Sampling large hyperplane-truncated multivariate normal distributions

Hassan Maatouk, Didier Rullière, Xavier Bay

Sampling large hyperplane-truncated multivariate normal distributions

Hassan Maatouk[†1], Didier Rullière[‡2] and Xavier Bay[‡3]

(†) CY Tech, CY Cergy Paris University, Laboratoire AGM, Site du Parc, 95011 Cergy-Pontoise, France
(‡) Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F-42023 Saint-Étienne France

Abstract Generating multivariate normal distributions is widely used in various fields, including engineering, statistics, finance and machine learning. In this paper, simulating large multivariate normal distributions truncated on the intersection of a set of hyperplanes is investigated. Specifically, the proposed methodology focuses on cases where the prior multivariate normal is extracted from a stationary Gaussian process (GP). It is based on combining both Karhunen-Loève expansions (KLE) and Matheron's update rules (MUR). The KLE requires the computation of the decomposition of the covariance matrix of the random variables, which can become expensive when the random vector is too large. To address this issue, the input domain is split in smallest subdomains where the eigendecomposition can be computed. Due to the stationary property, only the eigendecomposition of the first subdomain is required. Through this strategy, the computational complexity is drastically reduced. The mean-square *truncation* and *block* errors have been calculated. The efficiency of the proposed approach has been demonstrated through both synthetic and real data studies.

Keywords Large scale random fields · hyperplane-truncated · Karhunen-Loève expansion · Matheron's update rule.

[1]hmk@cy-tech.fr
[2]drulliere@emse.fr
[3]bay@emse.fr

1 Introduction

This paper investigates the simulation of large multivariate normal (MVN) distributions truncated on the intersection of a set of hyperplanes. This problem is widely used in Bayesian regression and is closely related to nonparametric function estimation through Gaussian processes (GPs) [2, 5, 13, 20]. For instance, in [13], the authors propose a finite-dimensional GP approximation for shape-restricted function estimation. They demonstrate that sampling from the posterior distribution is equivalent to simulating a MVN distribution truncated on the nonnegative orthant. Furthermore, in econometrics, the authors in [5] apply this approach to estimate discount factors and forward rates. Recently, in [17, 24], the authors employ a constrained GP model to address the proton radius problem in nuclear physics. In [11], the authors estimate the demand for electricity as a function of temperature using a GP regression approach that utilizes the Karhunen-Loève expansion (KLE).

The standard approach involves simulating from the posterior distribution using the location-scale transformation. A *scaling* matrix of the posterior covariance matrix can be computed, and sampling from a standard MVN distribution can be performed [14]. However, computing a *scaling* matrix becomes expensive in higher dimensions, with computational complexity scaling cubically with the dimension of the MVN random variable [7]. To address this issue, it is possible to compute a polynomial or Lanczos approximation of the *scaling* matrix of the covariance matrix [1, 6]. However, simulating from the posterior distribution leads to the loss of information on the unconditional (precision or prior) covariance matrix.

The proposed methodology is quite different, as there is no need to compute the posterior covariance matrix and its decomposition. Instead, it involves combining the Karhunen-Loève expansion (KLE) and Matheron's update rule (MUR).

On the one hand, the KLE can be seen as an efficient way of simulating random fields, based on the computation of the eigendecomposition of the covariance operator [12]. However, in higher dimensions, the eigendecomposition becomes computationally expensive. To address this issue, the input domain can be divided into smaller subdomains and the KLE coefficients conditioned to satisfy the given correlation between subdomains, as proposed in [3, 16].

On the other hand, the MUR, which first appeared in geostatistics [9] and later in astrophysics [8], can be summarized in two steps. First, we simulate from an unconstrained MVN distribution. Second, we project onto the

intersection of a set of hyperplanes. In [4], the MUR algorithm is generalized to efficiently simulate random variables from a MVN distribution whose covariance (precision) matrix can be decomposed as a positive-definite matrix minus (plus) a low-rank symmetric matrix. The idea is to simulate from a block diagonal covariance matrix and use the MUR. Recently, the authors in [21] used the MUR to simulate conditional Gaussian processes. This method is called *pathwise conditioning* and has been applied to global optimization problems.

In the present paper, the main idea is to focus on the first step of the MUR algorithm. The large-scale KLE is developed for simulating the unconstrained MVN distribution. Then, we utilize the second step of the MUR to map it into the intersection of a set of hyperplanes. With this approach, we can efficiently simulate a large MVN distribution that is truncated on the intersection of a set of hyperplanes.

The article is structured as follows: In Sect. 2, we briefly recall the KLE and present the large-scale KLE technique along with some numerical illustrations. We compute the *truncation* and *global block* errors between the proposed approach and the original random field. Furthermore, we include a comparison with alternative approaches. Section 3 is devoted to sampling *hyperplane-truncated* MVN distributions. The MUR and its extension to higher dimensions are presented, including real-world applications that examine the performance of the proposed approach.

2 Karhunen-Loève expansion

Karhunen-Loève expansion (KLE) is an accurate and highly efficient approximation of random processes [19]. It can be seen as a linear combination of basis functions, with standard Gaussian random variables. By truncating the expansion at a certain number of basis functions, it achieves a balance between accuracy and computational efficiency. Recently, this approximation has been used in [11] for shape-restricted function estimation ans has also been applied to estimate electric demand as a function of temperature. Furthermore, in [10], the Karhunen-Loève procedure has been employed for the characterization of human faces.

In the following section, we provide a brief overview of the KLE approach, discussing its advantages as well as limitations.

2.1 Standard Karhunen-Loève expansion

Without loss of generality, let $(Y(x))_{x\in\mathcal{D}}$ be a zero-mean stationary Gaussian random field, whose covariance function $C(|x-x'|)$ is equal to

$$C(|x-x'|) = \mathrm{Cov}(Y(x), Y(x')) = \mathrm{E}[Y(x)Y(x')], \quad \forall x, x' \in \mathcal{D}.$$

For simplicity, we suppose that $\mathcal{D}$ is the unit interval $[0,1]$ in $\mathbb{R}$.

Table 1: Some popular stationary covariance functions with their degree of smoothness used in the Machine Learning community [20]

Name	Expression	Class
Squared Exponential	$\exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$	$\mathcal{C}^\infty$
Matérn $\nu = 5/2$	$\left(1+\frac{\sqrt{5}\lvert x-x'\rvert}{\theta}+\frac{5(x-x')^2}{3\theta^2}\right)\exp\left(-\frac{\sqrt{5}\lvert x-x'\rvert}{\theta}\right)$	$\mathcal{C}^2$
Matérn $\nu = 3/2$	$\left(1+\frac{\sqrt{3}\lvert x-x'\rvert}{\theta}\right)\exp\left(-\frac{\sqrt{3}\lvert x-x'\rvert}{\theta}\right)$	$\mathcal{C}^1$
Exponential	$\exp\left(-\frac{\lvert x-x'\rvert}{\theta}\right)$	$\mathcal{C}^0$

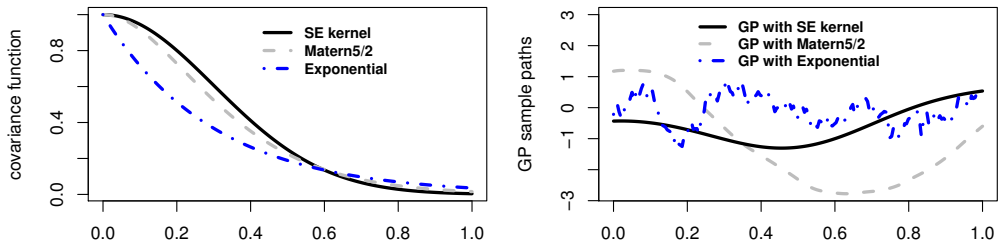

Figure 1: Some covariance functions (left) and associated GP sample paths (right). The length-scale parameter θ is fixed to 0.3

Table 1 shows some commonly used stationary covariance functions, ordered by decreasing degree of smoothness, where θ is the length-scale parameter. These kernels are widely used in the Machine Learning community [20]. Figure 1 displays sample paths of three covariance functions along with the associated GPs. The input domain $\mathcal{D} = [0,1]$ has been discretized into $N = 200$ equally spaced points, involving the simulation of a MVN distribution of size $N = 200$. In general, the computational complexity of simulating a MVN grows cubically as a function of the dimension, i.e., $\mathcal{O}(N^3)$ [7], where N is the dimension of the MVN random variable. Our aim in this section is to reduce the computational complexity when the dimension N is high (i.e.,

when $N \gg 1,000$) using the KLE method. In the following, we will provide a brief overview of the KLE approach, discussing its advantages as well as limitations. In the next section, we will introduce our approach, called *large-scale KLE*, which is specifically designed to tackle scenarios where $N \gg 1,000$.

The KLE approach is based on the eigendecomposition of the covariance function over the domain $\mathcal{D}$ [12]:

$$\int_{\mathcal{D}} C(|x - x'|)\varphi_i(x)\mathrm{d}x = \gamma_i \varphi_i(x'), \quad \forall i,\ \forall x, x' \in \mathcal{D}. \tag{1}$$

The deterministic functions $\varphi_i(x)$ and the coefficients γ_i are respectively the eigenfunctions and eigenvalues of the covariance function $C(|x - x'|)$ on the domain $\mathcal{D}$. Equation (1) is well known in mathematics as the Fredholm integral equation. The analytic solution of (1) is possible only for particular covariance functions such as Brownian motion covariance function. Numerical solutions are often used in applications. Let us recall that the eigenvalues are real and non-negative since the covariance is symmetric and positive semi-definite:

$$\iint_{\mathcal{D}^2} C(|x - x'|)f(x)f(x')\mathrm{d}x\mathrm{d}x' \geq 0$$

for any f having finite L^2 norm on $\mathcal{D}$. Let us recall also that the eigenfunctions $\varphi_i(x)$ form a complete orthonormal basis functions set [19]. This means that

$$\int_{\mathcal{D}} \varphi_i(x)\varphi_j(x)\mathrm{d}x = \delta_{ij};$$
$$\sum_{i=1}^{+\infty} \varphi_i(x)\varphi_i(x') = \delta(x - x'),$$

where δ_{ij} represents the Kronecker delta (equal one if $i = j$ and zero otherwise) and $\delta(x)$ is the Dirac distribution function.

In the KLE, we represent the Gaussian random field ($Y(x)$) using a sequence of independent standard Gaussian variables $\{\xi_i, i \in \mathbb{N}\}$ as follows:

$$Y(x) = \sum_{i=1}^{+\infty} \sqrt{\gamma_i}\xi_i\varphi_i(x), \quad \forall x \in \mathcal{D}, \tag{2}$$

where the KLE coefficients $\{\xi_i\}$ are zero-mean uncorrelated Gaussian random variables (independent) with unit variance. The KLE coefficients are defined as the projection of the Gaussian random process onto the KLE

eigenfunctions:

$$\xi_i = \frac{1}{\sqrt{\gamma_i}} \int_{\mathcal{D}} \varphi_i(x) Y(x) \mathrm{d}x, \tag{3}$$
$$\mathrm{E}[\xi_i] = 0 \quad \text{and} \quad \mathrm{E}[\xi_i \xi_j] = \delta_{ij}.$$

We have the following well-known result (see for instance [19]):

Proposition 1 (Covariance decomposition)**.** *If* $(Y(x))$ *is the random field defined as in* (2)*, then the associated covariance function can be expressed as follows:*

$$C(x, x') = \mathrm{E}[Y(x)Y(x')] = \sum_{i=1}^{+\infty} \gamma_i \varphi_i(x) \varphi_i(x'), \quad \forall x, x' \in \mathcal{D}.$$

To maintain clarity, the proof of Proposition 1 is deferred to Appendix 6.

In practice, the process $(Y(x))_{x \in \mathcal{D}}$ is approximated by the truncated sum of p terms on the domain $\mathcal{D}$ as follows:

$$Y(x) \approx \sum_{i=1}^{p} \sqrt{\gamma_i} \xi_i \varphi_i(x) = Y^p(x), \quad \forall x \in \mathcal{D}.$$

This approximation with a finite number of terms is called *truncated* KLE. Let us recall that the mean-square *truncation* error ϵ_{KL}^2 is related to the sum of the eigenvalues:

$$\epsilon_{KL}^2 = \frac{\displaystyle\int_{\mathcal{D}} \mathrm{E}\left[(Y(x) - Y^p(x))^2\right] \mathrm{d}x}{\displaystyle\int_{\mathcal{D}} \mathrm{E}[Y(x)^2] \mathrm{d}x} = 1 - \frac{\sum_{i=1}^{p} \gamma_i}{\sum_{i=1}^{+\infty} \gamma_i}. \tag{4}$$

This mean-square *truncation* error decreases with the number of terms p retained in the expansion.

When the Gaussian random field $(Y(x))$ is discretized into N equally spaced points over $\mathcal{D}$, the eigendecomposition leads to an eigenvalue problem of size $N \times N$. However, when the domain is huge and a fine discretization of the field is used, the eigendecomposition problem becomes expensive to solve, with a computational complexity of $\mathcal{O}(N^3)$. In the following sections, we propose splitting the domain $\mathcal{D}$ into smaller subdomains to reduce the complexity of the eigendecomposition problem. For instance, when the domain is split into two subdomains, the complexity reduces to $\mathcal{O}((N/2)^3)$. Instead of solving an eigendecomposition problem of order N, we only need to solve an eigendecomposition of order $N_1 = N/2$. Due to the stationary property of the random field, we show that the eigendecomposition of the first subdomain is sufficient for any number of subdomains.

2.2 Large-scale Karhunen-Loève expansion

The aim of this section is to reduce the computational complexity of simulating a MVN distribution of size N too high (i.e., $N \gg 1,000$). To the best of our knowledge, the methodology presented in this section was first introduced in [3] and then in [16]. We assume that the random field $(Y(x))_{x\in\mathcal{D}}$ is stationary, and the input domain $\mathcal{D}$ is discretized into N equally spaced points. For simplicity, the domain $\mathcal{D} = [0, MS]$ is split into M equally sized subdomains $\mathcal{D}^{(m)} = ((m-1)S, mS]$ for any block parameter $m \in \{1, \ldots, M\}$, where $MS = 1$. Thus, the first subdomain $[0, S]$ is discretized into $N_1 = N/M$ equally spaced points. The investigation of the extension to subdomains with different lengths is available in Appendix 5. The analytic expression of the covariance function $C(|x - x'|)$ of the field $(Y(x))$ is known for $x, x' \in \mathcal{D}$. For any $m \in \{1, \ldots, M\}$ and $x \in \mathcal{D}^{(m)}$, let

$$Y_m(x) = \sum_{i=1}^{+\infty} \sqrt{\lambda_i}\xi_i^{(m)}\phi_i(x - (m-1)S) \tag{5}$$

be the standard KLE covering the m^{th} subdomain $\mathcal{D}^{(m)}$, where the deterministic functions $\phi_i(x)$ and the coefficients λ_i are respectively the eigenfunctions and eigenvalues of the covariance function $C(|x - x'|)$ on the first subdomain $\mathcal{D}^{(1)} = [0, S]$, and where $\xi_i^{(m)}$ are random coefficients following a normal distribution (see Proposition 2 below for more details). Equation (5) implies that the GP Y_m is defined on the corresponding subdomain $\mathcal{D}^{(m)}$ for any $m = 1, \ldots, M$. Furthermore, when $M = 1$, Equations (2) and (5) are equivalent since $\lambda_i = \gamma_i$ and $\phi_i \equiv \varphi_i$, for any $i \in \mathbb{N}$.

Let us mention that the following notation will be used throughout this paper:

$$(Y_m \cup Y_{m'})(x) = Y_m(x)\mathbb{1}_{\mathcal{D}^{(m)}}(x) + Y_{m'}(x)\mathbb{1}_{\mathcal{D}^{(m')}}(x),$$

for any $x \in \mathcal{D}$ and $m, m' \in \{1, \ldots, M\}$, where $\mathbb{1}_{\mathcal{D}^{(m)}}(x)$ is the indicator function equal one if $x \in \mathcal{D}^{(m)}$ and zero otherwise.

Proposition 2 (Distribution on blocks)**.** *In the setting of the stationary property of the covariance function C, we have the following results:*

(i) For a given $m \in \{1, \ldots, M\}$, the process Y_m defined in (5) and Y have the same distribution on $\mathcal{D}^{(m)}$ if and only if the elements of the sequence $\{\xi_i^{(m)}\}$ are independent and identically distributed $\mathcal{N}(0, 1)$.

(ii) For any $m' > m$, $(Y_m \cup Y_{m'})$ and Y have the same distribution on $\mathcal{D}^{(m)} \cup \mathcal{D}^{(m')}$ if and only if the sequences $\{\xi_i^{(m)}\}$ and $\{\xi_j^{(m')}\}$ are normal

distributed $\mathcal{N}(0,1)$ *and the correlations between* $\xi_i^{(m)}$ *and* $\xi_j^{(m')}$ *is*

$$\mathrm{Cov}\left(\xi_i^{(m)}, \xi_j^{(m')}\right) = \frac{1}{\sqrt{\lambda_i \lambda_j}} \int_{x=0}^{S} \int_{x'=0}^{S} C\left(|x - x' - (m' - m)S|\right) \phi_i(x) \phi_j(x') \mathrm{d}x \mathrm{d}x'.$$

For clarity of the presentation, the proof of Proposition 2 is provided in Appendix 6.

In practice, we assume that p terms have been retained in the expansion

$$Y_m^p(x) = \sum_{i=1}^{p} \sqrt{\lambda_i} \xi_i^{(m)} \phi_i(x - (m-1)S), \quad \forall x \in \mathcal{D}^{(m)}.$$

Let us consider the so-called coupling matrix, which has been defined in [23] and used in [3, 16]. The coupling matrix $\boldsymbol{K}^{m,m'}$ represents the correlation between the KLE coefficients sets of any two subdomains $\mathcal{D}^{(m)}$ and $\mathcal{D}^{(m')}$, where $m < m'$. For any $i, j = 1, \ldots, p$

$$\boldsymbol{K}_{i,j}^{m,m'} = \mathrm{Cov}\left(\xi_i^{(m)}, \xi_j^{(m')}\right).$$

When $m' = m + 1$, we denote by $\boldsymbol{K}$ the coupling matrix of any two connected subdomains. It is important to recall that, under the assumption of a stationary random field and when using equally sized subdomains with uniform subdivision, the coupling matrix $\boldsymbol{K}$ is defined once between any two connected subdomains. In fact, we have for any $i, j = 1, \ldots, p$

$$\boldsymbol{K}_{i,j} = \boldsymbol{K}_{i,j}^{m,m+1} = \frac{1}{\sqrt{\lambda_i \lambda_j}} \int_{x=0}^{S} \int_{x'=0}^{S} C\left(|x - x' - S|\right) \phi_i(x) \phi_j(x') \mathrm{d}x \mathrm{d}x', \quad (6)$$

for all $m \in \{1, \ldots, M-1\}$. The coupling matrix $\boldsymbol{K}$ can be seen as the projection of $C(|x-t|)$, for $x \in [0, S]$ and $t \in [S, 2S]$, onto the basis ϕ_i.

The main idea is to generate M independent samples, each covering the corresponding subdomain, and then impose a correlation between the KLE coefficients of the two connected samples. This implies that the second item of Proposition 2 will be used when $m' = m + 1$.

Let $(Y_m^{\perp}(x))$ and $(Y_{m+1}^{\perp}(x))$ be two independent sets that cover the $\mathcal{D}^{(m)}$ and $\mathcal{D}^{(m+1)}$ subdomains, respectively, for any $m \in \{1, \ldots, M-1\}$. Thus,

$$Y_m^{\perp}(x) = \sum_{i=1}^{p} \sqrt{\lambda_i} \zeta_i^{(m)} \phi_i(x - (m-1)S), \quad \text{with } x \in ((m-1)S, mS];$$

$$Y_{m+1}^{\perp}(x) = \sum_{i=1}^{p} \sqrt{\lambda_i} \zeta_i^{(m+1)} \phi_i(x - mS), \quad \text{with } x \in (mS, (m+1)S],$$

where the KLE coefficients $\zeta_i^{(m)}$ and $\zeta_i^{(m+1)}$ are two independent replicates following a standard normal distribution $\mathcal{N}(0,1)$. Since the two sets $\boldsymbol{\zeta}^{(m)} = \left\{\zeta_i^{(m)}\right\}_i$ and $\boldsymbol{\zeta}^{(m+1)} = \left\{\zeta_i^{(m+1)}\right\}_i$ are independently generated, the two Gaussian random fields are uncorrelated:

$$\mathrm{E}\left[\zeta_i^{(m)}\zeta_j^{(m+1)}\right] = 0,\ \forall i,j \quad \Rightarrow \quad \mathrm{E}\left[Y_m^{\perp}(x)Y_{m+1}^{\perp}(t)\right] = 0,$$

for all $x \in \mathcal{D}^{(m)}$ and $t \in \mathcal{D}^{(m+1)}$.

Let us define the lower triangle matrix $\boldsymbol{L}$ of the Cholesky factorization as:

$$\mathbf{I}_p - \boldsymbol{K}^{\top}\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L}^{\top}, \tag{7}$$

where $\mathbf{I}_p$ is the $p \times p$ identity matrix and $\boldsymbol{K}^{\top}$ is the transpose of $\boldsymbol{K}$.

Proposition 3 (Construction of conditional coefficients)**.** *Under the stationary property of the random process Y, if the m^{th} conditional coefficients set $\boldsymbol{\xi}^{(m)}$ is computed as follows:*

$$\boldsymbol{\xi}^{(m)} = \boldsymbol{K}^{\top}\boldsymbol{\xi}^{(m-1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(m)}, \quad \forall m \in \{2,\ldots,M\}, \tag{8}$$

where $\boldsymbol{\xi}^{(1)} = \boldsymbol{\zeta}^{(1)}$ and $\{\boldsymbol{\zeta}^{(m)}\}$ is an independent and identically distributed sequence of standard Gaussian vectors, then

- *the two processes Y and $(Y_{m-1} \cup Y_m)$ have the same distribution on $\mathcal{D}^{(m-1)} \cup \mathcal{D}^{(m)}$, for any $m \in \{2,\ldots,M\}$.*
- *For any $m' \geq m$*

$$\begin{cases} \boldsymbol{K}^{m,m'} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}, \boldsymbol{\xi}^{(m')}\right) = \boldsymbol{K}^{m'-m}, \\ \boldsymbol{K}^{m,m} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}\right) = \mathbf{I}_p, \end{cases} \tag{9}$$

  where $\boldsymbol{K}^0$ is the identity matrix and $\boldsymbol{K}$ is the coupling matrix defined in Equation (6)*.*

For the sake of clarity, the proof of Proposition 3 is presented in Appendix 6.

Corollary 1 (Correlation between blocks)**.** *For any $(x,x') \in \mathcal{D}^{(m)} \times \mathcal{D}^{(m')}$, we have*

$$\mathrm{Cov}\left(Y_m^p(x), Y_{m'}^p(x')\right) = \sum_{i,j=1}^{p} \sqrt{\lambda_i\lambda_j}\phi_i(x-(m-1)S)\phi_j(x'-(m'-1)S)(\boldsymbol{K}^{m'-m})_{ij}, \tag{10}$$

where $\boldsymbol{K}$ is the coupling matrix given in Equation (6)*.*

Proof. The covariance between $Y_m(x)$ and $Y_{m'}(x')$ is a simple consequence of Equation (9). □

By construction, at each subdomain (except for the first two), the set of conditional coefficients is computed using the set of previous conditional coefficients on the left-hand side and the set of non-conditional coefficients on the right-hand side. From (9), when $m' = m+1$, we get the coupling matrix $\boldsymbol{K}$ between the connected subdomains. In fact, the matrix $\boldsymbol{K}^{m'-m}$ is interpreted as the coupling matrix between any two subdomains $\mathcal{D}^{(m)}$ and $\mathcal{D}^{(m')}$. To be more precise, the matrix $\boldsymbol{K}^{m'-m}$ represents the correlation between any two conditional coefficients sets $\boldsymbol{\xi}^{(m)}$ and $\boldsymbol{\xi}^{(m')}$. From Proposition 3, the elements of $\boldsymbol{\xi}^{(m)}$ are uncorrelated and then independent (since Gaussian). The elements of $\boldsymbol{\xi}^{(m)}$ are the conditional coefficients $\{\xi_i^{(m)}\}$ which are used in order to sample the random process defined on the entire domain $\mathcal{D}$:

$$\begin{cases} Y_1^p(x) = Y_1^{\perp}(x) = \sum_{i=1}^{p} \sqrt{\lambda_i}\zeta_i^{(1)}\phi_i(x), & \text{if } x \in \mathcal{D}^{(1)} \\ Y_m^p(x) = \sum_{i=1}^{p} \sqrt{\lambda_i}\xi_i^{(m)}\phi_i(x-(m-1)S), & \text{if } x \in \mathcal{D}^{(m)} \end{cases}$$

where $m \in \{2,\ldots,M\}$. Let us recall that for any $m \in \{2,\ldots,M\}$, λ_i and ϕ_i are respectively the eigenvalues and eigenfunctions of the covariance function C on the first subdomain $\mathcal{D}^{(1)} = [0,S]$.

Remark 1. *The methodology presented in this section can also be applied to the family of stationary covariance functions with compact support. As said in [20], 'compact support means that the covariance between points become exactly zero when their distance exceeds a certain threshold'. This is an interesting class of covariance function since the covariance matrix will become sparse by construction which lead to computational advantages. Let us mention that some piecewise polynomial covariance functions with compact support are given in [20] Sect. 4.2. In the present paper, the triangular covariance function will be used:*

$$C(|h|) = \max\left(1 - \frac{|h|}{\theta}, 0\right), \quad \forall h \in \mathcal{D},$$

where θ is the length-scale parameter. However, in the proposed approach, this class of covariance functions with compact support does not solve the problem of correlation errors between the coefficient sets of unconnected subdomains. This is because the proposed approach introduces non-zero correlations between the conditioning coefficient sets of unconnected subdomains $\mathbf{K}^{m,m'}$ for $m' > m+1$.

Proposition 4 (Truncation and block errors)**.** *In the setting of Proposition 3, we have the following mean-square truncation and global block errors respectively:*

$$\epsilon_T^2 = \frac{\mathrm{E}\left[\int_{\mathcal{D}^{(m)}} (Y_m(x) - Y_m^p(x))^2 \mathrm{d}x\right]}{\mathrm{E}\left[\int_{\mathcal{D}^{(m)}} Y_m(x)^2 \mathrm{d}x\right]} = 1 - \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{+\infty} \lambda_i}; \tag{11}$$

$$\epsilon_{B,M}^2 = \frac{\mathrm{E}\left[\int_{\mathcal{D}} (Y(x) - Y_{1:M}(x))^2 \mathrm{d}G(x)\right]}{\mathrm{E}\left[\int_{\mathcal{D}} Y(x)^2 \mathrm{d}G(x)\right]} = \frac{\mathrm{Trace}\left((S_Y - S_{Y_{1:M}})(S_Y - S_{Y_{1:M}})^\top\right)}{\mathrm{Trace}(S_Y S_Y^\top)}, \tag{12}$$

for any $m \in \{1, \ldots, M\}$*, where* $Y_{1:M} = \cup_{m=1}^{M} Y_m$ *and* S_Y *and* $S_{Y_{1:M}}$ *are the Cholesky matrices of the covariance functions of* Y *and* $Y_{1:M}$ *on the grid* $\mathcal{G} = \{x_1, \ldots, x_N\}$ *respectively and* G *is the cumulative distribution function (CDF) of the Uniform discrete random variable on* $\mathcal{G}$*.*

The proof of Proposition 4 is deferred to Appendix 6.

In the numerical examples presented in this section, the Matérn covariance function with a smoothness parameter of $\nu = 5/2$ (see Table 1) has been used [15]. The length-scale parameter θ is fixed at 0.2 unless otherwise specified. In this case, the mean-square *truncation* error (4) is equal to $\epsilon_{KL}^2 = 9.7 \times 10^{-6}$ when $p = 30$ terms are retained in the expansion. Other covariance functions such us the Squared Exponential (SE), exponential covariance function (see Table 1) can also be used.

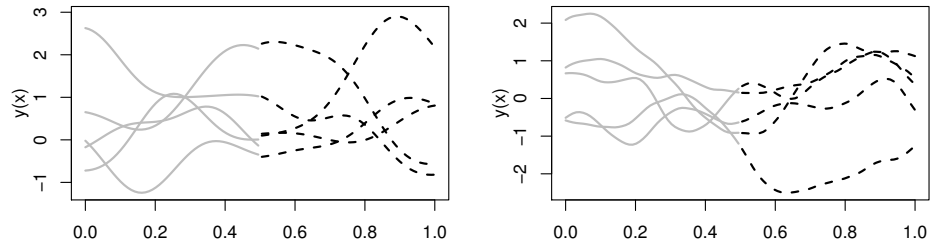

Figure 2: GP sample paths when the domain is split into two subdomains, with solid curves representing the paths before conditioning and dashed curves representing the paths after conditioning. The number of terms retained in the expansion is $p = 5$ on the left and $p = 30$ on the right

In Fig. 2, the domain $\mathcal{D} = [0, 1]$ is decomposed into two subdomains with equal size, i.e., $M = 2$ and $S = 0.5$. It is discretized into $N = 100$ uniformly

spaced points, with each subdomain consisting of $N_1 = 50$ uniformly spaced points. The black dashed curves represent the path of $Y_2^p(x)$ on $(S, 2S]$ (i.e., the random process path after conditioning), while the gray curves represent the paths of $Y_1^{\perp}(x)$ for any $x \in [0, S]$ (i.e., random process before conditioning). In the left panel, $p = 5$ terms are retained in the expansion, whereas in the right panel, $p = 30$ terms are retained. Let us mention that in the second part of the domain, the realizations are adjusted to match the generation in the first part of the domain.

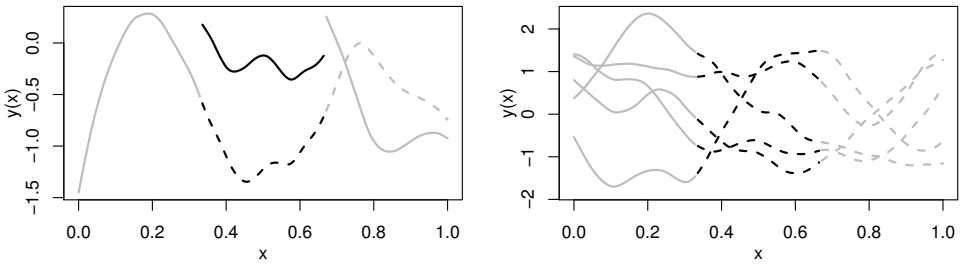

Figure 3: GP sample paths when the domain is split into three subdomains. Dashed (resp. solid) curves represent the paths after (resp. before) conditioning. The left panel shows one replicate, while the right panel shows five replicates

In Fig. 3, the domain $\mathcal{D} = [0, 1]$ is divided into three subdomains ($M = 3$) with equal lengths. It is discretized into $N = 150$ uniformly spaced points, resulting in $N_1 = 50$ uniformly spaced points per subdomain. Similar to the previous experiments, $p = 30$ terms are retained in the expansion. The solid curves (resp. dashed curves) represent the paths of the GP before (resp. after) conditioning. In the second and in the third subdomain, the realizations are adjusted to follow the generation in the left-hand side previous subdomain (see Fig. 4 for more details).

Under the same setting used in Fig. 3, the black solid curve in Fig. 4 represents the true correlation $C(|h|)$ for $h \in \mathcal{D}^{(2)}$ (left panel) and $h \in \mathcal{D}^{(3)}$ (right panel). Let us recall that the true covariance function is the Matérn covariance with a smoothness parameter $\nu = 5/2$ and a length-scale parameter of $\theta = 0.2$. The black dashed curve represents the simulated correlation using 1,000,000 replicates. In the right panel, the red dash-dotted curve represents the model correlation between $Y_1^p(0)$ and $Y_3^p(t)$ for $t \in \mathcal{D}^{(3)}$, calculated using Equation (10) from Corollary 1. As expected, the simulated correlation between the first two subdomains (left panel) coincides with the true correlation, whereas the correlation between the first and the last subdomain differs

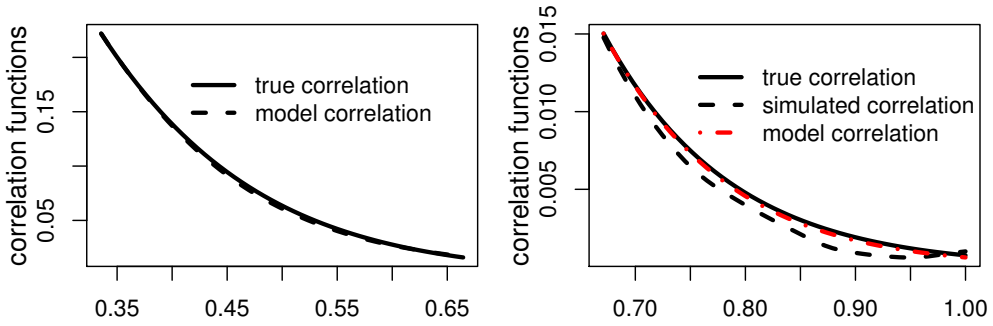

Figure 4: The covariance functions between $t = 0$ and $t \in \mathcal{D}^{(2)}$ (resp. $t \in \mathcal{D}^{(3)}$) are presented in the left (resp. right) panel. The simulated correlation is based on 1,000,000 replicates, while the model correlation is computed from Equation (10)

(right panel). This is because, by construction, the correlation between the first set of coefficients $\boldsymbol{\xi}^{(1)}$ and the third set $\boldsymbol{\xi}^{(3)}$ is estimated by the square of the coupling matrix, i.e., $\mathrm{Cov}(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(3)}) = \boldsymbol{K}^2$ as stated in the second item of Proposition 3. To improve clarity in the left panel, the model correlation between $Y_1^p(0)$ and $Y_2^p(t)$ for $t \in \mathcal{D}^{(2)}$ has been omitted.

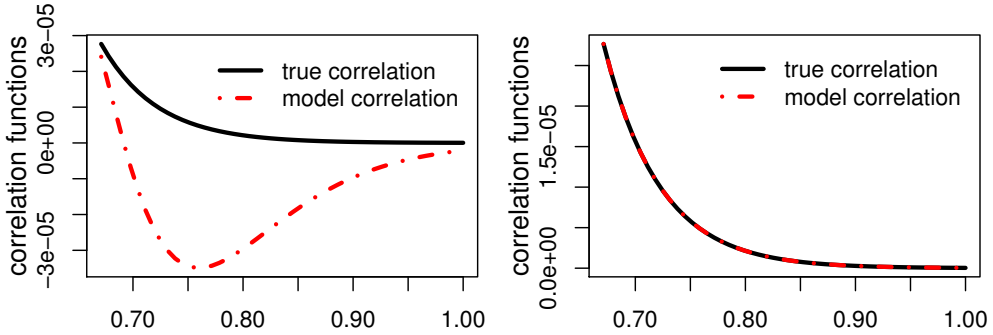

Figure 5: The covariance functions between $t = 0$ and $t \in \mathcal{D}^{(3)}$ are shown in the left (resp. right) panel, with (resp. without) truncation. The length-scale parameter is fixed at $\theta = 0.1$. The model correlation is obtained from Equation (10)

In Fig. 5, the domain $\mathcal{D}$ is divided into three subdomains and discretized into $N = 150$ equally spaced points, resulting in $N_1 = 50$ equally spaced points per subdomain. The black solid curve represents the true covariance function $C(|h|)$ for $h \in \mathcal{D}^{(3)}$, while the red dash-dotted curve represents the model correlation between $Y_1^p(0)$ and $Y_3^p(t)$ for $t \in \mathcal{D}^{(3)}$ obtained from Equation (10) in Corollary 1. The left panel shows the truncated expansion with

$p = 30$ terms, while the right panel shows the expansion without truncation, with all terms retained ($p = N_1 = 50$). When the proposed approach is used without truncation, the correlation between unconnected subdomains follows the true covariance function C, which is the Matérn covariance with a smoothness parameter of $\nu = 5/2$ and a length-scale parameter fixed at $\theta = 0.1$.

Table 2: Root-mean-square error on $\mathcal{D}$ between the true covariance function and the approximated one using the proposed approach

	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$
Triangular	3.02×10^{-16}	8.26×10^{-2}	7.56×10^{-2}
Matérn $\nu = 5/2$	5.96×10^{-13}	4.64×10^{-9}	1.65×10^{-8}
Matérn $\nu = 3/2$	3.47×10^{-16}	4.17×10^{-13}	5.02×10^{-12}
Exponential	7.86×10^{-16}	8.54×10^{-16}	1.75×10^{-15}

Table 2 shows the RMSE on $\mathcal{D}$ between the true covariance function C and the approximated covariance function obtained from the proposed approach. The domain $\mathcal{D}$ is decomposed into four subdomains and discretized into $N = 200$ equally spaced points, resulting in $N_1 = 50$ equally spaced points per subdomain. The proposed approach was used without truncation, meaning that all terms were retained ($p = N_1 = 50$). We observe that the RMSE decreases as the length-scale parameter θ and the degree of smoothness ν of the Matérn family of covariance functions decrease. Let us mention that the Markovian Exponential covariance function (refer to [20], Section 4.5 page 102) provides negligible errors compared to other covariance kernels for different values of θ and for different number of subdomains M. In this numerical example, the triangular covariance function represents the class of kernels with a compact support (see Remark 1). When $\theta = 1$, we obtain $C(|h|) = 1 - |h| > 0$ for any $h \in [0, 1[$.

Table 3: Root-mean-square *global block* error (12) on $\mathcal{D}$ between the original random process Y and the approximated one $Y_{1:M}$ with $M = 4$

	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$
Triangular	1.27×10^{-2}	2.29×10^{-1}	9.0×10^{-2}
Matérn $\nu = 5/2$	3.63×10^{-19}	5.84×10^{-12}	9.68×10^{-10}
Matérn $\nu = 3/2$	3.24×10^{-24}	2.56×10^{-19}	1.31×10^{-17}
Exponential	3.17×10^{-28}	7.22×10^{-27}	3.23×10^{-26}

Table 3 presents the mean-square *global block* error (12) on $\mathcal{D}$ between the original random process Y and the proposed process $Y_{1:M}$ with $M = 4$, under

the same setting as in Table 2. As for the RMSE of the covariance function, the *global block* error decreases as the length-scale parameter decreases and as the degree of smoothness ν of the Matérn family of covariance functions decreases. Similarly to the previous experiment, the Markovian Exponential covariance function exhibits negligible errors for various values of θ and different numbers of subdomains M. The domain $\mathcal{D}$ is decomposed into four subdomains in this case ($M = 4$). As the number of subdomains decreases, the *global block* error also decreases. For example, with only two subdomains ($M = 2$), the error $\epsilon^2_{B,M}$ is of order 5.24×10^{-23} for the Matérn covariance function with smoothness parameter $\nu = 5/2$ and length-scale parameter $\theta = 0.1$. In the same setting, when $M = 10$, we obtain $\epsilon^2_{B,M} = 2.89 \times 10^{-14}$.

2.3 Run-time comparison

This section aims to demonstrate the performance and efficiency of the proposed large-scale KLE approach through a comparison with alternative approaches, such as the naive KLE and the highly efficient Fast Fourier Transform (FFT) approach developed in [22].

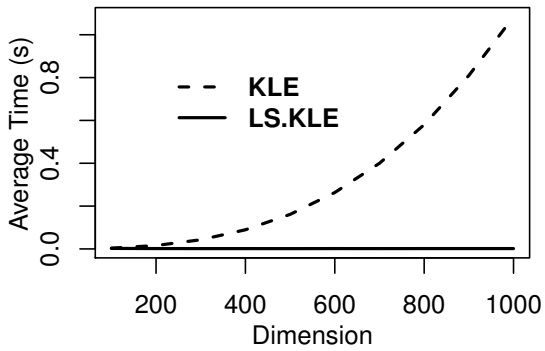

Figure 6: Average time of sampling a MVN over 25 replicates as a function of the dimension. The proposed approach, LS.KLE, has been compared to the naive KLE method

In Fig. 6, we compare the proposed large-scale KLE approach, namely, LS.KLE, with the naive KLE method. The dashed and solid curves represent the average running time (in seconds) of sampling a MVN over 25 replicates as a function of the dimension N for the naive KLE and LS.KLE, respectively. For both approaches, the number of terms retained in the expansion is $p = 30$. For the proposed approach, the first subdomain is discretized into $N_1 = 50$ equally spaced points, and the number of subdomains M takes values from

the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$. The proposed approach has a clear advantage over the naive KLE method.

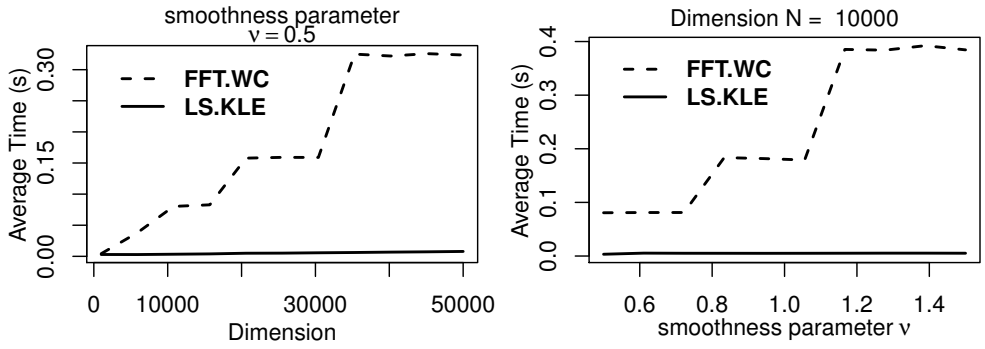

Figure 7: Average time of sampling a MVN over 25 replicates as a function of the dimension N (left) and of the smoothness parameter ν (right). The proposed approach, LS.KLE, has been compared to the Fast Fourier Transform (FFT) developed in [22]

In Fig. 7, we compare the proposed large-scale KLE approach, LS.KLE, with the FFT method developed in [22]. This is a highly efficient approach for simulating stationary GPs. The left panel shows the average running time (in seconds) of sampling a MVN distribution over 25 replicates as a function of the dimension N, with the smoothness parameter ν of the Matérn covariance function fixed at 0.5. The proposed approach demonstrates a clear advantage over the FFT method. In the right panel, we examine the computing time (in seconds) of sampling a MVN distribution over 25 replicates as a function of the smoothness parameter ν. The dimension N of the MVN variable $\boldsymbol{\eta}$ is fixed at $10,000$. Unlike the FFT method, the proposed LS.KLE approach remains stable as the smoothness parameter ν increases.

Figure 8 illustrates the performance of the proposed approach as a function of the dimension. The black curve represents the average running time (in seconds) for generating a MVN distribution over 25 replicates using the proposed LS.KLE method. The first subdomain was discretized into $N_1 = 100$ equally spaced points, and the block parameter M varied from $1,000$ to $100,000$. Notably, when employing the proposed approach, sampling a MVN of dimension $N = 10,000,000$ can be achieved in approximately 1.2 seconds, with the average time increasing linearly as a function of the dimension.

In the next section, we explain how simulating *hyperplane-truncated* MVN distributions in high dimensions $N \gg 1,000$ using the proposed large-scale KLE approach LS.KLE developed in Sect. 2.2.

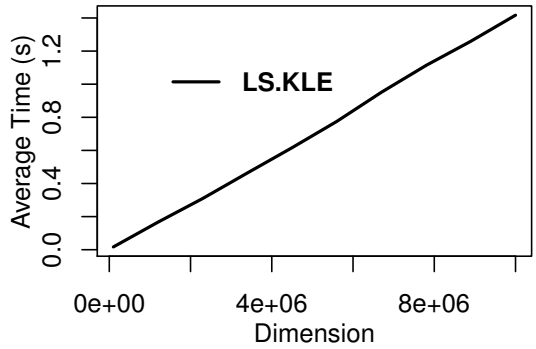

Figure 8: Average time (in seconds) of sampling a MVN over 25 replicates as a function of the dimension using the proposed approach LS.KLE

3 *Hyperplane-truncated* MVN distributions

In this section, we consider the problem of simulating a N-dimensional MVN prior $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ truncated on the intersection of a set of $n < N$ hyperplanes, denoted

$$\{\boldsymbol{\eta} \text{ s.t. } \boldsymbol{A}\boldsymbol{\eta} = \boldsymbol{y}\}, \tag{13}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times N}$, $\boldsymbol{y} \in \mathbb{R}^n$ and $\text{rank}(\boldsymbol{A}) = n$. This problem is widely used in nonparametric function estimation using GPs [13, 20, 4, 17]. For instance, the authors in [4] propose a fast algorithm for simulating a *hyperplane-truncated* MVN distribution with a covariance (precision) matrix $\boldsymbol{\Gamma}$ that can be decomposed as a positive-definite matrix minus (plus) a low-rank symmetric matrix. In the present paper, however, the only condition imposed on the precision covariance matrix $\boldsymbol{\Gamma}$ is that it is generated from a stationary GP.

In the following proposition, we recall the well-known result for computing the posterior distribution of a MVN distribution with linear equality constraints (13).

Proposition 5 (Posterior distribution)**.** *The conditional distribution of* $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ *given* $\{\boldsymbol{A}\boldsymbol{\eta} = \boldsymbol{y}\}$ *follows a MVN distribution*

$$\{\boldsymbol{\eta} \textit{ s.t. } \boldsymbol{A}\boldsymbol{\eta} = \boldsymbol{y}\} \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{C}), \quad \textit{where}$$

$$\begin{cases} \boldsymbol{\mu}_c = \boldsymbol{\mu} + (\boldsymbol{A}\boldsymbol{\Gamma})^\top (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu}); \\ \boldsymbol{C} = \boldsymbol{\Gamma} - (\boldsymbol{A}\boldsymbol{\Gamma})^\top (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}, \end{cases}$$

are the conditional mean and covariance matrix respectively.

Proof. See for instance [14, 20]. □

3.1 Sampling in low dimensions

The standard (or direct) approach for simulating from the posterior distribution $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{C})$ involves computing a *scaling* matrix of the posterior covariance matrix $\boldsymbol{C}$, denoted here as $\boldsymbol{S} \in \mathbb{R}^{N \times k}$, for any $k \in \mathbb{N}^*$ verifying $\boldsymbol{S}\boldsymbol{S}^\top = \boldsymbol{C}$, and using the following equation

$$X = \boldsymbol{\mu}_c + \boldsymbol{S}\boldsymbol{U},$$

where $\boldsymbol{U} \sim \mathcal{N}(\mathbf{0}_N, \mathbf{I}_N)$ is a N-dimensional standard Gaussian vector. To compute a *scaling* matrix $\boldsymbol{S}$, one can use Cholesky factorization [18] or eigendecomposition. Generally, the computational complexity of computing a *scaling* matrix grows cubically, and it is of order $\mathcal{O}(N^3)$ [7]. In practice, it becomes cumbersome when N is greater than one thousand.

Hereafter, we show how to simulate *hyperplane-truncated* MVN distributions without computing the covariance matrix of the posterior distribution $\boldsymbol{C}$ and its decomposition. This allows us to sample *hyperplane-truncated* MVN distributions in cases where the dimension N of the prior Gaussian vector $\boldsymbol{\eta}$ is too high (i.e., $N \gg 1,000$).

3.2 Sampling in high-dimensions

The term 'high-dimension' is used to refer to scenarios where the dimension N of the prior Gaussian vector $\boldsymbol{\eta}$ is too large (i.e., $N \gg 1,000$). We have observed that due to its complexity $\mathcal{O}(N^3)$, direct approaches become infeasible when N is significant (i.e., $N \gg 1,000$). Thus, we propose a novel solution to manage this issue.

Let us recall first the classical MUR method [9], which will be employed to address this problem.

Proposition 6 (MUR distribution)**.** *Let $\boldsymbol{\eta}$ be an N-dimensional MVN random vector with a prior distribution, characterized by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Gamma}$. Suppose that $\boldsymbol{A} \in \mathbb{R}^{n \times N}$ is a given matrix of rank n and $\boldsymbol{y} \in \mathbb{R}^n$ is an output vector. Then*

$$\{\boldsymbol{\eta} \ s.t. \ \boldsymbol{A}\boldsymbol{\eta} = \boldsymbol{y}\} \overset{d}{=} \boldsymbol{\eta} + (\boldsymbol{A}\boldsymbol{\Gamma})^\top \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top\right)^{-1} (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\eta}). \tag{14}$$

The proof of Proposition 6 is provided in Appendix 6. As mentioned in [21], a key difference with the standard approach (Proposition 5) is that we now sample before conditioning, rather than after. This observation serves as the foundation for the concept presented in this paper to update the

Algorithm 1: Sampling scheme by MUR of $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ given $\{\boldsymbol{A\eta} = \boldsymbol{y}\}$

- sample $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$;
- return $\boldsymbol{w} = \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top (\boldsymbol{A\Gamma A}^\top)^{-1} (\boldsymbol{y} - \boldsymbol{A\eta})$ which can be realized by
  - solve $\boldsymbol{\alpha}$ such that $(\boldsymbol{A\Gamma A}^\top)\boldsymbol{\alpha} = \boldsymbol{y} - \boldsymbol{A\eta}$;
  - return $\boldsymbol{w} = \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top \boldsymbol{\alpha}$.

MUR in higher dimensions. Equation (14) can be seen as a deterministic transformation of the Gaussian vector $\boldsymbol{\eta}$.

The computational complexity of Algorithm 1 is provided in Appendix 7 (Table 5) for both diagonal and non-diagonal covariance matrix $\boldsymbol{\Gamma}$.

Corollary 2. *Suppose that $\boldsymbol{w}$ is simulated with Algorithm 1. In this case, it follows a distribution $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, subject to the constraint $\{\boldsymbol{A\eta} = \boldsymbol{y}\}$, where $\boldsymbol{A} \in \mathbb{R}^{n \times N}$, $\boldsymbol{y} \in \mathbb{R}^n$, and rank$(\boldsymbol{A}) = n < N$.*

Proof. The proof is a simple consequence of Proposition 6. □

From Algorithm 1 and Proposition 6, simulating a *hyperplane-truncated* MVN distribution using MUR consists of two steps: first, we draw $\boldsymbol{\eta}$ from an unconstrained MVN as $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, and second, we map it to a vector $\boldsymbol{w}$ in the intersection of a set of hyperplanes by

$$\boldsymbol{w} = \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top \left(\boldsymbol{A\Gamma A}^\top\right)^{-1} (\boldsymbol{y} - \boldsymbol{A\eta}).$$

Using the MUR, we sample from the unconstrained MVN distribution, which offers several advantages. For instance, in the case where the unconstrained covariance (precision) matrix $\boldsymbol{\Gamma}$ is diagonal, we can exploit this property. Additionally, if $\boldsymbol{\Gamma}$ is generated from a stationary covariance function, we can preserve the stationary property in the sampling process, unlike in the direct approach, where sampling from the posterior distribution would result in a non-stationary posterior covariance. This advantage has been utilized in the large-scale KLE approach developed in Sect. 2.2, where the coupling matrix has been computed once for any arbitrary number of subdomains. Furthermore, only the eigendecomposition of the first subdomain is used. In this paper, we thus propose to combine both the large-scale KLE developed in Sect. 2.2 and the MUR in order to drastically reduce the computational complexity of sampling a hyperplane-truncated MVN distribution when N

is too large (i.e, $N \gg 1,000$). Let us mention that when the domain is split into two subdomains, the proposed approach is exact, however, when the domain is split into more than two subdomains ($M > 2$), an approximation error is generated, as discussed in Proposition 4 in Sect. 2.2.

Algorithm 2 outlines the different steps for simulating a *hyperplane-truncated* MVN distribution using the proposed approach when the domain is split into only two subdomains (i.e., $M = 2$). Recall that the input domain $\mathcal{D}$ is discretized into N equally spaced points, with $N/2$ equally spaced points in each subdomain.

Algorithm 2: Sampling scheme of $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, subject to $\{\boldsymbol{A\eta} = \boldsymbol{y}\}$ using the proposed approach when the domain is divided into two subdomains (i.e., $M = 2$)

Initialization: $\boldsymbol{A}, \boldsymbol{\Gamma}, \boldsymbol{y}, \boldsymbol{\mu}$, p, N_1 and M.

- Generating $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$:
  - sample $\boldsymbol{\zeta}^{(1)}, \boldsymbol{\zeta}^{(2)} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$;
  - compute matrices $\boldsymbol{K}$ and $\boldsymbol{L}$ using (6) and (7), respectively;
  - compute $\boldsymbol{\xi}^{(2)} = \boldsymbol{K}^\top \boldsymbol{\zeta}^{(1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(2)}$;
  - compute $\boldsymbol{\eta}_1 = \sum_{i=1}^{p} \sqrt{\lambda_i}\phi_i \zeta_i^{(1)}$ and $\boldsymbol{\eta}_2 = \sum_{i=1}^{p} \sqrt{\lambda_i}\phi_i \xi_i^{(2)}$;
  - bind $\boldsymbol{\eta} = \boldsymbol{\mu} + [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2]^\top$;
- return $\boldsymbol{w} = \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top (\boldsymbol{A\Gamma A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A\eta})$ which can be realized by
  - solve $\boldsymbol{\alpha}$ such that $(\boldsymbol{A\Gamma A}^\top)\boldsymbol{\alpha} = \boldsymbol{y} - \boldsymbol{A\eta}$;
  - return $\boldsymbol{w} = \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top \boldsymbol{\alpha}$.

The computational complexity of Algorithm 2 is given in Table 6 in Appendix 7 when the domain $\mathcal{D}$ is split into two subdomains ($M = 2$). Let us recall that in this case, Algorithm 2 is an exact method for generating *hyperplane-truncated* MVN distributions. However, for $M > 2$, the proposed approach introduces a *global block* error, which is provided in Proposition 4.

Table 4 presents a summary of the computational complexity of both the naive MUR Algorithm 1 and the proposed large-scale approach, namely LS.KLE.MUR when the domain $\mathcal{D}$ is split into M arbitrary subdomains. The parameter n represents the dimension of the set of constraints, while N represents the dimension of the prior Gaussian vector $\boldsymbol{\eta}$. As can be observed, the

Table 4: The computational complexity of Algorithms 1 and 2 when the domain $\mathcal{D}$ is split into M arbitrary subdomains. The dimension of the set of constraints is denoted by n, the dimension of the prior Gaussian vector $\boldsymbol{\eta}$ is denoted by N, and the number of subdomains is denoted by M

Operation	Computational complexity	
	naive MUR	Proposed approach
Summary	$\mathcal{O}(N^3)$	$\mathcal{O}(\max(n^3, (N/M)^3))$

proposed approach leads to a reduction in complexity, which is particularly significant when n is small.

3.3 Application to nonparametric function estimation

In this section, we explore the application of the algorithm developed in this paper to the GP regression problem [20]. The following regression problem is considered:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \tag{15}$$

$i = 1, \ldots, n$, where f represents an unknown latent function that generates the data $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$. Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a design point of dimension d, and ϵ_i is an additive independent and identically distributed (i.i.d.) zero-mean Gaussian noise with a constant variance of σ^2. A GP prior distribution on the unknown function is assumed. Let $(Y(\boldsymbol{x}))_{\boldsymbol{x}\in\mathbb{R}^d}$ be a zero-mean GP with covariance function k, i.e., $Y \sim \mathcal{GP}(0, C)$. Conditionally on the observations $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$, the conditional process remains a GP [20]

$$\{Y(\cdot) \text{ s.t. } \boldsymbol{y}\} \sim \mathcal{GP}\left(\widetilde{\mu}(\cdot), \widetilde{C}(\cdot, \cdot)\right),$$

where $\widetilde{\mu}$ and $\widetilde{C}$ are the conditional mean and covariance, respectively, given as follows:

$$\begin{aligned} \widetilde{\mu}(\boldsymbol{x}) &= \mathrm{E}\left[Y(\boldsymbol{x}) \text{ s.t.} \boldsymbol{y}\right] = C(\boldsymbol{x}, \boldsymbol{X})^\top \left(C(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \mathbf{I}_n\right)^{-1} \boldsymbol{y}; \\ \widetilde{C}(\boldsymbol{x}, \boldsymbol{x}') &= C(\boldsymbol{x}, \boldsymbol{x}') - C(\boldsymbol{x}, \boldsymbol{X})^\top (C(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \mathbf{I}_n)^{-1} C(\boldsymbol{x}', \boldsymbol{X}); \end{aligned} \tag{16}$$

for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, with $\mathbf{I}_n$ the $n \times n$ identity matrix and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$ the design matrix. Let us recall that $C(\boldsymbol{X}, \boldsymbol{X})$ is the covariance matrix of $Y(\boldsymbol{X})$ of dimension $n \times n$, and $C(\boldsymbol{x}, \boldsymbol{X}) = [C(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, C(\boldsymbol{x}, \boldsymbol{x}_n)]^\top$ is the vector of covariance between $Y(\boldsymbol{x})$ and $Y(\boldsymbol{X})$ of dimension n.

Figure 9 depicts an application of the proposed approach to nonparametric function estimation using GP regression. The finite-dimensional GP

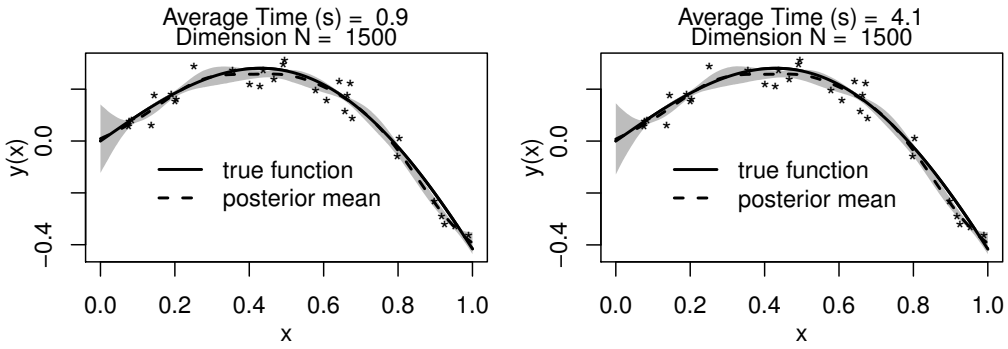

Figure 9: Nonparametric function estimation using GP regression. The black stars represent the data. The proposed approach LS.KLE.MUR is utilized in the left panel, while the naive MUR method is employed in the right panel. The average running time (in seconds) of sampling 5,000 MCMC posterior MVN over 25 replicates is displayed in the main part of each panel

approximation developed in [13] is employed. The simulation studies are based on a dataset of size $n = 50$ generated from (15) and the true function $f(x) = x\cos(2x)$, with a true standard deviation noise of $\sigma = 0.05$. The dataset is randomly split into training set of size 30 and testing set of size 20. The black stars in Figure 9 represent $n_{tr} = 30$ training data. The input domain $\mathcal{D}$ is discretized into $N = 1,500$ equally spaced points. The Matérn covariance function with a smoothness parameter $\nu = 5/2$ was employed, along with a length-scale parameter θ chosen such that the correlation at the maximum separation between covariates equals 0.2. The black solid curve represents the true unknown function, while the black dashed curve represents the posterior mean (16). The gray shaded area represents the posterior 95% credible interval of f based on 5,000 MCMC simulations. In the left panel, the proposed large-scale KLE and MUR approach, namely LS.KLE.MUR is utilized, where the domain $\mathcal{D}$ is split into just $M = 5$ subdomains. In the right panel, the naive MUR method is employed, where the sampling of the prior $\boldsymbol{\eta}$ (first step in Algorithm 1) is performed using the standard KLE approach. The average running time (in seconds) of sampling 5,000 MCMC posterior MVN over 25 replicates is displayed in the main part of each panel.

3.4 Real-world data applications

In this section, the proposed large-scale approach (LS.KLE.MUR) developed in this paper has been evaluated through two real-world data studies. The age-income dataset consisting of age (in years) and the logarithm of income

(log.income) for 205 Canadian workers from a 1971 Canadian Census Public Use Tape. The aim is to estimate the logarithm of income as function of age. This real-life data will be used to showcase the superior computational running time of the proposed approach.

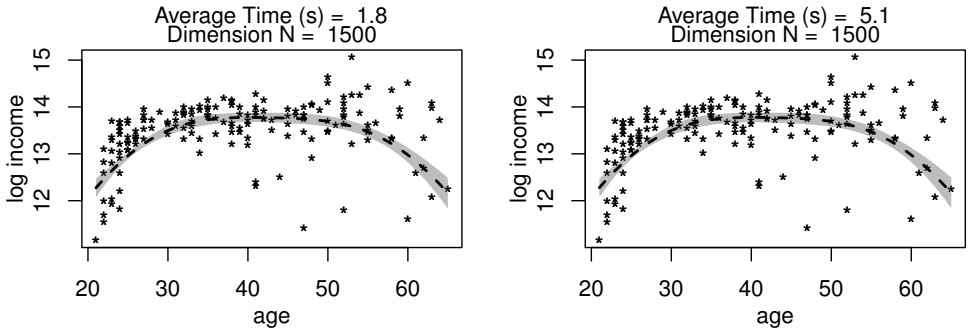

Figure 10: Nonparametric function estimation is performed through GP regression applied to the age-income dataset. The black stars represent the training noisy data. The black dashed curve represents the posterior mean, while the gray shaded area represents the 95% posterior coverage interval. The proposed approach is utilized in the left panel, while the naive MUR is employed in the right panel. The average running time (in seconds) over 25 replicates of sampling 5,000 MCMC posterior sample paths is displayed in the main part of each panel

Figure 10 presents a computational comparison between the proposed large-scale approach (LS.KLE.MUR) and the naive MUR for age-income real-world data application. The input domain is discretized into $N = 1,500$ equally spaced points. The dataset is randomly split into an 80% training set and a 20% testing set. The black stars in Fig. 10 represent the 164 training data points. The black dashed curve represents the posterior mean (16), while the gray shaded area represents the posterior 95% credible interval based on 5,000 MCMC simulations. In the left panel, the proposed approach (LS.KLE.MUR) is utilized, where the input domain is divided into $M = 10$ subdomains. Consequently, each subdomain is discretized into $N_1 = 150$ equally spaced points. In the right panel, the naive MUR method is employed, where the sampling of the prior $\boldsymbol{\eta}$ (first step in Algorithm 1) is performed using the standard KLE approach. In these cases, the average running time (in seconds) over 25 replicates of sampling 5,000 MCMC posterior sample paths is 1.8 seconds for the proposed approach, while it takes 5.1 seconds when using the naive MUR.

In Fig. 11, nonparametric function estimation is performed through GP regression applied to fossil data, which consists of 164 observations on fossil

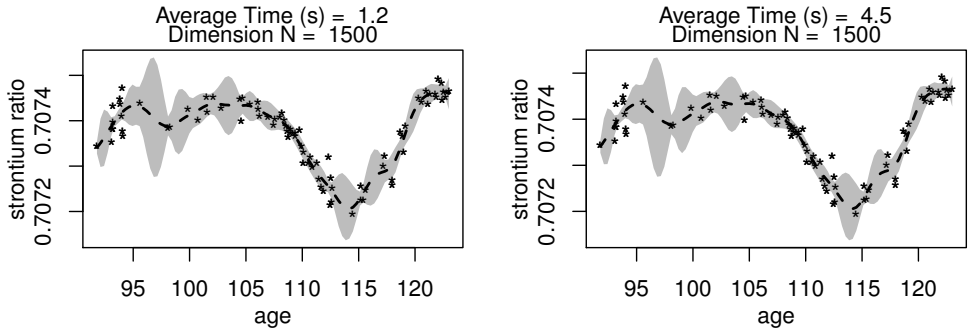

Figure 11: Nonparametric function estimation is performed through GP regression applied to fossil data. The panel description is the same as in Fig. 10

shells. The objective is to estimate the ratios of strontium isotopes as a function of age in million of years. Similar to the age-income application, the dataset is randomly divided into an 80% training dataset (represented by black stars) and a 20% testing dataset. The input domain is discretized into $N = 1,500$ equally spaced points. The proposed LS.KLE.MUR approach has been employed in the left panel, while the naive MUR is utilized in the right panel. The average running time (in seconds) for 25 replicates of sampling 5,000 MCMC posterior sample paths is of order 1.2 for the proposed approach, whereas it takes 4.5 seconds for the naive MUR.

4 Conclusion

This paper presents a highly efficient approach to simulate a large multivariate normal distribution truncated on the intersection of a set of hyperplanes. This problem is closely related to nonparametric function estimation through Gaussian process regression. The main idea is to combine both Karhunen-Loève expansion and Matheron's update rule. The proposed method is specific for cases where the prior multivariate normal variable is extracted from a stationary Gaussian process. The simulation process involves two steps. Firstly, we efficiently simulate from the unconstrained MVN distribution using a large-scale Karhunen-Loève expansion. We demonstrate that, due to the stationary property, only the eigendecomposition of the first subdomain is necessary. Secondly, we map the samples onto the intersection of a set of hyperplanes using Matheron's update rule. The proposed approach significantly reduces computational complexity. Mean-square *truncation* and *global block* errors have been calculated, and a comparison with competitor approaches is included. The performance of the proposed approach has been evaluated using both synthetic and real data studies.

Appendix

5 Subdomains with different lengths

The methodology presented in Sect. 2.2 can also be extended to cases where the domain is split into subdomains with different lengths. In such cases, the eigendecomposition and the coupling matrix must be computed at each subdomain. For example, suppose that the domain $\mathcal{D} = [0, 1]$ is divided into two subdomains $[0, S] \cup [S, 1]$ as illustrated in Fig. 12 below, where $S = 0.7$. The only modification in the methodology presented before is in computing both the coupling matrix $\boldsymbol{K}$ and the lower triangle $\boldsymbol{L}$. Let us denote these two matrices respectively by $\boldsymbol{K}^{1,2}$ and $\boldsymbol{L}^{1,2}$ since they depend on the two connected subdomains. The elements of $\boldsymbol{K}^{1,2} \in \mathbb{R}^{p\times p}$ are defined as

$$\boldsymbol{K}^{1,2}_{i,j} = \frac{1}{\sqrt{\lambda^{(1)}_i \lambda^{(2)}_j}} \int_{x=0}^{S} \int_{t=S}^{1} C(|x-t|)\phi^{(1)}_i(x)\phi^{(2)}_j(t)\mathrm{d}x\mathrm{d}t,$$

where $\{\lambda^{(1)}_i, \phi^{(1)}_i\}_{i=1}^p$ and $\{\lambda^{(2)}_j, \phi^{(2)}_j\}_{j=1}^p$ are the eigendecomposition of C on $[0, S]$ and $(S, 1]$, respectively. The lower triangle matrix $\boldsymbol{L}^{1,2}$ is computed as follows

$$\mathbf{I}_p - \left(\boldsymbol{K}^{1,2}\right)^\top \boldsymbol{K}^{1,2} = \boldsymbol{L}^{1,2}\left(\boldsymbol{L}^{1,2}\right)^\top.$$

As before, the second KLE conditional coefficients set is generated as follows:

$$\boldsymbol{\xi}^{(2)} = \left(\boldsymbol{K}^{1,2}\right)^\top \boldsymbol{\xi}^{(1)} + \boldsymbol{L}^{1,2}\boldsymbol{\zeta}^{(2)},$$

where $\boldsymbol{\xi}^{(1)} = \boldsymbol{\zeta}^{(1)} \in \mathbb{R}^p$ and $\boldsymbol{\zeta}^{(2)} \in \mathbb{R}^p$ are two replicates following a standard MVN distribution. Finally, the random process defined on the entire domain $\mathcal{D} = [0, 1]$ is given by

$$\begin{cases} Y^{\perp}_1(x) = \sum_{i=1}^p \sqrt{\lambda^{(1)}_i}\zeta^{(1)}_i\phi^{(1)}_i(x), & \text{if } x \in [0, S] \\ Y^{p}_2(x) = \sum_{j=1}^p \sqrt{\lambda^{(2)}_j}\xi^{(2)}_j\phi^{(2)}_j(x), & \text{if } x \in (S, 1]. \end{cases}$$

The correlation between any two coefficients sets $\boldsymbol{\xi}^{(m)}$ and $\boldsymbol{\xi}^{(m')}$ for any $m' > m$ is equal to

$$\begin{aligned} \boldsymbol{K}^{m,m'} &= \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}, \boldsymbol{\xi}^{(m')}\right) = \boldsymbol{K}^{m,m+1} \times \boldsymbol{K}^{m+1,m+2} \times \ldots \times \boldsymbol{K}^{m'-1,m'} \\ &= \prod_{\ell=m}^{m'-1} \boldsymbol{K}^{\ell,\ell+1}, \end{aligned}$$

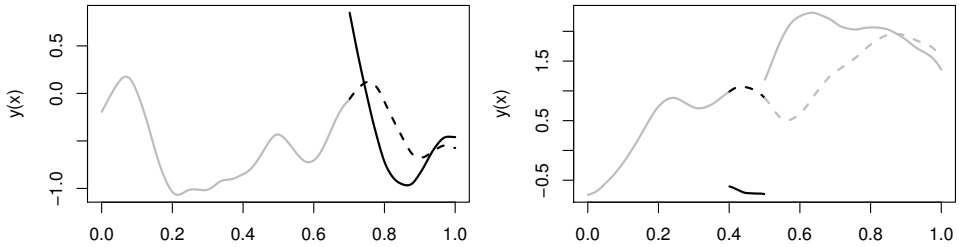

Figure 12: GP sample paths when the domain is split in two (left panel) and three (right panel) subdomains with different lengths

where $\boldsymbol{K}^{\ell,\ell+1} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(\ell)}, \boldsymbol{\xi}^{(\ell+1)}\right)$, for any $\ell \in \{m, \ldots, m'-1\}$.

In Fig. 12, the domain $\mathcal{D}$ is split in two (resp. three) subdomains in the left (resp. right) panel with different lengths. The solid curves (resp. dashed curves) represent the paths before (resp. after) conditioning. The Matérn covariance function with a smoothness parameter $\nu = 5/2$ has been utilized, where the length-scale parameter θ is fixed at 0.2.

6 Proofs of Propositions

This appendix contains the proofs of the propositions presented in the main body of the manuscript.

Proof of Proposition 1. We have for any $x, x' \in \mathcal{D}$

$$\begin{aligned} C(x,x') &= \mathrm{E}[Y(x)Y(x')] = \sum_{i=1}^{+\infty}\sum_{j=1}^{+\infty}\sqrt{\gamma_i\gamma_j}\varphi_i(x)\varphi_j(x')\mathrm{E}[\xi_i\xi_j] \\ &= \sum_{i=1}^{+\infty}\sum_{j=1}^{+\infty}\sqrt{\gamma_i\gamma_j}\varphi_i(x)\varphi_j(x')\delta_{ij} = \sum_{i=1}^{+\infty}\gamma_i\varphi_i(x)\varphi_i(x'), \end{aligned}$$

which concludes the proof of the proposition. □

Proof of Proposition 2.

(i) From right to left: suppose that $\{\xi_i^{(m)}\}$ are independent and identically distributed $\mathcal{N}(0,1)$. The two processes Y and Y_m are zero-mean Gaussian. Firstly, from Proposition 1 and the stationary property of the

covariance function, we have for any $t, t' \in \mathcal{D}^{(m)}$

$$\begin{aligned} C(t,t') &= \mathrm{Cov}(Y(t), Y(t')) = \mathrm{Cov}\left(Y(t-(m-1)S), Y(t'-(m-1)S)\right) \\ &= \mathrm{Cov}(Y(x), Y(x')) = \sum_{i=1}^{+\infty} \lambda_i \phi_i(x) \phi_i(x'), \end{aligned}$$

where $x = t-(m-1)S$ and $x' = t'-(m-1)S$ are in $\mathcal{D}^{(1)}$. Secondly, we have for any $t, t' \in \mathcal{D}^{(m)}$

$$\begin{aligned} \mathrm{Cov}(Y_m(t), Y_m(t')) &= \sum_{i,j=1}^{+\infty} \sqrt{\lambda_i \lambda_j} \phi_i(t-(m-1)S) \phi_j(t'-(m-1)S) \delta_{ij} \\ &= \sum_{i=1}^{+\infty} \lambda_i \phi_i(t-(m-1)S) \phi_i(t'-(m-1)S) \\ &= \sum_{i=1}^{+\infty} \lambda_i \phi_i(x) \phi_i(x'), \end{aligned}$$

where $x = t-(m-1)S$ and $x' = t'-(m-1)S$ are in $\mathcal{D}^{(1)}$. Now, from left to right: suppose that Y_m and Y have the same distribution on $\mathcal{D}^{(m)}$. This means that for any $x, x' \in \mathcal{D}^{(m)}$

$$\mathrm{Cov}\left(Y_m(x), Y_m(x')\right) = \mathrm{Cov}(Y(x), Y(x')) = C(|x-x'|).$$

Therefore, for any i, j

$$\begin{aligned} \mathrm{Cov}(\xi_i^{(m)}, \xi_j^{(m)}) &= \frac{1}{\sqrt{\lambda_i \lambda_j}} \iint_{\mathcal{D}^{(1)}} \phi_i(t) \phi_j(t') C(|t-t'|) \mathrm{d}t \mathrm{d}t' \\ &= \frac{\lambda_j}{\sqrt{\lambda_i \lambda_j}} \int_{\mathcal{D}^{(1)}} \phi_i(t) \phi_j(t) \mathrm{d}t = \delta_{ij}, \end{aligned}$$

which concludes the proof of the first item of the proposition.

(ii) Let us consider the simple case when $m' = m+1$. For example, let $m = 1$ then $m' = 2$. From right to left: on the one hand, we have for any $(s,t) \in \mathcal{D}^{(1)} \times \mathcal{D}^{(2)}$,

$$\mathrm{Cov}(Y(s), Y(t)) = \mathrm{E}[Y(s)Y(t)] = C(|s-t|).$$

On the other hand, we have

$$\begin{aligned}
\mathrm{E}[Y_1(s)Y_2(t)] &= \sum_{i,j=1}^{+\infty} \sqrt{\lambda_i \lambda_j}\phi_i(s)\phi_j(t-S)\mathrm{E}[\xi_i^{(1)}\xi_j^{(2)}] \\
&= \sum_{i,j=1}^{+\infty} \phi_i(s)\phi_j(t-S)\int_{x=0}^{S}\int_{x'=S}^{2S}\phi_i(x)\phi_j(x'-S)C(|x-x'|)\mathrm{d}x\mathrm{d}x' \\
&= \int_0^S\int_S^{2S}\sum_{i=1}^{+\infty}\phi_i(s)\phi_i(x)\sum_{j=1}^{+\infty}\phi_j(t-S)\phi_j(x'-S)C(|x-x'|)\mathrm{d}x\mathrm{d}x' \\
&= \int_0^S\int_S^{2S}\delta(s-x)\delta(t-x')C(|x-x'|)\mathrm{d}x\mathrm{d}x' = C(|s-t|).
\end{aligned}$$

From left to right: we have

$$\begin{aligned}
\mathrm{Cov}\left(\xi_i^{(1)},\xi_j^{(2)}\right) &= \mathrm{Cov}\left(\frac{1}{\sqrt{\lambda_i}}\int_0^S\phi_i(x)Y_1(x)\mathrm{d}x, \frac{1}{\sqrt{\lambda_j}}\int_{t=S}^{2S}\phi_j(t-S)Y_2(t)\mathrm{d}t\right) \\
&= \frac{1}{\sqrt{\lambda_i\lambda_j}}\int_{x=0}^{S}\int_{t=S}^{2S}C(|x-t|)\phi_i(x)\phi_j(t-S)\mathrm{d}x\mathrm{d}t \\
&= \frac{1}{\sqrt{\lambda_i\lambda_j}}\int_{x=0}^{S}\int_{x'=0}^{S}C\left(|x-x'-S|\right)\phi_i(x)\phi_j\left(x'\right)\mathrm{d}x\mathrm{d}x'.
\end{aligned}$$

The general case can be proved in a similar way.

□

Proof of Proposition 3.

- From Equation (8), the correlation between $\boldsymbol{\xi}^{(m-1)}$ and $\boldsymbol{\xi}^{(m)}$ is imposed in order to satisfy the correlation structure between the connected subdomains $\mathcal{D}^{(m-1)}$ and $\mathcal{D}^{(m)}$

$$\boldsymbol{K}^{m-1,m} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m-1)},\boldsymbol{\xi}^{(m)}\right) = \boldsymbol{K}, \quad \forall m \in \{2,\ldots,M\}. \tag{17}$$

For example, when $m = 3$, we get

$$\begin{aligned}
\boldsymbol{K}^{2,3} &= \mathrm{Cov}\left(\boldsymbol{\xi}^{(2)},\boldsymbol{\xi}^{(3)}\right) = \mathrm{Cov}\left(\boldsymbol{K}^\top\boldsymbol{\zeta}^{(1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(2)}, \boldsymbol{K}^\top\boldsymbol{\xi}^{(2)} + \boldsymbol{L}\boldsymbol{\zeta}^{(3)}\right) \\
&= \boldsymbol{K}^\top\mathrm{Cov}\left(\boldsymbol{\zeta}^{(1)},\boldsymbol{\xi}^{(2)}\right)\boldsymbol{K} + \boldsymbol{L}\mathrm{Cov}\left(\boldsymbol{\zeta}^{(2)},\boldsymbol{\xi}^{(2)}\right)\boldsymbol{K} \\
&= \boldsymbol{K}^\top\boldsymbol{K}\boldsymbol{K} + \boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{K} = (\boldsymbol{K}^\top\boldsymbol{K} + \boldsymbol{L}\boldsymbol{L}^\top)\boldsymbol{K} = \boldsymbol{K}.
\end{aligned}$$

The general case can be proved in a similar way. The second item of Proposition 2 concludes the proof.

- The second item is a simple consequence of Equations (8) and (17). For example, when $m = 1$ and $m' = 3$, we get

$$\begin{aligned} \boldsymbol{K}^{1,3} &= \mathrm{Cov}\left(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(3)}\right) = \mathrm{Cov}\left(\boldsymbol{\xi}^{(1)}, \boldsymbol{K}^\top \boldsymbol{\xi}^{(2)} + \boldsymbol{L}\boldsymbol{\zeta}^{(3)}\right) \\ &= \mathrm{Cov}\left(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}\right) \boldsymbol{K} = \boldsymbol{K}^2. \end{aligned}$$

  In particular, for any $m \in \{2, \ldots, M\}$

$$\begin{aligned} \boldsymbol{K}^{m,m} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}\right) &= \mathrm{Cov}\left(\boldsymbol{K}^\top \boldsymbol{\xi}^{(m-1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(m)}\right) \\ &= \boldsymbol{K}^\top \mathrm{Cov}(\boldsymbol{\xi}^{(m-1)})\boldsymbol{K} + \boldsymbol{L}\mathrm{Cov}(\boldsymbol{\zeta}^{(m)})\boldsymbol{L}^\top \\ &= \boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{L}\boldsymbol{L}^\top = \mathbf{I}_p, \end{aligned}$$

  and it is evident for $m = 1$. This concludes the proof of the second item and thus of the proposition.

□

Proof of Proposition 4. For the mean-square *truncation* error (11), on the one hand,

$$\begin{aligned} \mathrm{E}\left[\int_{\mathcal{D}^{(m)}} Y_m(x)^2 \mathrm{d}x\right] &= \mathrm{E}\left[\int_{\mathcal{D}^{(m)}} \sum_{i,j=1}^{+\infty} \sqrt{\lambda_i \lambda_j}\phi_i(x-(m-1)S)\phi_j(x-(m-1)S)\xi_i^{(m)}\xi_j^{(m)}\right] \\ &= \sum_{i,j=1}^{+\infty} \sqrt{\lambda_i \lambda_j}\mathrm{E}[\xi_i^{(m)}\xi_j^{(m)}] \int_{\mathcal{D}^{(m)}} \phi_i(x-(m-1)S)\phi_j(x-(m-1)S)\mathrm{d}x \\ &= \sum_{i,j=1}^{+\infty} \sqrt{\lambda_i \lambda_j}\mathrm{E}[\xi_i^{(m)}\xi_j^{(m)}]\delta_{i,j} = \sum_{i=1}^{+\infty} \lambda_i. \end{aligned}$$

On the other hand,

$$\begin{aligned} \mathrm{E}\left[\int_{\mathcal{D}^{(m)}} (Y_m(x) - Y_m^p(x))^2 \mathrm{d}x\right] &= \mathrm{E}\left[\int_{\mathcal{D}^{(m)}} \left(\sum_{i=p+1}^{+\infty} \sqrt{\lambda_i}\phi_i(x-(m-1)S)\xi_i^{(m)}\right)^2\right] \\ &= \sum_{i=p+1}^{+\infty} \sqrt{\lambda_i \lambda_j}\mathrm{E}[\xi_i^{(m)}\xi_j^{(m)}]\delta_{i,j} = \sum_{i=p+1}^{+\infty} \lambda_i. \end{aligned}$$

Thus,

$$\epsilon_T^2 = \frac{\sum_{i=p+1}^{+\infty} \lambda_i}{\sum_{i=1}^{+\infty} \lambda_i} = \frac{\sum_{i=1}^{+\infty} \lambda_i - \sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{+\infty} \lambda_i} = 1 - \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{+\infty} \lambda_i}.$$

For the mean-square *global block* error, the result is a simple consequence of the fact that any Gaussian vector can be written as follows [14]

$$\begin{pmatrix} Y(x_1) \\ \vdots \\ Y(x_N) \end{pmatrix} = S_Y \times \epsilon \quad \text{and} \quad \begin{pmatrix} Y_{1:M}(x_1) \\ \vdots \\ Y_{1:M}(x_N) \end{pmatrix} = S_{Y_{1:M}} \times \epsilon,$$

where ϵ is a N-dimensional standard Gaussian vector chosen the same for Y and $Y_{1:M}$ to get a specific dependence structure. The two matrices S_Y and $S_{Y_{1:M}}$ are the Cholesky factorization of the covariance of Y and $Y_{1:M}$ on the grid $\mathcal{G} = \{x_1, \ldots, x_N\}$ respectively. □

The following Lemma is presented before proving Proposition 6.

Lemma 1. *Let W_1, W_2 and W_3 be three Gaussian random vectors such that W_2 is independent of W_3 verifying:*

$$W_1 \stackrel{d}{=} f(W_2) + W_3,$$

where f is a measurable function of W_2. Then

$$\{W_1 \ s.t. \ W_2 = \beta\} \stackrel{d}{=} f(\beta) + W_3.$$

Proof of Lemma 1. The proof is given in [21]. □

Proof of Proposition 6. We have $\mathrm{E}[\boldsymbol{\eta} \text{ s.t. } \boldsymbol{A\eta}] = \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}\boldsymbol{A\eta}$. Let $W_3 = \boldsymbol{\eta} - \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}\boldsymbol{A\eta}$. The two vectors $\boldsymbol{A\eta}$ and W_3 are uncorrelated:

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{A\eta}, W_3) &= \mathrm{Cov}(\boldsymbol{A\eta}, W_3) = \mathrm{Cov}(\boldsymbol{A\eta}, \boldsymbol{\eta} - \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}\boldsymbol{A\eta}) \\ &= \mathrm{Cov}(\boldsymbol{A\eta}, \boldsymbol{\eta}) - \mathrm{Cov}(\boldsymbol{A\eta}, \boldsymbol{\Gamma A}^\top(\boldsymbol{A\Gamma A}^\top)^{-1}\boldsymbol{A\eta}) \\ &= \boldsymbol{A\Gamma} - \boldsymbol{A\Gamma A}^\top(\boldsymbol{A\Gamma A}^\top)^{-1}\boldsymbol{A\Gamma} = \boldsymbol{A\Gamma} - \boldsymbol{A\Gamma} = \mathbf{0}_{n,N}. \end{aligned}$$

Thus, $\boldsymbol{A\eta}$ and W_3 are independent (since Gaussian). Applying Lemma 1, we get

$$\begin{aligned} \{\boldsymbol{\eta} \text{ s.t. } \boldsymbol{A\eta} = \boldsymbol{y}\} &\stackrel{d}{=} f(\boldsymbol{y}) + W_3 = \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}\boldsymbol{y} + \boldsymbol{\eta} - \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}\boldsymbol{A\eta} \\ &\stackrel{d}{=} \boldsymbol{\eta} + \Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}}\Sigma^{-1}_{\boldsymbol{A\eta},\boldsymbol{A\eta}}(\boldsymbol{y} - \boldsymbol{A\eta}) \stackrel{d}{=} \boldsymbol{\eta} + \boldsymbol{\Gamma A}^\top(\boldsymbol{A\Gamma A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A\eta}) \\ &\stackrel{d}{=} \boldsymbol{\eta} + (\boldsymbol{A\Gamma})^\top(\boldsymbol{A\Gamma A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A\eta}) \end{aligned}$$

where $\Sigma_{\boldsymbol{\eta},\boldsymbol{A\eta}} = \mathrm{Cov}(\boldsymbol{\eta}, \boldsymbol{A\eta}) = \mathrm{Cov}(\boldsymbol{\eta}, \boldsymbol{\eta})\boldsymbol{A}^\top = \Gamma\boldsymbol{A}^\top = (\boldsymbol{A\Gamma})^\top$ and $\Sigma_{\boldsymbol{A\eta},\boldsymbol{A\eta}} = \mathrm{Cov}(\boldsymbol{A\eta}, \boldsymbol{A\eta}) = \boldsymbol{A}\mathrm{Cov}(\boldsymbol{\eta}, \boldsymbol{\eta})\boldsymbol{A}^\top = \boldsymbol{A\Gamma A}^\top$. □

Table 5: The computational complexity of Algorithm 1. The parameters n and N are respectively the dimension of the set of constraints and of the prior MVN $\boldsymbol{\eta}$

Operation	Computational complexity	
	Non-diagonal $\boldsymbol{\Gamma}$	Diagonal $\boldsymbol{\Gamma}$
$\boldsymbol{\eta}$	$\mathcal{O}(N^3)$	$\boldsymbol{\mathcal{O}(N)}$
$\boldsymbol{A\Gamma A}^\top$	$\mathcal{O}(nN^2)$	$\boldsymbol{\mathcal{O}(n^2N)}$
$\boldsymbol{\alpha}$	$\mathcal{O}(\max(nN, n^3))$	$\mathcal{O}(\max(nN, n^3))$
$\boldsymbol{w}$	$\mathcal{O}(nN)$	$\mathcal{O}(nN)$
Summary	$\mathcal{O}(N^3)$	$\boldsymbol{\mathcal{O}(n^2N)}$

Table 6: The computational complexity of Algorithm 2 when the domain $\mathcal{D}$ is split into two subdomains $M = 2$. The parameters n and N represent the dimension of the set of constraints and the prior MVN $\boldsymbol{\eta}$, respectively, while p denotes the number of terms retained in the expansion

Operation	Computational complexity
$\boldsymbol{\zeta}^{(1)}, \boldsymbol{\zeta}^{(2)}$	$\mathcal{O}(p)$
$\{\lambda_i, \phi_i\}$	$\mathcal{O}((N/2)^3)$
$\boldsymbol{K}$ and $\boldsymbol{L}$	$\mathcal{O}(p(N/2)^2)$ and $\mathcal{O}(p^3)$
$\boldsymbol{\xi}^{(2)}$	$\mathcal{O}(p^2)$
w_1 and w_2	$\mathcal{O}(pN/2)$
$\boldsymbol{A\Gamma A}^\top$	$\mathcal{O}(nN^2)$
$\boldsymbol{\alpha}$	$\mathcal{O}(\max(nN, n^3))$
$\boldsymbol{w}$	$\mathcal{O}(nN)$
Summary	$\mathcal{O}(\max(n^3, (N/2)^3))$

7 Computational complexity of Algorithms 1 and 2

Acknowledgements

This research was conducted with the support of the consortium in Applied Mathematics CIROQUO, gathering partners in technological and academia in the development of advanced methods for Computer Experiments. `https://doi.org/10.5281/zenodo.6581217`

References

[1] E. Aune, J. Eidsvik, and Y. Pokern. Iterative numerical methods for sampling from high dimensional Gaussian distributions. *Stat. Comput.*, 23(4):501–521, 2013.

[2] C. Chevalier, X. Emery, and D. Ginsbourger. Fast update of conditional simulation ensembles. *Math. Geosci.*, 47(7):771–789, 2015.

[3] H. Cho, D. Venturi, and G.E. Karniadakis. Karhunen–Loève expansion for multi-correlated stochastic processes. *Probabilistic Eng. Mech.*, 34:157–167, 2013.

[4] Y. Cong, B. Chen, and M. Zhou. Fast simulation of hyperplane-truncated multivariate normal distributions. *Bayesian Anal.*, 12(4):1017 – 1037, 2017.

[5] A. Cousin, H. Maatouk, and D. Rullière. Kriging of financial term-structures. *Eur. J. Oper. Res.*, 255(2):631–648, 2016.

[6] M.W. Davis. Generating large stochastic simulations - The matrix polynomial approximation method. *Math. Geol.*, 19(2):99–107, 1987.

[7] G. Golub and C.F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 1996.

[8] Y. Hoffman and E. Ribak. Constrained realizations of Gaussian fields: A simple algorithm. *ApJ*, 380:L5, October 1991.

[9] A.G. Journel and C.J. Huijbregts. *Mining geostatistics*. Academic Press, 1976.

[10] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE PAMI*, 12(1):103–108, 1990.

[11] P. J. Lenk and T. Choi. Bayesian analysis of shape-restricted functions using Gaussian process priors. *Stat. Sin.*, pages 43–69, 2017.

[12] M. Loève. *Elementary probability theory*. Springer, 1977.

[13] H. Maatouk and X. Bay. Gaussian process emulators for computer experiments with inequality constraints. *Math. Geosci.*, 49(5):557–582, 2017.

[14] H. Maatouk, X. Bay, and D. Rullière. A note on simulating hyperplane-truncated multivariate normal distributions. *Stat Probab Lett*, 191:109650, 2022.

[15] B. Matérn. *Spatial variation*, volume 36. Springer Science & Business Media, 1986.

[16] A.M. Panunzio, R. Cottereau, and G. Puel. Large scale random fields generation using localized Karhunen–Loève expansion. *AMSES*, 5(1):1–29, 2018.

[17] P. Ray, D. Pati, and A. Bhattacharya. Efficient Bayesian shape-restricted function estimation with constrained Gaussian process priors. *Stat. Comput.*, 30(4):839–853, 2020.

[18] H. Rue. Fast sampling of Gaussian Markov random fields. *J. R. Stat. Soc., B: Stat.*, 63(2):325–338, 2001.

[19] L. Wang. *Karhunen-Loève expansions and their applications*. London School of Economics and Political Science (United Kingdom), 2008.

[20] C.K. Williams and C.E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[21] J.T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M.P. Deisenroth. Pathwise conditioning of Gaussian processes. *JMLR*, 22(105):1–47, 2021.

[22] A. TA Wood and G. Chan. Simulation of stationary Gaussian processes in $[0,1]^d$. *J Comput Graph Stat*, 3(4):409–432, 1994.

[23] J. Zhang and B. Ellingwood. Orthogonal series expansions of random fields in reliability analysis. *J. Eng. Mech.*, 120(12):2660–2677, 1994.

[24] S. Zhou, P. Giulani, J. Piekarewicz, A. Bhattacharya, and D. Pati. Re-examining the proton-radius problem using constrained Gaussian processes. *Phys. Rev. C*, 99:055202, May 2019.