectation framework.

For instance, in the case of bound or monotonicity constraints (increasing constraint w.r.t. x^j) on a subset I of $\mathbb{R}^D$, the predictor $\tilde{\mu}(\mathbf{x}^*)$ is naturally replaced with the conditional expectation

$$\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\forall \mathbf{x}\in I, a\leq \tilde{Y}(\mathbf{x})\leq b\right) \text{ or } \mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\forall \mathbf{x}\in I, \frac{\partial \tilde{Y}}{\partial x^j}(\mathbf{x})\geq 0\right) \text{ resp.} \tag{5}$$

Unfortunately, no explicit formulations of these quantities are available. As a result, instead of imposing a constraint on a given subset, we proposed in Da Veiga and Marrel (2012) to discretize it into a (large) number of conditioning points. As an illustration, considering a set of N points $\mathbf{x}_1,\ldots,\mathbf{x}_N$ chosen in a subset I, the above conditional expectations are approximated by

$$\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\forall i=1,\ldots,N, a\leq \tilde{Y}(\mathbf{x}_i)\leq b\right) \text{ or } \mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\forall i=1,\ldots,N, \frac{\partial \tilde{Y}}{\partial x^j}(\mathbf{x}_i)\geq 0\right) \text{ resp.} \tag{6}$$

Note that this conditioning does not imply that the computer code g must be evaluated at points $\mathbf{x}_1,\ldots,\mathbf{x}_N$. More generally, in this work, any constraint defined on a subset will be replaced with its discrete-location counterpart.

Actually, the proposed framework can accommodate any linear constraint. We propose the following general formulation to account for K different constraint types, imposed on N_k points for $k=1,\ldots,K$:

$$\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\ \forall k=1,\ldots,K,\ \forall i=1,\ldots,N_k,\ a_i^{(k)}\leq Z^{(k)}(\mathbf{x}_i^{(k)})\leq b_i^{(k)}\right). \tag{7}$$

Here, $Z^{(k)}=\mathcal{L}^{(k)}\left[\tilde{Y}\right]$ where $\mathcal{L}^{(k)}$ is the linear operator corresponding to the k^{th} constraint type imposed on the constraint design $X^{(k)}=\left\{\mathbf{x}_1^{(k)},\ldots,\mathbf{x}_{N_k}^{(k)}\right\}$, $a_i^{(k)}=a^{(k)}(\mathbf{x}_i^{(k)})$ and $b_i^{(k)}=b^{(k)}(\mathbf{x}_i^{(k)})$ with $a^{(k)}$ and $b^{(k)}$ the bound functions for the k^{th} constraint type. For example, if we consider a $[0,1]$ bound and an increasing constraint w.r.t. the j^{th} input variable, we have: $K=2$, $Z^{(1)}=\tilde{Y}$, $Z^{(2)}=\partial\tilde{Y}/\partial x^j$ and four constant bound functions $a^{(1)}(\mathbf{x})=0$, $b^{(1)}(\mathbf{x})=1$, $a^{(2)}(\mathbf{x})=0$ and $b^{(2)}(\mathbf{x})=\infty$.

In the following, the simplified notation $\mathbf{Z}=\left(Z^{(1)}(\mathbf{x}_1^{(1)}),\ldots,Z^{(1)}(\mathbf{x}_{N_1}^{(1)}),\ldots,Z^{(K)}(\mathbf{x}_1^{(K)}),\ldots,Z^{(K)}(\mathbf{x}_{N_K}^{(K)})\right)^T$ is used and the corresponding bound vectors are denoted $\mathbf{a}=\left(a_1^{(1)},\ldots,a_{N_1}^{(1)},\ldots,a_1^{(K)},\ldots,a_{N_K}^{(K)}\right)^T$ and $\mathbf{b}=\left(b_1^{(1)},\ldots,b_{N_1}^{(1)},\ldots,b_1^{(K)},\ldots,b_{N_K}^{(K)}\right)^T$. The full constraint design of $N=\sum_{k=1}^K N_k$ points is denoted $\mathbf{X}=\left\{X^{(1)},\ldots,X^{(K)}\right\}$.

The linearity assumption of the constraints is crucial, in the sense that $\mathbf{W}=\left(\tilde{Y}(\mathbf{x}^*),\mathbf{Z}\right)$ is Gaussian if all the $Z^{(k)}$ for $k=1,\ldots,K$ are linear operators. In this setting, the constrained predictor $\hat{Y}(\mathbf{x}^*)=\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a}\leq\mathbf{Z}\leq\mathbf{b}\right)$ in (7) is then the expectation of a truncated normal distribution. The question now is to know how such truncated moments can be approximated.

2.2.1 Approximations of truncated moments

There is a large amount of literature dedicated to truncation of Gaussian vectors, the pioneering work being that of Tallis (Tallis (1961), Tallis (1963), Tallis (1965)). By definition, vector $\mathbf{Z}$ above is a N-dimensional Gaussian vector with probability density function (pdf) given by

$$\phi_{\mu_\mathbf{Z},\Sigma_\mathbf{Z}}(\mathbf{z})=\frac{1}{(2\pi)^{N/2}|\Sigma_\mathbf{Z}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{z}-\mu_\mathbf{Z})^T\Sigma_\mathbf{Z}^{-1}(\mathbf{z}-\mu_\mathbf{Z})\right),\mathbf{z}\in\mathbb{R}^N.$$

with mean $\mu_{\mathbf{Z}} \in \mathbb{R}^N$ and covariance matrix $\Sigma_{\mathbf{Z}}$. By definition of $\mathcal{L}^{(k)}$, $\mu_{\mathbf{Z}}$ can be written as $\mu_{\mathbf{Z}} = \left[{\mu^{(1)}}^T \dots {\mu^{(K)}}^T\right]^T$ with $\mu^{(k)} = \mathbb{E}\left(Z^{(k)}(X^{(k)})\right) = \mathbb{E}\left(\mathcal{L}^{(k)}\left[Y\right]\left(X^{(k)}\right)|Y(X_s) = Y_s\right)$. $\Sigma_{\mathbf{Z}}$ is a $K \times K$ block matrix with each block given by $\Sigma_{\mathbf{Z},k,l} = \mathrm{Cov}\left(Z^{(k)}(X^{(k)}), Z^{(l)}(X^{(l)})\right) = \mathrm{Cov}\left(\mathcal{L}^{(k)}\left[Y\right]\left(X^{(k)}\right), \mathcal{L}^{(l)}\left[Y\right]\left(X^{(l)}\right)|Y(X_s) = Y_s\right)$. Since by linearity $\left(\mathcal{L}^{(k)}\left[Y\right]\left(X^{(k)}\right), Y_S\right)$ is Gaussian, $\mu_{\mathbf{Z}}$ and $\Sigma_{\mathbf{Z}}$ are computed by classical Gaussian regression formulas, as in eqs (1) and (2).

Following Tallis (1961), the truncated expectation of $\mathbf{Z}$ is given by

$$\forall i = 1, \dots, N \quad \mathbb{E}(Z_i|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}) = \mu_i + \sum_{j=1}^{N} \sigma_{ij}\left(f_j(a_j) - f_j(b_j)\right) \tag{8}$$

where $\mu_i = (\mu_{\mathbf{Z}})_i$, $\sigma_{ij} = (\Sigma_{\mathbf{Z}})_{ij}$ and

$$f_i(z) = \int_{a_1}^{b_1} \dots \int_{a_{i-1}}^{b_{i-1}} \int_{a_{i+1}}^{b_{i+1}} \dots \int_{a_N}^{b_N} \phi_{\mu,\Sigma,\mathbf{a},\mathbf{b}}(z_1, \dots, z_{i-1}, z, z_{i+1}, \dots, z_N) d\mathbf{z}_{-i} \tag{9}$$

where ϕ is the truncated Gaussian pdf

$$\phi_{\mu,\Sigma,\mathbf{a},\mathbf{b}}(\mathbf{z}) = \begin{cases} \frac{\phi_{\mu,\Sigma}(\mathbf{z})}{\mathbb{P}(\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b})} & \text{for } \mathbf{a} \leq \mathbf{z} \leq \mathbf{b}, \\ 0 & \text{otherwise,} \end{cases}$$

with $\phi_{\mu,\Sigma}$ the pdf of the multivariate normal distribution of mean μ and covariance matrix Σ. Similar results are available in Tallis (1961) for the truncated covariance of $\mathbf{Z}$. Remark that Formula (8) involves the computation of normal integrals of dimension $N-1$. When N is large, as should be the case in the discrete-location method we propose, it is necessary to have powerful algorithms capable of producing accurate approximations. In Da Veiga and Marrel (2012) we investigated three dedicated numerical approaches:

- The first one is based on a simple correlation-free approximation. More precisely $\Sigma_{\mathbf{Z}}$ is assumed to be diagonal, thus yielding truncated moments approximated by

$$\begin{aligned} \mathbb{E}(Z_i|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}) &\approx \mathbb{E}(Z_i|a_i \leq Z_i \leq b_i) && (10) \\ &\approx \mu_i + \sigma_i \frac{\phi(\tilde{a}_i) - \phi(\tilde{b}_i)}{\Phi(\tilde{b}_i) - \Phi(\tilde{a}_i)} && (11) \\ \mathrm{Var}(Z_i|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}) &\approx \mathrm{Var}(Z_i|a_i \leq Z_i \leq b_i) && (12) \\ &\approx \sigma_i^2 \left[1 + \frac{\tilde{a}_i\phi(\tilde{a}_i) - \tilde{b}_i\phi(\tilde{b}_i)}{\Phi(\tilde{b}_i) - \Phi(\tilde{a}_i)} - \left(\frac{\phi(\tilde{a}_i) - \phi(\tilde{b}_i)}{\Phi(\tilde{b}_i) - \Phi(\tilde{a}_i)}\right)^2\right] && (13) \end{aligned}$$

  where $\tilde{a}_i = (a_i - \mu_i)/\sigma_i$ and $\tilde{b}_i = (b_i - \mu_i)/\sigma_i$ with $\sigma_i = \sqrt{\sigma_{ii}}$, ϕ and Φ denoting the pdf and cumulative density function of standard normal distribution, respectively.

- The two other ones aim at producing a numerical approximation of Formula (8) with two complementary tools:
  - Gibbs sampling of truncated normal distribution (Robert (1995)) coupled with MCMC, where moments are estimated by their empirical counterpart on samples of the distribution. The implementation is fully detailed in Da Veiga and Marrel (2012);
  - Numerical integration tools introduced by Genz (Genz (1992), Genz (1993)). It relies on a preliminary Cholesky decomposition of $\Sigma_{\mathbf{Z}}$ and successive mono-dimensional integrations with a Quasi Monte-Carlo procedure. The algorithm complexity depends on the dimension N (the number of constraints) and on the size of the Quasi Monte-Carlo sample denoted by q. More precisely, the complexity is given by $O(N^3 + Nq)$, where the $O(N^3)$ term corresponds to the Cholesky decomposition and $O(Nq)$ to mono-dimensional integrations.

Several numerical tests on analytical functions with bounds, monotonicity and convexity constraints were performed in Da Veiga and Marrel (2012). They reveal that Genz and sampling approximations produce the same well-behaved constrained predictors, Genz's method being faster in practice especially when the number of constraint points increases. Both outperform the correlation-free formula, but the latter yields relatively accurate predictions given that its computational time is considerably lower and increases much more slowly with the number of constraints. For these reasons, we will focus here on Genz's method to compute the final constrained metamodel while relying on the correlation-free approximation when intensive computations are required as in our sequential strategy, see Section 3.1.

2.2.2 Algorithm

From a practical perspective, recall that our goal is to compute $\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right)$. As an intermediate step, remark that the Gaussian vector $\mathbf{W} = \left(\tilde{Y}(\mathbf{x}^*), \mathbf{Z}\right)$ follows the normal distribution

$$\mathbf{W} = \left(\tilde{Y}(\mathbf{x}^*), \mathbf{Z}\right) \sim \mathcal{N}\left(\begin{bmatrix} \tilde{\mu}(\mathbf{x}^*) \\ \mu_{\mathbf{Z}} \end{bmatrix}, \begin{bmatrix} \tilde{C}(\mathbf{x}^*, \mathbf{x}^*) & \Sigma_{\tilde{Y}\mathbf{Z}} \\ \Sigma_{\tilde{Y}\mathbf{Z}}^T & \Sigma_{\mathbf{Z}} \end{bmatrix}\right) \tag{14}$$

where $\Sigma_{\tilde{Y}\mathbf{Z}} = \left[\Sigma_{\tilde{Y}\mathbf{Z}}^{(1)}, \ldots, \Sigma_{\tilde{Y}\mathbf{Z}}^{(K)}\right]$ with $\Sigma_{\tilde{Y}\mathbf{Z}}^{(k)} = \mathrm{Cov}\left(\tilde{Y}(\mathbf{x}^*), \mathbf{Z}^{(k)}\right) = \mathrm{Cov}\left(\tilde{Y}(\mathbf{x}^*), \mathcal{L}^{(k)}[Y]\left(X^{(k)}\right)|Y(X_s) = Y_s\right)$ for $k = 1, \ldots, K$.

From a result on general truncation given in Kotz et al. (2000), the first moments of $\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}$ are equal to

$$\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right) = \tilde{\mu}(\mathbf{x}^*) + \Sigma_{\tilde{Y}\mathbf{Z}}\Sigma_{\mathbf{Z}}^{-1}\left(\nu_{\mathbf{Z}} - \mu_{\mathbf{Z}}\right) \tag{15}$$

$$\mathrm{Var}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right) = \tilde{C}(\mathbf{x}^*, \mathbf{x}^*) - \Sigma_{\tilde{Y}\mathbf{Z}}\left(\Sigma_{\mathbf{Z}}^{-1} - \Sigma_{\mathbf{Z}}^{-1}\Gamma_{\mathbf{Z}}\Sigma_{\mathbf{Z}}^{-1}\right)\Sigma_{\tilde{Y}\mathbf{Z}}^T \tag{16}$$

where $\nu_{\mathbf{Z}} = \mathbb{E}(\mathbf{Z}|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b})$ and $\Gamma_{\mathbf{Z}} = \mathrm{Var}(\mathbf{Z}|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b})$ are the mean and covariance matrix of the truncated normal vector $\mathbf{Z}$. Once $\nu_{\mathbf{Z}}$ and $\Gamma_{\mathbf{Z}}$ are computed via Tallis formula and Genz's approximation (Section 2.2.1), the constrained predictor (15) and its associated variance (16) are easily deduced. The whole estimation procedure is summarized in Algorithm 1.

Algorithm 1: Constrained GP prediction at $\mathbf{x}^*$

Given X_s, Y_s, $f_0(\mathbf{x})$, $R_\psi(\mathbf{x} - \mathbf{x}')$, $\mathcal{L}^{(1)}, \ldots, \mathcal{L}^{(K)}$, $\mathbf{X} = \left\{X^{(1)}, \ldots, X^{(k)}\right\}$, $\mathbf{a}$, $\mathbf{b}$

(1) Estimate the GP parameters β, σ and ψ.

(2) Distribution of $\mathbf{W} = \left(\tilde{Y}(\mathbf{x}^*), \mathbf{Z}\right)$ (Equation (14))

  (a) Build the Gaussian vector $\left(\tilde{Y}(\mathbf{x}^*), \mathcal{L}^{(1)}[Y]\left(X^{(1)}\right), \ldots, \mathcal{L}^{(K)}[Y]\left(X^{(K)}\right), Y_s\right)$

  (b) By conditioning w.r.t. Y_s, compute $\tilde{\mu}(\mathbf{x}^*)$ and $\tilde{C}(\mathbf{x}^*, \mathbf{x}^*)$, $\mu_{\mathbf{Z}}$, $\Sigma_{\mathbf{Z}}$ and $\Sigma_{\tilde{Y}\mathbf{Z}}$ (formulas for conditional moments of Gaussian vector, Equations (3) and (4)).

(3) Compute the truncated moments $\nu_{\mathbf{Z}}$ and $\Gamma_{\mathbf{Z}}$ with Tallis formula and Genz's approximation.

(4) Build the final constrained predictor and variance by computing the truncated moments $\mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right)$ and $\mathrm{Var}\left(\tilde{Y}(\mathbf{x}^*)|\mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right)$ with equations (15) and (16).

The only tedious part in the algorithm is to evaluate the covariance between the GP process Y and the constraint processes $\mathcal{L}^{(k)}[Y]$ for $k = 1, \ldots, K$ in step 2.b. Formulas for derivatives are available for example in Cramér and Leadbetter (1967).

2.2.3 Parameter estimation and generalizations

As mentioned in Section 2.1, regression parameters β, variance parameter σ^2 and covariance hyperparameters ψ are preliminarily estimated by maximizing the unconditional likelihood of the observations. In other words, all the above constraints are not used in this estimation process and are included in subsequent predictions exclusively. In the context of bound truncation of multinormal variables, Griffiths (2002) proposed a Gibbs sampling approach for mean and covariance estimation. However this work cannot be used as it is based on a false assumption (it is assumed that incomplete[1] conditional distributions of a truncated normal distribution are truncated normal distributions, which they are not; only full conditional distributions are). To the best of our knowledge, no other approaches was developed for general linear constraints and covariance hyperparameters. We hope to be able to address conditional maximum likelihood estimation in future research.

In Da Veiga and Marrel (2012) we also discussed convexity integral constraints and we would like to point out that we can accommodate any linear constraint, which includes divergence or curl sign information when working with multidimensional outputs. Indeed, if we assume for example that the output code is a vector field $\left(Y^1(x_1,x_2), Y^2(x_1,x_2)\right)$, one can build a 2-D GP surrogate and imposes a divergence constraint by using $\mathcal{L}^{(1)}\left[\tilde{Y^1}, \tilde{Y^2}\right] = \frac{\partial \tilde{Y^1}}{\partial x_1} + \frac{\partial \tilde{Y^2}}{\partial x_2}$, and similarly for a curl. In this sense, we can generalize the work of Scheuerer and Schlather (2012) where only equality constraints on the divergence and the curl are considered.

2.3 Numerical illustration

Since additional examples are given later in the paper, we illustrate here the behavior of the constrained predictors only on a simple one-dimensional function g_1 given by

$$g_1(x) = \frac{\sin(10\pi x^{5/2})}{10\pi x}$$

for $x \in [0,1]$. This function is quite difficult to approximate because it has a frequency which exhibits strong variations on $[0,1]$. We assume that n observations $(x_i, y_i = g_1(x_i))_{i=1,\ldots,n}$ are available, where the x_i are sampled according to the uniform distribution on $[0,1]$. These observations are used to build the conditional field $\tilde{Y}(\mathbf{x})$ and the corresponding unconstrained kriging predictor $\tilde{\mu}(\mathbf{x}^*) = \mathbb{E}\left(\tilde{Y}(\mathbf{x}^*)\right)$ introduced in Section 2.1. The mean function $g_0(\mathbf{x})$ is assumed to be constant, a Gaussian covariance function is used (which is sufficiently differentiable to account for convexity constraints) and its hyperparameters are estimated by maximum likelihood.

Using Algorithm 1, we build the constrained predictors with four configurations involving three different types of constraints:

- Bound constraints only at the N locations (sign of g_1);
- Derivative constraints only at the N locations (sign of g_1')
- Bound and derivative constraints at the N locations (sign of g_1 and g_1');
- Bound, derivative and convexity constraints at the N locations (sign of g_1, g_1' and g_1'');.

To evaluate the accuracy of the metamodels, we use the predictivity coefficient Q^2. It is the determination coefficient R^2 computed from a test sample (composed here by $n_{\text{test}} = 100$ equally spaced points):

$$Q^2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{n_{\text{test}}}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n_{\text{test}}}\left(Y_i - \bar{Y}\right)^2},$$

[1] All the elements are not constrained.

where Y denotes the n_{test} true observations (or exact values) of the test set, $\bar{Y}$ their empirical mean and $\hat{Y}$ the metamodel predicted values. We consider a learning sample of $n = 15$ observations and, for each constraint type, $N = 20$ constraint locations equally spaced on $[0, 1]$ are chosen. The predictions are performed on a set of 100 points chosen uniformly on $[0, 1]$ and we repeat this procedure 100 times, varying the learning sample. The mean and the standard deviation of Q^2 are given in Table 1.

Constraints	Q^2
No	0.62 +/- 0.20
Bounds only	0.79 +/- 0.19
Derivatives only	0.80 +/- 0.19
Bounds + derivatives	0.80 +/- 0.23
Bounds + derivatives + convexity	0.85 +/- 0.19

Table 1: Mean and standard deviation of Q^2 when accounting for several constraints for function g_1.

When we integrate more constraints, the accuracy of predictions is increasing, as expected. In Figure 1, we show the cumulative incorporation of bounds, first derivatives and convexity for $n = 12$ and $N = 20$ on g_1, as well as the corresponding predictor variance. While the unconstrained predictor fails at retrieving information between observed points ($Q^2 = 0.43$), constraints greatly help reconstruct the function in these regions. Bound constraints start by improving predictions on the right part ($Q^2 = 0.84$ in Figure 1, top). First-derivative constraints further enhance the approximation on the left part ($Q^2 = 0.95$ in Figure 1, middle), but deteriorate it on the right. This phenomenon is finally compensated by second-order derivative constraints ($Q^2 = 0.98$ in Figure 1, bottom). The predictor variance is first largely reduced with bound constraints, and one can observe further reduction with derivative ones. Convexity has a small impact on variance reduction, since there is only a slight improvement between predictions with first- and second-order derivative constraints.

3 Designs for constraint locations

Results showed that incorporation of constraints with our algorithm greatly improve predictions. The final constrained predictor is based on a discrete-location approximation of conditional expectations. Since they involve computation of integrals of dimensionality equal to the number of constraints, we proposed a numerical approximation based on Genz' method.

However, its complexity heavily depends on the number of constraints points. When dealing with common industrial problems, this approach will require many points and subsequent integral approximations could suffer from the curse of dimensionality if a naive design for constraint locations is used (e.g. points on a grid).

To face this limitation, a solution could be to use a space-filling design such as Latin Hypercube Sampling (LHS) with optimal space covering properties (e.g. maximin LHS, optimal discrepancy LHS, ... see Fang et al. (2006) or Johnson et al. (1990) for more details). This strategy will serve as a baseline approach for all subsequent numerical comparison. But this is a blind strategy: constraint locations do not depend on the regions where the standard GP predictor respects the constraints or not. Ideally, we expect to put more constraint points where the constraints are not satisfied.

For this, given the total number of constraint points $N = \sum_{k=1}^{K} N_k$, intuitively an optimal design $\mathbf{X}^* = \left\{X^{(1)*}, \ldots, X^{(K)*}\right\}$ should be the one that minimizes the maximal probability of not respecting the constraints on the whole input domain:

$$\mathbf{X}^* = \text{Argmin}_{X=\bigcup_k X^{(k)}, \sum_k |X^{(k)}|=N} \max_{\mathbf{x}\in\mathbb{R}^D} 1 - P\left(\mathbf{x}; \left\{X^{(1)}, \ldots, X^{(K)}\right\}\right) \tag{17}$$

where

$$P\left(\mathbf{x}; \left\{X^{(1)}, \ldots, X^{(K)}\right\}\right) = \mathbb{P}\left(\forall k = 1, \ldots, K,\ a^{(k)}(\mathbf{x}) \leq \mathcal{L}^{(k)}\left[\tilde{Y}\right](\mathbf{x}) \leq b^{(k)}(\mathbf{x}) \mid \mathbf{a} \leq \mathbf{Z} \leq \mathbf{b}\right). \tag{18}$$

This criterion is unfortunately out of reach, since both the combinatorics and the computational cost for a given design $\mathbf{X}$ are large. To a lesser extent, the same phenomenon arises in standard GP regression with the Integrated Mean Squared Error criterion. Following previous work on adaptive learning designs for GP regression, a possible remedy lies in a sequential strategy which is detailed below.

3.1 Adaptive sequential strategy

We start from a current constraint design $\mathbf{X}_{\text{cur}} = \left\{X^{(1)}, \ldots, X^{(K)}\right\}$ of $N_{\text{cur}} = \sum_{k=1}^{K} N_{k,\text{cur}}$ points. Using similar notations as in 2.2, the corresponding constraint and bound vectors are denoted $\mathbf{Z}_{\text{cur}}$, $\mathbf{a}_{\text{cur}}$ and $\mathbf{b}_{\text{cur}}$, respectively. The additional locations are chosen where $P(\mathbf{x}; \mathbf{X}_{\text{cur}})$, the probability that the constrained predictor built with $\mathbf{X}_{\text{cur}}$ respects the constraints, is minimal. If we look closely at this probability, we see that it involves the distribution of a truncated multivariate normal distribution. Even if it is possible to get an approximation with Genz's method, this means that for each location to be tested Genz's algorithm should be processed. This is intractable when the input domain lies in $\mathbb{R}^D$.

To go beyond, we propose a crude approximation of this probability based on a normal assumption. More precisely, we approximate $P(\mathbf{x}; \mathbf{X}_{\text{cur}})$ with

$$\hat{P}(\mathbf{x}; \mathbf{X}_{\text{cur}}) = \int_{a^{(1)}(\mathbf{x})}^{b^{(1)}(\mathbf{x})} \cdots \int_{a^{(K)}(\mathbf{x})}^{b^{(K)}(\mathbf{x})} \phi_{\mu_{\mathcal{L}}(\mathbf{x}), \Sigma_{\mathcal{L}}(\mathbf{x})}(\mathbf{t})\, d\mathbf{t} \tag{19}$$

where

$$\mu_{\mathcal{L}}^{k}(\mathbf{x}) = \mathbb{E}\left(\mathcal{L}^{(k)}\left[\tilde{Y}\right](\mathbf{x}) \mid \mathbf{a}_{\text{cur}} \leq \mathbf{Z}_{\text{cur}} \leq \mathbf{b}_{\text{cur}}\right) \tag{20}$$

$$\Sigma_{\mathcal{L},k,k'}(\mathbf{x}) = \text{Cov}\left(\mathcal{L}^{(k)}\left[\tilde{Y}\right](\mathbf{x}), \mathcal{L}^{(k')}\left[\tilde{Y}\right](\mathbf{x}) \mid \mathbf{a}_{\text{cur}} \leq \mathbf{Z}_{\text{cur}} \leq \mathbf{b}_{\text{cur}}\right). \tag{21}$$

Since the number K of different constraint types will usually be small, computing the integral is straightforward once the moments of the inner Gaussian are given. Thus, the only stake lies in evaluating $\mu_{\mathcal{L}}(\mathbf{x})$ and $\Sigma_{\mathcal{L}}(\mathbf{x})$. For this, we can easily extend formulas (15) and (16) to $\mathcal{L}^{(k)}\left[\tilde{Y}\right]$ where the computational burden consists in approximating $\nu_{\mathbf{Z}}$ and $\Gamma_{\mathbf{Z}}$. As explained in Section 2.2.1, one can use Genz's integration method or the correlation-free approximation, the former being more precise while the latter is much faster. Once all the new constraint points are identified, the final predictor is computed using Genz's formula as in Algorithm 1 from Section 2.2.2. The resulting complete adaptive procedure is recapped in Algorithm 2.

Algorithm 2: Adaptive sequential strategy for constrained GP prediction

Given X_s, Y_s, $g_0(\mathbf{x})$, $R_\psi(\mathbf{x}-\mathbf{x}')$, $\mathcal{L}^{(1)},\ldots,\mathcal{L}^{(K)}$, N

Initialization

- Estimate the GP parameters β, σ and ψ
- Find $\mathbf{x}^* = \text{Argmin}_{\mathbf{x}\in\mathbb{R}^D} \hat{P}(\mathbf{x};\emptyset)$, the point where the standard GP predictor has the highest probability of violating the constraints
- Set $\mathbf{X}_{\text{cur}} = \mathbf{x}^*$

Sequential loop
for $iter = 1,\ldots,N-1$
do

- Find $\mathbf{x}^* = \text{Argmin}_{\mathbf{x}\in\mathbb{R}^D} \hat{P}(\mathbf{x};\mathbf{X}_{\text{cur}})$.
  Probability $\hat{P}()$ can be computed with either Genz's integration method or the correlation-free approximation
- $\mathbf{X}_{\text{cur}} = \mathbf{X}_{\text{cur}} \cup \mathbf{x}^*$, $iter = iter + 1$

endfor

Final constrained predictor and variance
Use Algorithm 1 with $\mathbf{X} = \mathbf{X}_{\text{cur}}$

3.2 Numerical comparisons

For all examples, we will compare the constrained predictors given by three different designs for constraint locations:

- LHS with optimal maximin properties (in one-dimension, equivalent to equally spaced);
- sequential strategy based on correlation-free approximation;
- sequential strategy based on Genz's method.

The efficiency of the strategies is evaluated w.r.t. the Q^2 criterion, but also w.r.t. the percentage (in $[0,1]$) of constraint respect α of the final predictor on a test sample of size 200. This second indicator is especially useful when the constraints are not directly on the function, but on one of its derivative for example. For each numerical test, each experiment is repeated 50 times and report the mean and standard deviation of Q^2 and α accordingly.

3.2.1 Illustration on 1-D example with bound and monotonicity constraints

First, we come back to function g_1 already introduced in Section 2.3. Figure 2 illustrates where the sequential strategies tend to add points. From a given learning sample of $n = 20$ points randomly planned (grey triangles in Figure 2), two constrained predictors based on $N = 10$ constraint points are built following two strategies: the non-sequential one with constraint points equally spaced and the sequential one based on correlation-free approximation. The constraint points successively added with the second strategy are represented by red full circles. Unsurprisingly, the sequential algorithm detects the regions where the predictor does not respect the constraints and adds constraint points in these regions, which significantly improves the

constrained predictor.

To confirm these results, we repeat the experiment (50 times) for different values of n and N. Tables 2 and 3 summarize the results for bound constraints obtained with a Matérn 3/2 and a Gaussian covariance function, respectively. First observe that the Matérn 3/2 covariance consistently surpasses the Gaussian one. Then, as expected, the addition of constraints improves the predictivity with all three methods. But more importantly, the sequential strategies yield better results than the static approach, even if the gap vanishes when n and N. Note also that despite the crude approximation of the correlation-free sequential method, the constraint points are sufficiently well-placed to obtain a final prediction with the same quality as the greedy method involving Genz's computations.

Figure 3 left is another representation of these results, obtained with $n = 10$, a Matérn 3/2 covariance function and for a number of constraint N up to 100. Note that only the correlation-free sequential method is represented, Genz's sequential method yielding similar results. To further analyze the models and check that they actually respect the constraints on the whole domain, we report in Figure 3 right the percentage α of independent points (i.e. neither in the experimental design nor in the constraints set) which respect the constraints. Again, sequential strategies perform better, even achieving a high level of accuracy with only 20 constraint points.

Finally, we compare the computational burden of Genz' approximation w.r.t. the correlation-free approach in Figure 4. While the cost grows very rapidly for Genz, it is only linear when we neglect correlations.

n	N	Unconstrained	Optimal LHS	Correlation-free sequential	Genz sequential
8	10	0.50 ±0.09	0.73 ±0.05	0.78 ±0.08	0.78 ±0.08
	20	0.50 ±0.09	0.81 ±0.04	0.82 ±0.07	0.82 ±0.07
	30	0.50 ±0.09	0.85 ±0.03	0.86 ±0.03	0.87 ±0.04
	40	0.50 ±0.09	0.86 ±0.03	0.87 ±0.03	0.87 ±0.03
	50	0.50 ±0.09	0.86 ±0.03	0.87 ±0.03	0.88 ±0.03
10	10	0.66 ±0.08	0.75 ±0.07	0.88 ±0.09	0.89 ±0.09
	20	0.66 ±0.08	0.87 ±0.05	0.89 ±0.09	0.90 ±0.08
	30	0.66 ±0.08	0.89 ±0.04	0.92 ±0.04	0.92 ±0.05
	40	0.66 ±0.08	0.90 ±0.04	0.92 ±0.04	0.92 ±0.04
	50	0.66 ±0.08	0.91 ±0.04	0.92 ±0.04	0.92 ±0.04
20	10	0.94 ±0.02	0.96 ±0.02	0.98 ±0.01	0.98 ±0.01
	20	0.94 ±0.02	0.97 ±0.01	0.98 ±0.01	0.98 ±0.01
	30	0.94 ±0.02	0.97 ±0.01	0.98 ±0.01	0.98 ±0.01
	40	0.94 ±0.02	0.98 ±0.01	0.98 ±0.01	0.98 ±0.01
	50	0.94 ±0.02	0.98 ±0.01	0.98 ±0.01	0.98 ±0.01

Table 2: Function g_1 – Mean and standard deviation of Q^2 with **bound constraints**, for different values of n and N with a **Matérn 3/2 covariance** function.

We now repeat this experiment but with derivative constraints on g_1 instead. Table 4 reports the results obtained with a Matérn 3/2 covariance function. The analysis of Q^2 shows that the incorporation of derivative information improves much further the model than bound constraints. This phenomenon makes sense, since this is a much more informative constraint for the model. Moreover, the two sequential strategies, which provide similar results, again significantly outperform the non-sequential strategy in terms of both predictor accuracy Q^2 and percentage of constraint respect α.

Figure 5 illustrates these results with $n = 10$. Figure 5 right also provides an additional insight: respecting the derivative constraint is harder than the bounds and the sequential addition of constraint points helps improve the model more slowly, but each addition of such information enhances the predictivity faster.

n	N	Unconstrained	Optimal LHS	Correlation-free sequential	Genz sequential
8	10	0.38 ±0.17	0.57 ±0.14	0.61 ±0.20	0.61 ±0.21
	20	0.38 ±0.17	0.64 ±0.15	0.68 ±0.17	0.68 ±0.17
	30	0.38 ±0.17	0.69 ±0.11	0.72 ±0.14	0.72 ±0.14
	40	0.38 ±0.17	0.70 ±0.10	0.77 ±0.13	0.78 ±0.12
10	10	0.59 ±0.17	0.66 ±0.14	0.72 ±0.21	0.72 ±0.21
	20	0.59 ±0.17	0.73 ±0.14	0.75 ±0.18	0.74 ±0.18
	30	0.59 ±0.17	0.73 ±0.11	0.79 ±0.14	0.78 ±0.14
	40	0.59 ±0.17	0.73 ±0.09	0.81 ±0.13	0.80 ±0.13
20	10	0.94 ±0.03	0.95 ±0.02	0.98 ±0.01	0.98 ±0.01
	20	0.94 ±0.03	0.95 ±0.01	0.98 ±0.01	0.98 ±0.01
	30	0.94 ±0.03	0.97 ±0.01	0.98 ±0.01	0.98 ±0.01
	40	0.94 ±0.03	0.98 ±0.01	0.98 ±0.01	0.98 ±0.01

Table 3: Function g_1 – Mean and standard deviation of Q^2 with **bound constraints**, for different values of n and N with a **Gaussian covariance** function.

n	N	Unconstrained		Optimal LHS		Correlation-free sequential		Genz sequential	
		Q^2	α	Q^2	α	Q^2	α	Q^2	α
8	10	0.50 ±0.09	0.71 ±0.06	0.57 ±0.09	0.72 ±0.05	0.72 ±0.07	0.84 ±0.07	0.72 ±0.07	0.84 ±0.07
	20	0.50 ±0.09	0.71 ±0.06	0.63 ±0.08	0.71 ±0.06	0.75 ±0.08	0.92 ±0.06	0.76 ±0.08	0.92 ±0.06
	30	0.50 ±0.09	0.71 ±0.06	0.79 ±0.07	0.82 ±0.05	0.81 ±0.06	0.97 ±0.03	0.83 ±0.06	0.97 ±0.03
	40	0.50 ±0.09	0.71 ±0.06	0.84 ±0.06	0.86 ±0.05	0.84 ±0.04	0.97 ±0.02	0.84 ±0.04	0.97 ±0.02
10	10	0.66 ±0.08	0.75 ±0.06	0.73 ±0.08	0.80 ±0.06	0.83 ±0.06	0.92 ±0.07	0.83 ±0.06	0.92 ±0.07
	20	0.66 ±0.08	0.75 ±0.06	0.77 ±0.07	0.81 ±0.07	0.89 ±0.06	0.96 ±0.06	0.90 ±0.06	0.96 ±0.06
	30	0.66 ±0.08	0.75 ±0.06	0.86 ±0.06	0.87 ±0.07	0.89 ±0.05	0.98 ±0.03	0.89 ±0.05	0.98 ±0.02
	40	0.66 ±0.08	0.75 ±0.06	0.90 ±0.05	0.91 ±0.05	0.90 ±0.04	0.99 ±0.02	0.90 ±0.04	0.99 ±0.02
20	10	0.94 ±0.02	0.89 ±0.02	0.96 ±0.02	0.99 ±0.01	0.98 ±0.01	0.92 ±0.02	0.98 ±0.01	0.99 ±0.01
	20	0.94 ±0.02	0.89 ±0.02	0.90 ±0.02	0.90 ±0.03	0.98 ±0.01	0.99 ±0.01	0.98 ±0.01	0.99 ±0.01
	30	0.94 ±0.02	0.89 ±0.02	0.97 ±0.01	0.93 ±0.02	0.98 ±0.01	0.99 ±0.01	0.98 ±0.01	0.99 ±0.01
	40	0.94 ±0.02	0.89 ±0.02	0.98 ±0.01	0.94 ±0.02	0.98 ±0.01	0.99 ±0.01	0.98 ±0.01	0.99 ±0.01

Table 4: Function g_1 – Mean and standard deviation of Q^2 and α with **derivative constraints**, for different values of n and N with a **Matérn 3/2 covariance**.

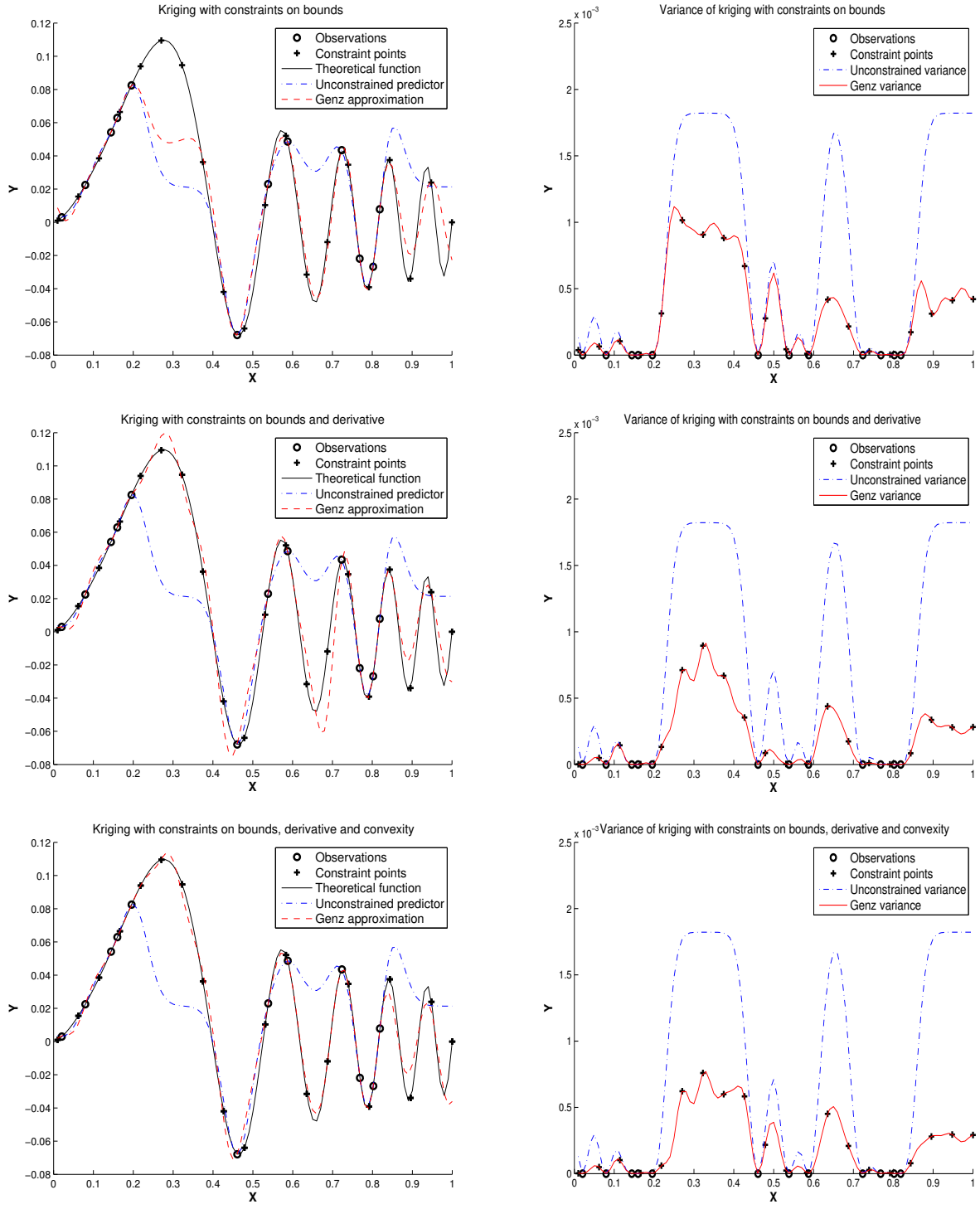

Figure 1: Function g_1 – Unconstrained and constrained predictor (left) and predictor variance (right) accounting for constraints on bounds only (top), on bounds and derivatives (middle) and on bounds, derivatives and convexity (bottom), with a Gaussian covariance function.

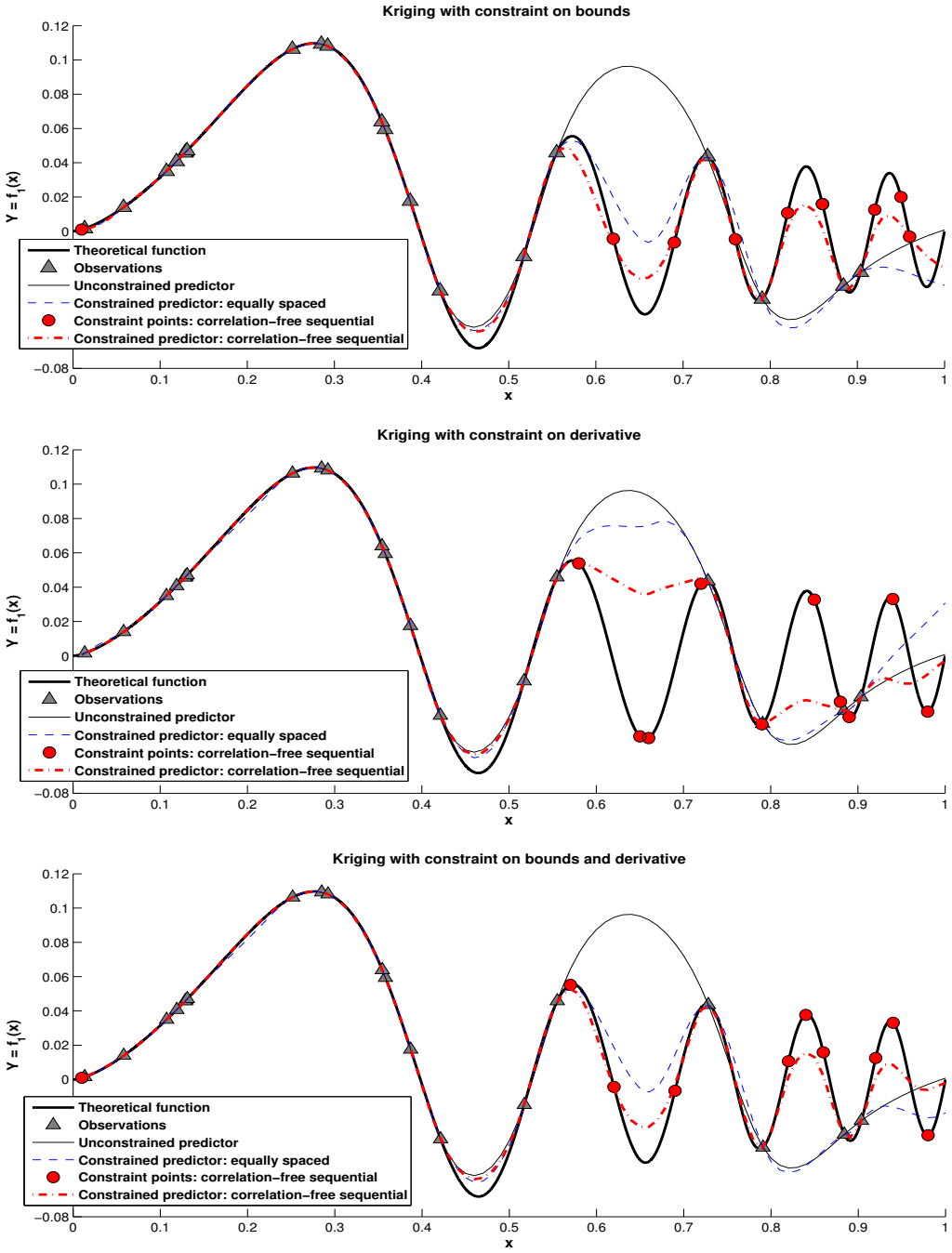

Figure 2: Function g_1 – Unconstrained and constrained predictor accounting for constraints on bounds only (top), on derivatives (middle) and on both bounds and derivatives (bottom), with a Matérn 3/2 covariance.

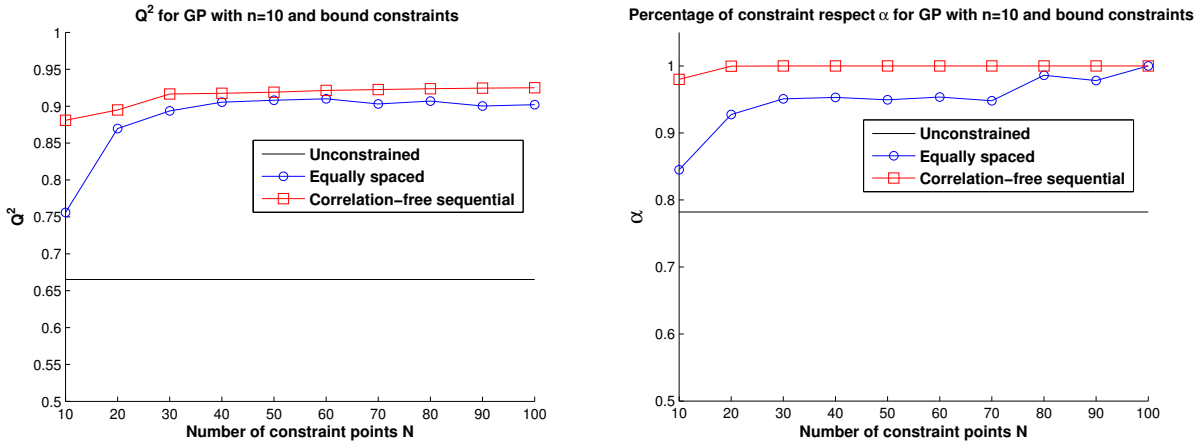

Figure 3: Function g_1 – Mean of Q^2 (left) and α (right) of the unconstrained and constrained predictors with $n = 10$ observations and **bound constraints**, with a **Matérn 3/2 covariance** function.

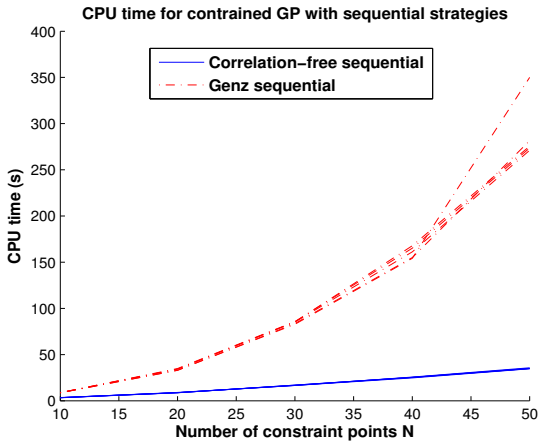

Figure 4: Function g_1 – Mean of CPU time for constrained predictors with $n = 8, 10, \ldots, 50$ observations and **bound constraints**, with a **Matérn 3/2 covariance** function.

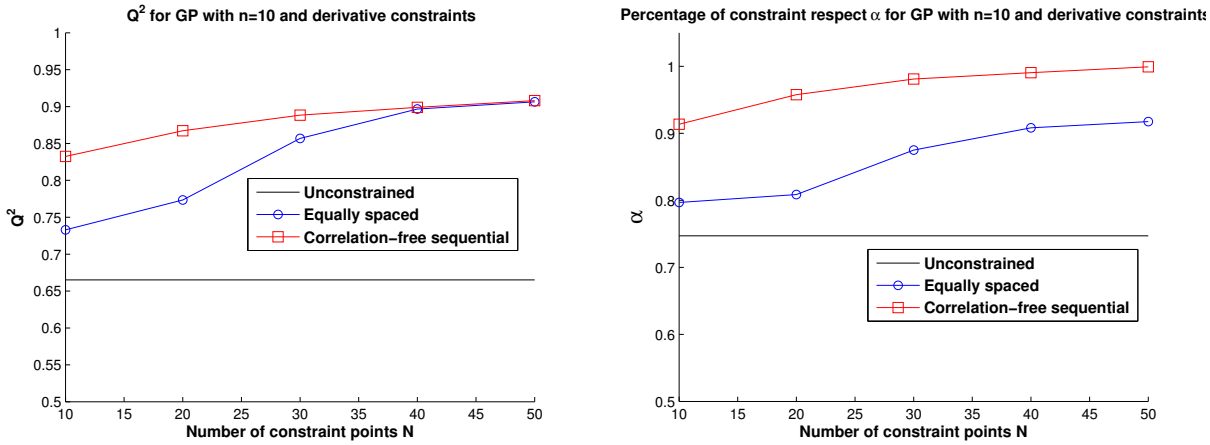

Figure 5: Function g_1 – Mean of Q^2 (left) and α (right) of the unconstrained and constrained predictors with $n = 10$ observations and **derivative constraints** with a **Matérn 3/2 covariance**.

3.2.2 Illustration on 2-D example with bound constraint

Consider now the two-dimensional function g_2 defined on $[0,1]^2$:

$$g_2(x_1, x_2) = 1.8x_1 + 5\log\left(g_1(x_2) + 2\right) + 10x_2 g_1(x_1) - 4.5,$$

which exhibits strong interactions between the two input variables and is a challenging function to approximate. The results for bound constraints are reported in Table 5 for a Matérn 3/2 covariance function. While the unconstrained predictor has a very low Q^2, once again the addition of constraints greatly improves the model quality, especially with the sequential strategies. Note for example that we can reach a $Q^2 = 0.7$ in this case with only $n = 50$ and $N = 100$ while the unconstrained predictor has a $Q^2 = 0.37$ and can only reach a $Q^2 = 0.68$ with $n = 130$. Similarly, the percentage of points respecting the constraints grow rapidly with the number of constraint points and closely approaches 100% when $N = 100$, see Figure 6, right.

Finally, the computational time between the two sequential strategies is compared in Figure 7. Again, this illustrates that the computational burden of Genz's integration method heavily increases with the number of constraint.

The sequential correlation-free strategy performing as well as the Genz's one and the latter having a higher computational cost, we recommend, in practice, the use of the correlation-free formulas to sequentially build the constraint design. But, note that the final constrained predictor is still computed with Genz's method.

n	N	Unconstrained		Optimal LHS		Correlation-free sequential		Genz sequential	
		Q^2	α	Q^2	α	Q^2	α	Q^2	α
50	25	0.37 ±0.08	0.68 ±0.03	0.45 ±0.08	0.73 ±0.03	0.53 ±0.08	0.79 ±0.03	0.53 ±0.08	0.79 ±0.03
	36	0.37 ±0.08	0.68 ±0.03	0.50 ±0.07	0.74 ±0.03	0.57 ±0.08	0.83 ±0.04	0.57 ±0.09	0.84 ±0.04
	64	0.37 ±0.08	0.68 ±0.03	0.53 ±0.08	0.76 ±0.03	0.64 ±0.08	0.91 ±0.04	0.64 ±0.09	0.92 ±0.04
	100	0.37 ±0.08	0.68 ±0.03	0.58 ±0.08	0.78 ±0.03	0.70 ±0.08	0.97 ±0.03	0.70 ±0.09	0.98 ±0.04
90	25	0.58 ±0.05	0.76 ±0.02	0.64 ±0.04	0.78 ±0.02	0.72 ±0.04	0.87 ±0.02	0.72 ±0.04	0.87 ±0.02
	36	0.58 ±0.05	0.76 ±0.02	0.67 ±0.04	0.79 ±0.02	0.75 ±0.04	0.90 ±0.02	0.75 ±0.04	0.90 ±0.02
	64	0.58 ±0.05	0.76 ±0.02	0.68 ±0.03	0.81 ±0.02	0.80 ±0.03	0.97 ±0.01	0.80 ±0.03	0.97 ±0.01
	100	0.58 ±0.05	0.76 ±0.02	0.72 ±0.03	0.82 ±0.02	0.83 ±0.02	0.99 ±0.01	0.83 ±0.03	0.99 ±0.01
130	25	0.68 ±0.03	0.80 ±0.02	0.72 ±0.04	0.82 ±0.02	0.80 ±0.02	0.89 ±0.02	0.80 ±0.02	0.89 ±0.02
	36	0.68 ±0.03	0.80 ±0.02	0.74 ±0.03	0.82 ±0.02	0.82 ±0.02	0.93 ±0.02	0.82 ±0.02	0.93 ±0.01
	64	0.68 ±0.03	0.80 ±0.02	0.75 ±0.03	0.84 ±0.02	0.85 ±0.02	0.99 ±0.01	0.85 ±0.02	0.99 ±0.01
	100	0.68 ±0.03	0.80 ±0.02	0.77 ±0.03	0.84 ±0.02	0.87 ±0.02	0.99 ±0.01	0.88 ±0.02	0.99 ±0.01

Table 5: Function g_2 – Mean and standard deviation of Q^2 and α with **bound constraints**, for different values of n and N, with a **Matérn 3/2 covariance**.

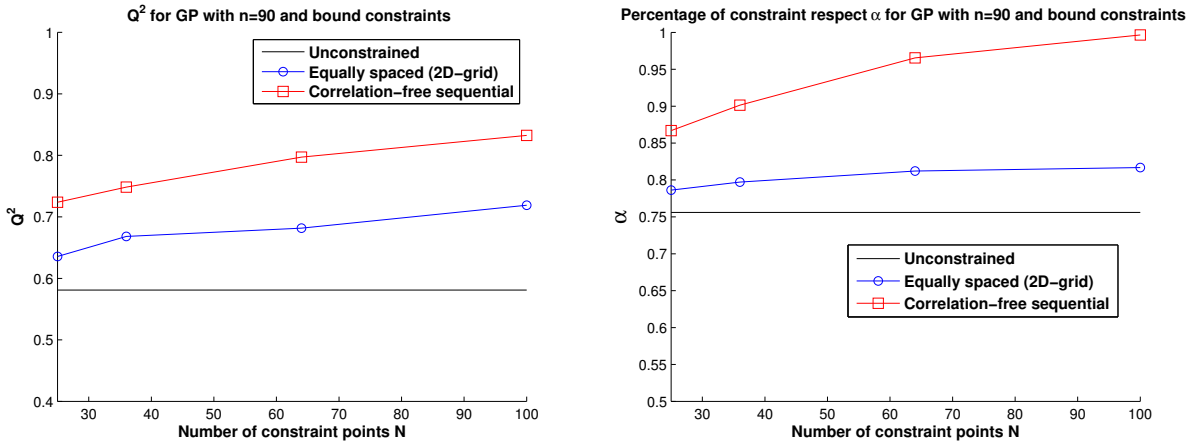

Figure 6: Function g_2 – Mean of Q^2 (left) and α (right) of the unconstrained and constrained predictors with $n = 90$ observations and **bound constraints**, with a **Matérn 3/2 covariance**.

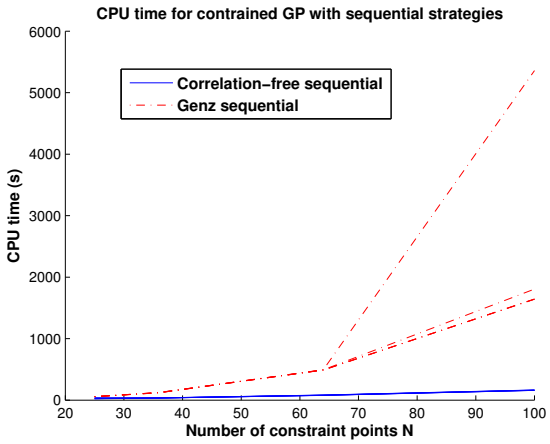

Figure 7: Function g_2 – Mean of CPU time for constrained predictors with $n = 50, 70, \ldots, 130$ observations and **bound constraints**, with a **Matérn 3/2 covariance**.

3.2.3 Illustration on 5-D example with monotonicity constraint

Finally, we consider a higher dimensional example which is more representative of industrial applications:

$$g_5(x) = 10(x_1 - 0.5)^3 + 0.3 \prod_{i=2}^{5} \left(1 + \frac{(-1)^i}{i}\right) h_i \left(x_i - \frac{2i-1}{10}\right)$$

with

$$h_n(x) = \frac{D_n(x) - 1}{\sqrt{2n}}$$

and D_n is the Dirichlet kernel :

$$D_n(x) = \frac{\sin((2n+1)\pi x)}{\sin(\pi x)}.$$

A typical constraint which can be encountered in practice is that only one partial monotonicity is known (*i.e.* constraint on the sign of a single partial derivative). This is the case we examine here, where we impose a constraint on the partial derivative w.r.t. x_1 only. From a maximin-LHS learning sample of $n = 50$ points, the unconstrained and constrained predictors are built. For the constrained ones, $N = 400$ constraint points are chosen following the non-sequential strategy (optimal LHS) and the sequential correlation-free strategy. Q^2 and α criteria are then computed on a test basis of 5000 points. Results are reported in Table 6. The unconstrained predictor has an average $Q^2 = 0.7$ and only 73% of the independent points satisfy the monotonicity. As before, the constraint incorporation via the static approach improves the results but a sequential strategy performs much better, at least in terms of constraint complying. It is important to note, however, that we need a large number of constraint points here since we are working in a 5-D space. Genz's sequential strategy is thus impractical in this case, thus highlighting the computational advantage of the correlation-free approach. Despite the crude assumption it can still reach a satisfying level of accuracy for the final predictor.

Criterion	Unconstrained	Optimal LHS	Correlation-free sequential
Q^2	0.70	0.74	0.74
Monotonicity respect α	0.73	0.86	**0.96**
Q^2 on derivative $\frac{\partial g_5}{\partial x_1}(x)$	0	0.38	**0.50**

Table 6: Function g_5 – Q^2 and monotonicity constraint respect α (percentage in $[0, 1]$) with **monotonous constraints**, for $n = 50$ and $N = 400$ with a **Matérn 3/2 covariance** function.

4 Conclusion

In this paper, we introduced a new theoretical framework for incorporating any type of linear constraints in Gaussian process modeling, including usual bound and monotonicity ones. The final constrained predictor is based on a discrete-location approximation of conditional expectations: from a computational perspective, the main result is that a vector encompassing the kriging underlying Gaussian field and any linear operator is still Gaussian. Consequently, our constrained predictors can be written as expectations of the truncated multinormal distribution and thus can be approximated by MCMC sampling, Genz's approximation or via a crude correlation-free assumption. These procedures were shown to perform very well on several 1-D or 2-D examples with various types of constraints.

However, if generalization to more dimensions seems straightforward, the discrete-location approximation for the constraints will require many points and subsequent integral approximations will suffer from the curse of dimensionality. In this case, we developed a sequential strategy inspired by previous work on GP modeling: Starting from given initial locations of constraint points, additional locations are proposed where the predictor is more likely to violate the constraints. Such relevant constraint points can be easily identified with the

Gaussian process assumption. Since Genz's approximation can become very computationally intensive in this context, we also illustrate how the naive correlation-free approach yields equivalent results with a negligible CPU time.

From our perspective, two points still have to be addressed to improve the procedure. First, maximum-likelihood estimation of kriging hyperparameters should be investigated for consistency under constraint assumptions. In practice, we hope that this will limit the number of constraint points needed for an effective discrete-location approximation. Second, error bounds on this discrete-location approximation should be examined. Apart from quality control, they could also be used in order to propose suitable locations for the constraints.

References

Abrahamsen P. and Benth F.E. (2001). Kriging with inequality constraints. *Mathematical Geology*, 33(6):719–744.

Azaïs J.-M. and Wschebor M. (2009). Level sets and extrema of random processes and fields. *New York: Wiley.*

Bigot J. and Gadat S. (2010). Smoothing under diffeomorphic constraints with homeomorphic splines. *SIAM Journal on Numerical Analysis*, 48(1):224–243.

Chopin N. (2011). Fast simulation of truncated Gaussian distributions. *Statistics and Computing*, 21:275–288.

Cozman F. and Krotkov E. (1995). Truncated Gaussians as Tolerance Sets. Fifth Workshop on Artificial Intelligence and Statistics, Fort Lauderdale Florida.

Cramér H. and Leadbetter M.R. (1967). Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications. *New York: Wiley.*

Da Veiga S., Wahl F. and Gamboa F. (2009). Local Polynomial Estimation for Sensitivity Analysis on Models With Correlated Inputs *Technometrics*, 51(4):452–463.

Da Veiga S. and Marrel A. (2012). Gaussian process modeling with inequality constraints. *Annales de la Faculté des Sciences de Toulouse*, 21(3):529–555.

Dette, H. and Scheder, R. (2006). Strictly monotone and smooth nonparametric regression for two or more variables. *The Canadian Journal of Statistics*, 34(44):535–561.

Ellis N. and Maitra R. (2007). Multivariate Gaussian Simulation Outside Arbitrary Ellipsoids. *Journal of Computational and Graphical Statistics*, 16(3):692–798.

Fang K.-T., Li R. and Sudjianto, A. (2006). Design and modeling for computer experiments. CRC Press.

Fernandez P.J., Ferrari P.A. and Grynberg S.P. (2007). Perfectly random sampling of truncated multinormal distributions. *Advances in Applied Probability*, 39(4):973–990.

Genz A. (1992). Numerical Computation of Multivariate Normal Probabilities. *Journal of Computational and Graphical Statistics*, 1:141–149.

Genz A. (1993). Comparison of Methods for the Computation of Multivariate Normal Probabilities. *Computing Science and Statistics*, 25:400–405.

Genz A. and Bretz F. (2009). Computation of Multivariate Normal and t Probabilities. *Lecture Notes in Statistics*, Vol. 195, Springer-Verlag, Heidelberg.

Geweke J. (1991). Efficient simulation from the multivariate normal and student t-distribution subject to linear constraints. *Computing Science and Statistics: Proceedings of the Twenty-Third Symposium on the Interface*, 571–578.

Ginsbourger D., Bay X. and Carraro L. (2009). Noyaux de covariance pour le Krigeage de fonctions symétriques. submitted to C. R. Acad. Sci. Paris, section Maths.

Griffiths W. (2002). A Gibbs' sampler for the parameters of a truncated multivariate normal distribution. Working Paper, http://ideas.repec.org/p/mlb/wpaper/856.html.

Hall P. and Huang L.-S. (2001). Nonparametric kernel regression subject to monotonicity constraints. *The Annals of Statistics*, 29(3):624–647.

Hazelton M.L. and Turlach B.A. (2011). Semiparametric regression with shape-constrained penalized slpines. *Computational Statistics and Data Analysis*, 55:2871–2879.

Horrace W.C. (2005). Some results on the multivariate truncated normal distribution. *Journal of Multivariate Analysis*, 94:209–221.

Johnson M. E., Moore L. M. and Ylvisaker D. (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148.

Kleijnen J.P.C. and van Beers, W.C.M (2010). Monotonicity-preserving bootstrapped Kriging metamodels for expensive simulations. Working Paper, http://www.tilburguniversity.edu/research/institutes-and-research-groups/center/staff/kleijnen/monotone_Kriging.pdf.

Kotecha J.H. and Djuric P.M. (1999). Gibbs sampling approach for generation of truncated multivariate gaussian random variables. *IEEE Computer Society*, 1757–1760.

Kotz S., Balakrishnan N. and Johnson N.L. (2000). Continuous multivariate distributions, Volume 1: models and applications New York: Wiley.

Lee L.-F. (1979). On the first and second moments of the truncated multi-normal distribution and a simple estimator. *Economics Letters*, 3:165–169.

Lee L.-F. (1983). The determination of moments of the doubly truncated multivariate tobit model. *Economics Letters*, 11:245–250.

Maatouk, H. and Bay, X. (2014). A new rejection sampling method for truncated multivariate Gaussian random variables. In *Eleventh International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*.

Marrel A., Iooss B., Van Dorpe F. and Volkova E. (2008). An efficient methodology for modeling complex computer codes with Gaussian processes. *Computational Statistics and Data Analysis*, 52:4731–4744.

Michalak A.M. (2008). A Gibbs sampler for inequality-constrained geostatistical interpolation and inverse modeling. *Water Resources Research*, 44, W09437, doi:10.1029/2007WR006645.

Muthén B. (1990). Moments of the censored and truncated bivariate normal distribution. *British Journal of Mathematical and Statistical Psychology*, 43:131–143.

Oakley JE. and O'Hagan A. (2004). Probabilistic sensitivity analysis of complex models: A Bayesian approach. *Journal of the Royal Statistical Society, Series B*, 66:751–769.

Philippe A. and Robert C. (2003). Perfect simulation of positive Gaussian distributions. *Statistics and Computing*, 13(2):179–186.

Racine J.S., Parmeter C.F. and Du P. (2009). Constrained nonparametric kernel regression: Estimation and inference. Working Paper, http:/economics.ucr.edu/spring09/Racine paper for 5 8 09.pdf.

Ramsay J.O. and Silverman B.W. (2005). Functional Data Analysis. *Springer Series in Statistics*, Springer-Verlag.

Rasmussen C.E. and Williams C.K.I (2006). Gaussian Processes for Machine Learning. The MIT Press.

Riihimäki J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In *International Conference on Artificial Intelligence and Statistics*, 645–652.

Robert C.P. (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5:121–125.

Rodriguez-Yam G., Davis R.A. and Scharf L. (2004). Efficient Gibbs Sampling of Truncated Multivariate Normal with Application to Constrained Linear Regression. Working Paper, http://www.stat.columbia.edu/ rdavis/papers/CLR.pdf.

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

Saltelli A., Chan K. and Scott E.M. (Eds.) (2000). Sensitivity Analysis. Wiley.

Santner T., Williams B. and Notz W. (2003). The design and analysis of computer experiments. Springer.

Scheuerer, M., and Schlather, M. (2012). Covariance models for divergence-free and curl-free random vector fields. *Stochastic Models*, 28(3):433–451.

Tallis G.M. (1961). The moment generating function of the truncated multinormal distribution. *Journal of the Royal Statistical Society, Series B*, 23(1):223–229.

Tallis G.M. (1963). Elliptical and radial truncation in normal populations. *Annals of Mathematical Statistics*, 34:940–944.

Tallis G.M. (1965). Plane truncation in normal populations. *Journal of the Royal Statistical Society, Series B*, 27(2):301–307.

Yoo E.-H. and Kyriakidis P.C. (2006). Area-to-point Kriging with inequality-type data. *Journal of Geographical Systems*, 8(4):357.