Sampling large hyperplane-truncated multivariate normal distributions

Hassan Maatouk, Didier Rullière, Xavier Bay

▶ **To cite this version:**

Hassan Maatouk, Didier Rullière, Xavier Bay. Sampling large hyperplane-truncated multivariate normal distributions. 2022. hal-03741860v1

HAL Id: hal-03741860
https://hal.science/hal-03741860v1

Preprint submitted on 2 Aug 2022 (v1), last revised 19 Sep 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 many fields (such as engineering). In this paper, simulating large multivariate normal distributions truncated on the intersection of a set of hyperplanes is investigated. The proposed methodology focuses on Gaussian vectors extracted from a Gaussian process (GP) in one dimension. 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. This step becomes expensive when the random vector is too large. To deal with this issue, the input domain is split in smallest subdomains where the eigendecomposition can be computed. By this strategy, the computational complexity is drastically reduced. The mean-square *truncation* and *block* errors have been calculated. Some numerical experiments are presented in order to study the efficiency of the proposed approach.

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

In this paper, simulating large multivariate normal (MVN) distributions truncated on the intersection of a set of hyperplanes is studied. This problem is widely used in Bayesian regression and quite related to simulate conditional Gaussian processes (GPs) [2, 5, 12, 18]. The standard approach is based on simulating from the posterior distribution using the location-scale transformation. One can compute a *scaling* matrix of the posterior covariance matrix and sample from a standard MVN distribution [13]. Computing a *scaling* matrix becomes expensive in higher dimensions. The computational complexity scales cubically with the dimension of the MVN random variable [8]. To handle with this, computing a polynomial or Lanczos approximation of a *scaling* matrix of the covariance matrix is possible [1, 6]. By simulating from the posterior distribution, we lost the information on the unconditional (precision or prior) covariance matrix.

The proposed methodology is quite different, where we do not need to compute the posterior covariance matrix and its decomposition. It is based on combining Karhunen-Loève expansion (KLE) and Matheron's update rule (MUR). It is however specific to Gaussian vectors extracted from a GP in dimension one.

In the first hand, the KLE can be seen as an efficient way to simulate random fields, which it is based on computing the eigendecomposition of the covariance operator [11]. In higher dimensions, the eigendecomposition becomes numerically heavy. To address this issue, one can split the input domain in smallest subdomains and conditioning the KLE coefficients in order to satisfy the given correlation between subdomains, as proposed in [3, 15].

In the other hand, the MUR which first appeared in geostatistics [10] and then in astrophysics [9] can be summarized in two steps. First, we simulate from the unconstrained MVN distribution. Second, we project into 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 [19] use the MUR for simulating conditional Gaussian processes. This method is called *pathwise conditioning*, which 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 map into the intersection of a set of

hyperplanes using the second step of the MUR. By this approach, one can simulate efficiently a large *hyperplane-truncated* MVN distribution.

The article is structured as follows: In Section 2, the KLE is briefly recalled. Additionally, the large scale KLE technique is presented with some numerical illustrations. The *truncation* as well as the *global block* errors between the proposed approach and the original random field are computed. Moreover, the extension to different lengths of subdomains is included. Section 3 is devoted to sampling *hyperplane-truncated* MVN distributions. The MUR and its extension to higher dimensions are presented. Numerical results to study the performance of the proposed approach are included.

2 Karhunen-Loève expansion

In this section, the Karhunen-Loève expansion is briefly recalled. Without loss of generality, we suppose that $\mathcal{D}$ is the interval $[0, 1]$ in $\mathbb{R}$.

2.1 Standard Karhunen-Loève expansion

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

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

Table 1: Some popular stationary covariance functions with their degree of smoothness used in the Machine Learning community [18].

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	$\exp\left(-\frac{\lvert x-x'\rvert}{\theta}\right)$	$\mathcal{C}^0$

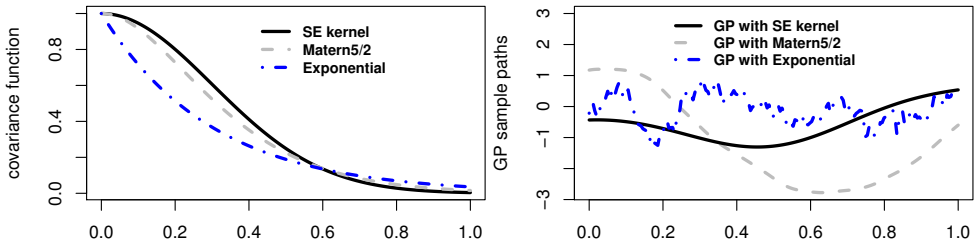

Figure 1: Some covariance functions (left) and associated GP sample paths (right). The correlation length θ is fixed to 0.3.

Table 1 shows some popular stationary covariance functions, ordered by decreasing degree of smoothness. These kernels have been widely used in the Machine Learning community [18]. In Figure 1, three covariance functions have been illustrated with the associated random process sample paths.

The eigendecomposition of the covariance function on the domain $\mathcal{D}$ [11] is:

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

The deterministic functions $\varphi_i(x)$ and the coefficients γ_i are respectively the eigen-functions and eigen-values 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 eigen-values 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 eigen-functions $\varphi_i(x)$ form a complete orthonormal basis functions set [17]. 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.

The random field $(Y(x))$ can be written as:

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

where the KLE coefficients ξ_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:

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

We have the following well-known result (see for instance [17]):

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:*

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

Proof. 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. □

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* Karhunen-Loève expansion. Let us recall that the mean-square *truncation* error ϵ_{KL}^2 is related to the sum of the eigen-values:

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

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 a $N \times N$ eigen-value problem. When the domain is huge and a fine discretization of the field is used, the eigendecomposition problem becomes expensive to solve, having $\mathcal{O}(N^3)$ complexity. The idea in the following sections is to split the domain $\mathcal{D}$ in smallest subdomains in order to reduce the complexity of the eigendecomposition problem. For instance, when the domain is split into two subdomains, the complexity becomes of order $\mathcal{O}((N/2)^3)$. Instead of resolving an eigendecomposition problem of order N, we only resolve an eigendecomposition of order $N/2$. By the stationary property of the random field, we show that the eigendecomposition of the first subdomain is only needed for any number of subdomains.

2.2 Large scale Karhunen-Loève expansion

To the best of our knowledge, the methodology presented in this section has first appeared in [3] and then in [15]. In this section, the random field $(Y(x))_{x\in\mathcal{D}}$ is assumed stationary. For simplicity, the domain $\mathcal{D} = [0, MS]$ is split in M equal sized subdomains $\mathcal{D}^{(m)} = ((m-1)S, mS]$ for any block parameter $m \in \{1, \ldots, M\}$. The extension to subdomains with different lengths has been investigated in Section 2.3. 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 eigen-values 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).

Let us mention that the following notation will be used in the entire paper:

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

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

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

Proof.

(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. In the first hand, from Proposition 1 and the stationary property of the covariance function, we have for any $t, t' \in \mathcal{D}^{(m)}$

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

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

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

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 $x, x' \in \mathcal{D}^{(m)}$

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

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: in the first hand, we have for any $(s, t) \in \mathcal{D}^{(1)} \times \mathcal{D}^{(2)}$,

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

In the second one, 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.

□

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 [21] and used in [3, 15]. The coupling matrix $\boldsymbol{K}^{m,m'}$ represents the correlation between the KLE coefficients sets of any two subdomains $\mathcal{D}^{(m)}$ and $\mathcal{D}^{(m')}$ for $m < m'$. 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. Let us recall that since the random field is assumed stationary and an equal sized subdomains with a uniform subdivision is used, the coupling matrix $\boldsymbol{K}$ is defined once between any two connected subdomains. In fact, we have for any $i, j = 1, \ldots, p$

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

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 means that the second item of Proposition 2 will be used when $m' = m+1$.

Let $(Y_m^{\perp}(x))$ and $(Y_{m+1}^{\perp}(x))$ are two independent sets covering respectively the $\mathcal{D}^{(m)}$ and $\mathcal{D}^{(m+1)}$ subdomains, 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 follows:

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

where $\boldsymbol{I}_p$ 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:*

$$\boldsymbol{\xi}^{(m)} = \boldsymbol{K}^\top \boldsymbol{\xi}^{(m-1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(m)}, \quad \forall m \in \{2, \ldots, M\}, \tag{7}$$

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 $m' \geq m$*

$$\begin{cases} \boldsymbol{K}^{m,m'} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}, \boldsymbol{\xi}^{(m')}\right) = \boldsymbol{K}^{m'-m}, \\ \boldsymbol{K}^{m,m} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}\right) = \boldsymbol{I}_p, \end{cases} \tag{8}$$

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

Proof.

- From Equation (7), 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 $\mathcal{D}^{(m)}$

$$\boldsymbol{K}^{m-1,m} = \mathrm{Cov}\left(\boldsymbol{\xi}^{(m-1)}, \boldsymbol{\xi}^{(m)}\right) = \boldsymbol{K}, \quad \forall m \in \{2, \ldots, M\}. \tag{9}$$

  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 (7) and (9). 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 $m \in \{2, \ldots, M\}$

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

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

□

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

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

Proof. The covariance between $Y_m(x)$ and $Y_{m'}(x')$ is a simple consequence of Equation (8). □

By construction, at each subdomain (except the two first ones), the conditional coefficients set is computed using the left-hand side previous conditional coefficients set and the right-hand side non conditional coefficients set. From (8), 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 $\mathcal{D}$:

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

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 eigen-values and eigen-functions of the correlation 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 [18], '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 [18] Section 4.2. In this paper, the triangular correlation function will be used:*

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

where θ is the correlation length parameter. However, in the proposed approach case this class of covariance functions with compact support does not resolve the problem of correlation error between the coefficients sets of unconnected subdomains. This is because with the proposed approach, the correlations between the conditioning coefficients sets of unconnected subdomains $\boldsymbol{K}^{m,m'}$ for $m' > m+1$ are not zero anymore.

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 $\mathcal{G}$.

Proof. For the mean-square *truncation* error (11), in the first hand, we have

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

In the second 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 [13]

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

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. □

Generally speaking, in the numerical examples of this paper and without loss of generality, the domain $\mathcal{D}$ is the interval $[0, 1]$. The Matérn covariance function with regularity parameter $\nu = 5/2$ (see Table 1) has been used [14], where the correlation length θ is fixed at 0.2 (except when mentioned). In that case, the mean-square truncation error (4) is equal to $\epsilon_{KL}^2 = 9.7 \times 10^{-6}$ when $p = 30$ (terms retained in the expansion). Other covariance functions

can be used such us Squared Exponential (SE), exponential covariance function (see Table 1).

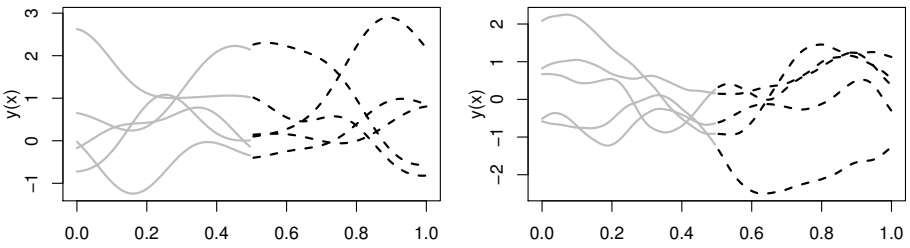

Figure 2: GP sample paths when the domain is split in two subdomains. Solid lines (before) and dashed lines (after) conditioning. $p = 5$ terms are retained in the expansion (*left*) versus $p = 30$ terms (*right*).

In Figure 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. Thus, each subdomain is discretized into fifty uniformly spaced points. The black dashed lines represent the path of $Y_2^p(x)$ on $(S, 2S]$ (i.e., the random process path after conditioning). The gray lines represents 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 versus $p = 30$ terms in the *right* panel. Let us mention that in the second part of the domain, the realizations are adjusted in order to follow the generation in the first part of the domain (see Figure 3 for more details).

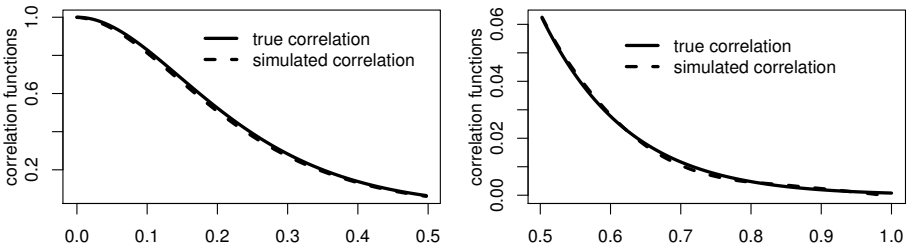

Figure 3: True and simulated correlation functions between $t = 0.5$ (resp. $t = 0$) and $t \in \mathcal{D}^{(2)}$ *left* panel (resp. *right* panel) when the domain is split into two subdomains. The simulated correlation is based on 1,000,000 replicates.

Figure 3 shows the true correlation function $C(|h|)$ at a distance $h \in [0, 0.5]$ (*left* solid line) and $h \in [0.5, 1]$ (*right* solid line) together with the

simulated correlation (dashed line) between $Y_2^p(0.5)$ (resp. $Y_1^p(0)$) and $Y_2^p(t)$ for $t \in \mathcal{D}^{(2)}$ *left* panel (resp. *right* panel) when $\mathcal{D}$ is split into two subdomains. The simulated correlation function has been computed using 1,000,000 replicates. The *left* (resp. *right*) panel verifies the theoretical result obtain in the first (resp. second) item of Proposition 2. Let us recall that the true correlation function C is the Matérn covariance with regularity parameter $\nu = 5/2$. Furthermore, $p = 30$ terms have been retained in the expansion, where the domain $\mathcal{D}$ is discretized into $N = 100$ equally spaced points. Thus, each subdomain is discretized into fifty equally spaced points.

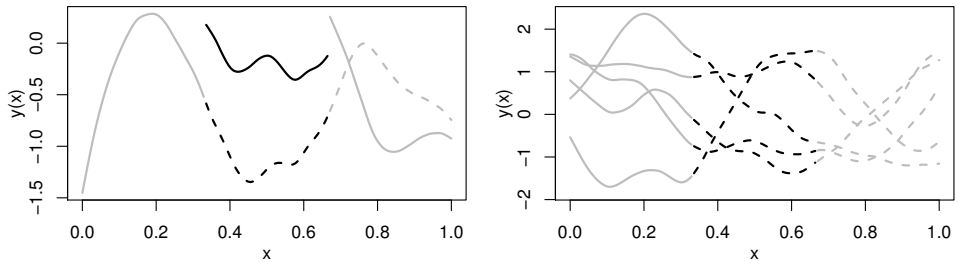

Figure 4: GP sample paths when the domain is split in three subdomains. Dashed (resp. solid) lines represent the paths after (resp. before) conditioning. One replicate (*left*) versus 5 replicates (*right*).

In Figure 4, the domain $\mathcal{D} = [0, 1]$ is decomposed into three subdomains (M=3) with equal lengths. It is discretized into $N = 150$ uniformly spaced points. So each subdomain is discretized into fifty uniformly spaced points. As before, $p = 30$ terms are retained in the expansion which lead to a truncation error of order $\epsilon_{KL}^2 = 9.7 \times 10^{-6}$. The solid lines (resp. dashed lines) represent the paths of the Gaussian process before (resp. after) conditioning. In the second and in the third subdomain, the realizations are adjusted in order to follow the generation in the first subdomain (see Figure 5 for more details).

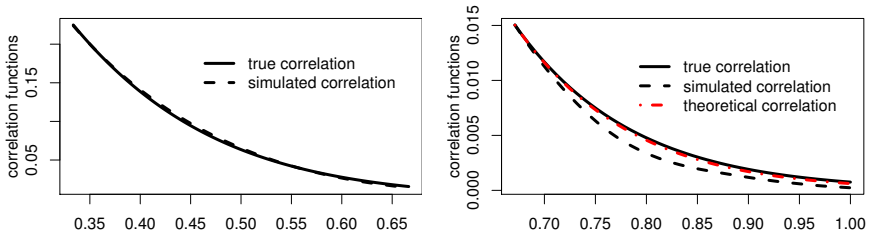

Figure 5: The correlation functions between $t = 0$ and $t \in \mathcal{D}^{(2)}$ (resp. $t \in \mathcal{D}^{(3)}$) *left* (resp. *right*) panel. The simulated correlation is based on 1,000,000 replicates. The theoretical correlation is computed from Equation (10).

In Figure 5, the domain $\mathcal{D}$ is decomposed into three subdomains. It is discretized into $N = 150$ equally spaced points. Thus, each subdomain is discretized into fifty equally spaced points. Furthermore, the number of terms retained in the expansion is fixed at $p = 30$. The black solid line 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 correlation function is the Matérn covariance with regularity parameter $\nu = 5/2$, where the length parameter is fixed at $\theta = 0.2$. The black dashed line represents the simulated correlation using 1,000,000 replicates. The red dash-dotted line in the *right* panel represents the theoretical correlation between $Y_1^p(0)$ and $Y_3^p(t)$ for $t \in \mathcal{D}^{(3)}$ using Equation (10) in Proposition 3. As expected, the simulated correlation between the first two subdomains coincides with the true one contrarily to the case between the first and the last subdomain. This is because by construction, the correlation between the first coefficients set $\boldsymbol{\xi}^{(1)}$ and the third one $\boldsymbol{\xi}^{(3)}$ is estimated by the coupling matrix square, i.e., $\mathrm{Cov}(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(3)}) = \boldsymbol{K}^2$ according to the second item of Proposition 3. For the clarity of the *left* panel, the theoretical correlation between $Y_1^p(0)$ and $Y_2^p(t)$ for $t \in \mathcal{D}^{(2)}$ has been omitted.

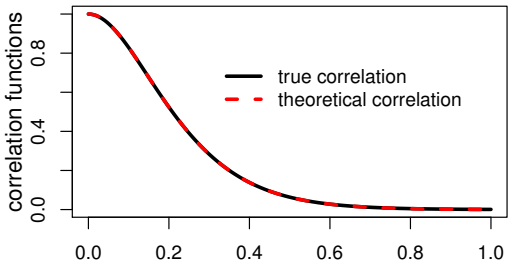

Figure 6: The true correlation function together with the theoretical one obtained from Equation (10) using the proposed approach on $\mathcal{D}$.

In the same setting of Figure 5, the black solid line in Figure 6 represents the true correlation function C on $\mathcal{D}$. However, the red dashed line represents the correlation function using the proposed approach and Equation (10). One can remark that the two correlation functions are very close with a root-mean-square error (RMSE) equal to 1.02×10^{-4}.

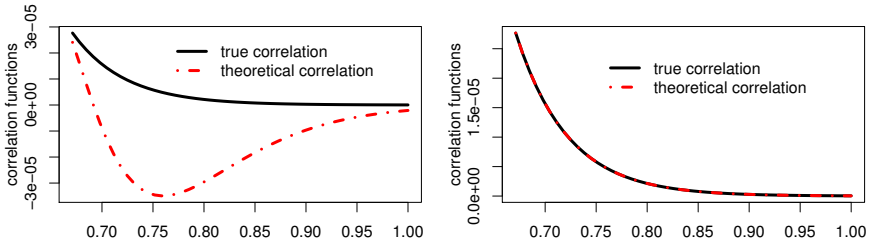

Figure 7: The correlation functions between $t = 0$ and $t \in \mathcal{D}^{(3)}$ with (resp. without) truncation in the *left* (resp. *right*) panel. The length parameter is fixed at $\theta = 0.1$. The theoretical correlation is obtained from Equation (10).

In Figure 7, the domain $\mathcal{D}$ is decomposed into three subdomains. It is discretized into $N = 150$ equally spaced points. Thus, each subdomain is discretized into fifty equally spaced points. The black solid line represents the true correlation function $C(|h|)$ for $h \in \mathcal{D}^{(3)}$. However, the red dash-dotted line represents the theoretical correlation between $Y_1^p(0)$ and $Y_3^p(t)$ for $t \in \mathcal{D}^{(3)}$ obtained from Equation (10) in Proposition 3. In the *left* panel, $p = 30$ terms have been retained in the expansion (i.e., truncated expansion). However, in the *right* panel, all terms $p = 50$ have been retained in the expansion (i.e., expansion without truncation). When the proposed approach is used without truncation, the correlation between unconnected subdomains follows the true correlation function C. Let us recall that the true correlation

function is the Matérn covariance with regularity parameter $\nu = 5/2$ and length parameter θ fixed at 0.1.

Table 2: Root-mean-square error on $\mathcal{D}$ between the true correlation 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 correlation function C and the approximated correlation function obtained from the proposed approach. The domain $\mathcal{D}$ is decomposed into four subdomains and it is discretized into $N = 200$ equally spaced points. Thus, each subdomain is discretized into fifty equally spaced points. Furthermore, the proposed approach without truncation has been used, this means that $p = 50$. The RMSE decreases when the length parameter θ decreases as well as when the degree of smoothness ν of the Matérn family of correlation functions decreases. Let us mention that the Markovian Exponential covariance function (see [18], Section 4.5 page 102) provides negligible errors compared to other covariance kernels for different values of θ as well as 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 get $C(|h|) = 1 - |h| > 0$ 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}

Under the same setting of Table 2, Table 3 shows the mean-square *global block* error (12) on $\mathcal{D}$ between the original random process Y and the proposed one $Y_{1:M}$ with $M = 4$. As for the RMSE of the correlation function, the *global block* error decreases when the length parameter decreases as well as when the degree of smoothness ν of the Matérn family of correlation functions decreases. Again, the Markovian Exponential covariance function provides negligible errors for different values of θ and for different number of

subdomains M. Let us mention that the domain $\mathcal{D}$ is decomposed into four subdomains (i.e. $M = 4$). The *global block* error decreases also when the number of subdomains decreases. For instance, with only two subdomains ($M = 2$), this error $\epsilon^2_{B,M}$ is of order 5.24×10^{-23} in the case of Matérn covariance function with regularity parameter $\nu = 5/2$ and length parameter $\theta = 0.1$. In the same setting, when $M = 10$, we get $\epsilon^2_{B,M} = 2.89 \times 10^{-14}$.

2.3 Subdomains with different lengths

The methodology presented in Section 2.2 can also be extended to the case when the domain is split in subdomains with different lengths. In that case, the eigendecomposition and the coupling matrix should be computed at each subdomain. For example, suppose that the domain $\mathcal{D} = [0, 1]$ is split in two subdomains $[0, S] \cup [S, 1]$ as shown in Figure 8 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

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

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

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

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

$$\begin{cases} Y_1^{\perp}(x) = \sum_{i=1}^p \sqrt{\lambda^{(1)}_i}\zeta^{(1)}_i\phi^{(1)}_i(x), & \text{if } x \in [0, S] \\ Y_2^p(x) := \sum_{j=1}^p \sqrt{\lambda^{(2)}_j}\xi^{(2)}_j\phi^{(2)}_j(x), & \text{if } x \in (S, 1]. \end{cases}$$

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

$m' > m$ is equal to

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

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

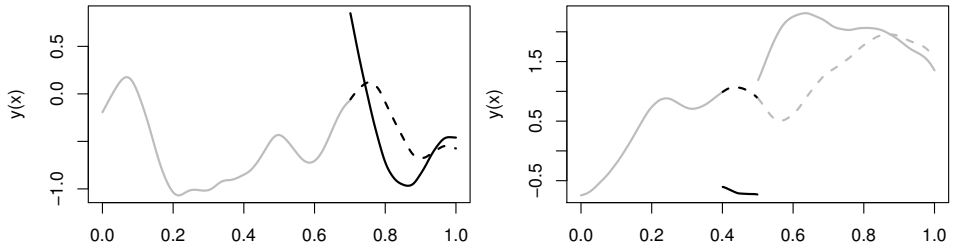

Figure 8: GP sample paths when the domain is split in two (*left* panel) and three (*right* panel) subdomains with different lengths.

In Figure 8, the domain $\mathcal{D}$ is split in two (resp. three) subdomains in the *left* (resp. *right*) panel with different lengths. The solid lines (resp. dashed lines) represent the paths before (resp. after) conditioning. The Matérn covariance function with regularity parameter $\nu = 5/2$ has been used where the length parameter θ is fixed at 0.2.

In the next section, the problem of sampling *hyperplane-truncated* MVN distributions is considered and the MUR with the results obtained in Section 2 is used to simulate from the posterior distribution without computing the posterior covariance matrix and its decomposition.

3 *Hyperplane-truncated* MVN distributions

3.1 Sampling in low dimensions

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

$$\eta \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Gamma}), \quad T = \{\boldsymbol{x} \in \mathbb{R}^N | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times N}$, $\boldsymbol{y} \in \mathbb{R}^n$ and $\text{rank}(\boldsymbol{A}) = n$.

The well-known result for computing the posterior distribution of conditional Gaussian vectors with linear equality constraints is recalled in the following proposition.

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

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

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

are the conditional mean and covariance matrix respectively.

Proof. See for instance [13]. □

The standard (or direct) approach that simulate $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{C})$ consists of computing a *scaling* matrix of the posterior covariance matrix $\boldsymbol{C}$ denoted here $\boldsymbol{S} \in \mathbb{R}^{N\times k}$, for any $k \in \mathbb{N}^*$ verifying $\boldsymbol{SS}^\top = \boldsymbol{C}$ and using

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

where $\epsilon \sim \mathcal{N}(\boldsymbol{0}_N, \boldsymbol{I}_N)$ is a N-dimensional standard Gaussian vector. To compute a *scaling* matrix $\boldsymbol{S}$, one can use Cholesky factorization [16] or eigen-decomposition. In general, the computational complexity of computing a *scaling* matrix is of order $\mathcal{O}(N^3)$ [8].

Hereafter we show how simulating *hyperplane-truncated* MVN distributions is possible without computing the covariance matrix of the posterior distribution and its decomposition. This allows to sample hyperplane truncated MVN distributions when the dimension N of the prior Gaussian vector η is high.

3.2 Sampling in high-dimensions

The term 'high-dimension' refers to the case when the dimension N of the prior Gaussian vector η is too high. We have seen that due to complexity $\mathcal{O}(N^3)$, the use of direct approaches was difficult when N is large. Thus, we propose here an original method to cope with this problem.

Let us recall first the classical Matheron update rule [10].

Proposition 6 (MUR distribution)**.** *Let* η *be a* N*-dimensional MVN random vector with mean* $\boldsymbol{\mu}$ *and 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*

$$\{\eta | \boldsymbol{A}\eta = \boldsymbol{y}\} \overset{d}{=} \eta + (\boldsymbol{A\Gamma})^\top (\boldsymbol{A\Gamma A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A}\eta). \tag{13}$$

The proof of Proposition 6 is given in the Appendix. As mentioned in [19], a key difference with the standard approach (Proposition 5) is that we now sample before conditioning, rather than after. This remark is the key of the idea presented in this paper in order to update the MUR in higher dimensions. Equation (13) can be seen as a deterministic transformation of the Gaussian vector η.

Algorithm 1: Sampling scheme by MUR of $\eta \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, where $T = \{\boldsymbol{x} \in \mathbb{R}^N | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$.

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

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

Corollary 2. *Suppose η is simulated with Algorithm 1, then it is distributed as $\eta \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, where $T = \{\boldsymbol{x} \in \mathbb{R}^N | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$, $\boldsymbol{A} \in \mathbb{R}^{n \times N}$, $\boldsymbol{y} \in \mathbb{R}^n$, and $\operatorname{rank}(\boldsymbol{A}) = n < N$.*

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

From Algorithm 1 and Proposition 2, simulating an *hyperplane-truncated* MVN distribution using MUR is made by two steps: first, we draw w from unconstrained MVN as $w \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ and second, we map it to η on the intersection of a set of hyperplanes by

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

By the Matheron's update rule, we sample from the unconstrained MVN which is an advantage. For example, when the unconstrained covariance (precision) matrix $\boldsymbol{\Gamma}$ is diagonal (see the motivating Example 1 below). Moreover, if $\boldsymbol{\Gamma}$ is generated from stationary covariance function, we preserve the stationary property in sampling unlike the direct approach based on sampling from the posterior distribution where the posterior covariance is not stationary anymore. This advantage has been used in the block splitting approach developed in Section 2.2, where the coupling matrix has been computed ones

from any arbitrary sized subdivision. Furthermore, only the eigendecomposition of the first subdomain is used. In this paper, we thus propose to combine both the KLE update (block splitting approach developed in Section 2.2) and the MUR. By this new approach, the computational complexity of sampling an *hyperplane-truncated* MVN distribution when N is large is drastically reduced. Let us mention that when the domain is split into two subdomains, the proposed approach become exact. However, in the case when the domain is split into more than two subdomains ($M > 2$), we get an error due to the approximation (see Proposition 4 in Section 2.2).

Algorithm 2 shows the different steps to simulate an *hyperplane-truncated* MVN distribution using the proposed approach when the domain is split in only two subdomains (i.e., $M = 2$). Let us recall that this algorithm is specific to the case where η is a Gaussian vector extracted from a stationary GP.

Algorithm 2: Sampling scheme of $\eta \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Gamma})$, where $T = \{\boldsymbol{x} \in \mathbb{R}^N | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$ when the domain is split in two subdomains (i.e., $M = 2$).

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

- Generating $w \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$:
  - sample $\boldsymbol{\zeta}^{(1)}, \boldsymbol{\zeta}^{(2)} \sim \mathcal{N}(\boldsymbol{0}_p, \boldsymbol{I}_p)$;
  - compute matrices $\boldsymbol{K}$ and $\boldsymbol{L}$;
  - compute $\boldsymbol{\xi}^{(2)} = \boldsymbol{K}^\top \boldsymbol{\zeta}^{(1)} + \boldsymbol{L}\boldsymbol{\zeta}^{(2)}$;
  - compute $w_1 = \sum_{i=1}^p \sqrt{\lambda_i}\phi_i \zeta_i^{(1)}$ and $w_2 = \sum_{i=1}^p \sqrt{\lambda_i}\phi_i \xi_i^{(2)}$;
  - assemble $w = \boldsymbol{\mu} + [w_1, w_2]^\top$;
- return $\eta = w + (\boldsymbol{A}\boldsymbol{\Gamma})^\top (\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top)^{-1}(\boldsymbol{y} - \boldsymbol{A}w)$ which can be realized by
  - solve $\boldsymbol{\alpha}$ such that $(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top)\boldsymbol{\alpha} = \boldsymbol{y} - \boldsymbol{A}w$;
  - return $\eta = w + (\boldsymbol{A}\boldsymbol{\Gamma})^\top \boldsymbol{\alpha}$.

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

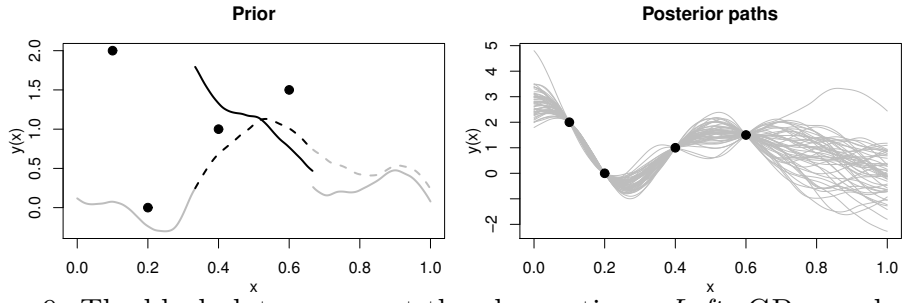

Figure 9: The black dots represent the observations. *Left*: GP sample paths prior where the domain is split in three subdomains. Solid lines (before) and dashed-line (after) conditioning. *Right*: the posterior paths (gray solid lines) obtained as in Equation (13). Each path is a deterministic transformation of the (random) dashed-line on the *left* panel.

Figure 9 illustrates the sampling procedure using the proposed approach. The black dots represent the observations. *Left* panel: one sample path of the Gaussian prior together with the observations. As in the *left* panel of Figure 4, the domain $\mathcal{D}$ is split into three subdomains. The solid (resp. dashed) lines represent the prior before (resp. after) conditioning using the block splitting approach developed in Section 2.2. *Right* panel: the posterior sample paths (gray lines) are obtained using the MUR as in Equation (13).

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

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

Table 4 summarizes the computational complexity of the naive MUR Algorithm 1 and the proposed approach Algorithm 2 when the domain $\mathcal{D}$ is split into M arbitrary subdomains. One can see that the proposed approach reduces the complexity. If $M \geq 3$, the reduced of complexity is related to an error due to the approximation (see Proposition 4 in Section 2.2).

3.3 Numerical illustrations

The aim of this section is to study the performance of the proposed approach. The following example is a favorable case for simulating an *hyperplane-*

truncated MVN distribution by the MUR instead of the direct approach based on the posterior distribution as in Proposition 5.

Example 1 (Motivating example)**.** *Suppose that we need to simulate from the following hyperplane-truncated MVN distribution:*

$$\eta \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{I}_N), \quad T = \{\boldsymbol{x} \in \mathbb{R}^N \ : \ \mathbf{1}^\top \boldsymbol{x} = 1\},$$

where $\boldsymbol{\mu} \in \mathbb{R}^N$, $\mathbf{1}^\top \boldsymbol{x} = \sum_{j=1}^N x_j$ *and* $\boldsymbol{I}_N$ *is the* $(N \times N)$ *identity matrix. The sampling scheme can be realized by the following two steps:*

- *sample* $w \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}_N)$*;*
- *return* $\eta = w + (1 - \mathbf{1}^\top w)\mathbf{1}$*.*

The two steps follow Algorithm 1 and Proposition 6, where $\boldsymbol{\Gamma} = \boldsymbol{I}_N$, $\boldsymbol{A} = \mathbf{1} \in \mathbb{R}^N$ *and* $\boldsymbol{y} = 1$*. In that case, the MUR is an efficient way to simulate from the hyperplane-truncated MVN distribution. This is because we sample from the standard normal distribution instead of computing the covariance matrix of the posterior distribution and its decomposition.*

Example 2 (Computational illustrations)**.** *Hereafter, the elements of* $\boldsymbol{\mu}$, $\boldsymbol{A}$ *and* $\boldsymbol{y}$ *are sampled from* $\mathcal{N}(0, 1)$*. The unconditional covariance (precision) matrix* $\boldsymbol{\Gamma}$ *is generated using the Matérn* $\nu = 5/2$ *covariance function with correlation length* $\theta = 0.2$*. In that case, the covariance matrix* $\boldsymbol{\Gamma}$ *has no special structure (dense matrix) and the Cholesky factorization has complexity of order* $\mathcal{O}(N^3)$*.*

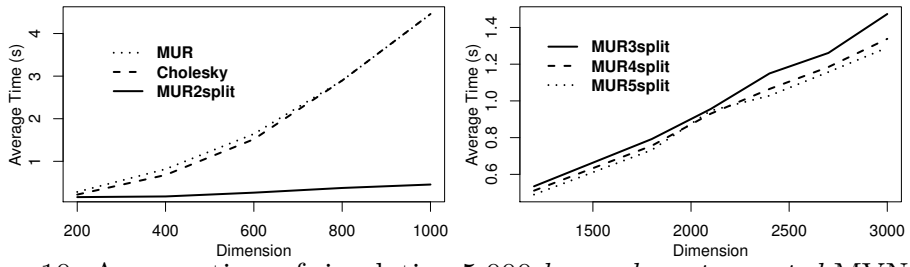

Figure 10: Average time of simulating 5,000 *hyperplane-truncated* MVN samples over twenty random trials, when the number of the data dimension N increases and the constraints dimension is fixed at $n = 20$.

The computation time of generating 5,000 hyperplane-truncated MVN distributions averaged over twenty random trials is shown in Figure 10. The constraints dimension is fixed at $n = 20$ *and the data dimension* N *increases.*

Left panel: the Cholesky factorization and the Matheron's update rule have been compared to the proposed approach when the domain is split in just two subdomains. The later has a clear advantage over both the Cholesky factorization and MUR. Right panel: the black solid line represents the average time in second of simulating 5,000 hyperplane-truncated MVN distributions using the proposed approach when the input domain is split into three subdomains. The black dashed (resp. dotted) line represents the average time in second using the proposed approach with four subdomains (resp. five subdomains). The computational times decreases as function of the number of subdomains. Let us finally recall that $p = 30$ *terms have been retained in the expansion. The proposed approach can be faster when* p *is less than* 30*. This is related to the truncation error (see Proposition 4).*

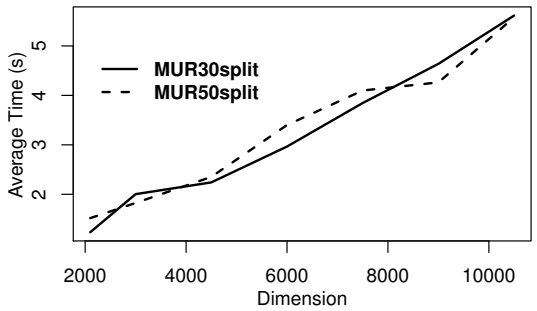

Figure 11: Average time of simulating 5,000 *hyperplane-truncated* MVN samples over twenty random trials, when the number of data dimension N increases and the dimension of the set of constraints is fixed at $n = 20$.

In Figure 11, the dimension of the set of constraints is fixed at $n = 20$ *and the data dimension* N *increases from 2,000 to 10,500. In that case, the naive MUR and Cholesky decomposition become numerically heavy. This is because we just illustrate the average time in second of simulating 5,000 hyperplane-truncated MVN samples over twenty random trials using the proposed approach when the domain is split in thirty subdomains (black solid line) and fifty subdomains (black dashed line). By the proposed approach, one can generate 5,000 Gaussian vectors of size 10,500 conditionally to an hyperplane of dimension* $n = 20$ *in around five seconds. Let us mention that we do not need a high decomposition of* $\mathcal{D}$ *to get a good performance in terms of computational time. Indeed, when the domain is decomposed into fifty subdomains the computational time is too close to the case when it is decomposed into thirty subdomains.*

Conclusion

In this paper, a new efficient approach to simulate large multivariate normal distribution truncated on the intersection of a set of hyperplanes is presented. The main idea is to combine both Karhunen-Loève expansion and Matheron's update rule. The proposed method is specific to the case where the Gaussian vector is extracted from a stationary GP. Firstly, we simulate efficiency from the unconstrained MVN distribution using the cross-correlated splitting subdomains. By the stationary property, only the eigendecomposition of the first subdomain is needed. Secondly, we map on the intersection of a set of hyperplanes using the Matheron's update rule. By the proposed approach, the computational complexity is drastically reduced. The mean-square *truncation* and *global block* errors have been calculated. The performance of the proposed approach has been studied through numerical examples.

Appendix

Before showing the proof of Proposition 6, let us give the following lemma.

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:*

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

where f is a measurable function of W_2. Then

$$\{W_1 | W_2 = \beta\} \overset{d}{=} f(\beta) + W_3.$$

Proof of Lemma 1. The proof is given in [19]. □

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

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

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

get

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

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

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

Table 5: The computational complexity of Algorithm 1. The parameter n and N represent respectively the dimension of the set of constraints and of the prior Gaussian vector η.

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	$\mathcal{O}(pN/2)$
$\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\top$	$\mathcal{O}(nN^2)$
$\boldsymbol{\alpha}$	$\mathcal{O}(\max(nN,n^3))$
η	$\mathcal{O}(nN)$
Summary	$\mathcal{O}(\max(n^3,(N/2)^3))$

Table 6: The computational complexity of Algorithm 2 when the domain $\mathcal{D}$ is split in two subdomains $M=2$. The parameters n and N represent respectively the dimension of the set of constraints and of the prior Gaussian vector η.

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. *Statistics and Computing*, 23(4):501–521, 2013.

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

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

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

[5] A. Cousin, H. Maatouk, and D. Rullière. Kriging of financial term-structures. *European Journal of Operational Research*, 255(2):631–648, 2016.

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

[7] T.E. Fricker, J.E. Oakley, and N.M. Urban. Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55(1):47–56, 2013.

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

[9] Y. Hoffman and E. Ribak. Constrained realizations of Gaussian fields: A simple algorithm. *The Astrophysical Journal*, 380:L5, October 1991.

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

[11] M. Loève. Elementary probability theory. In *Probability theory i*, pages 1–52. Springer, 1977.

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

[13] H. Maatouk, X. Bay, and D. Rullière. A note on simulating hyperplane-truncated multivariate normal distributions. February 2022. To appear in Statistics and Probability Letters.

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

[15] A.M. Panunzio, R. Cottereau, and G. Puel. Large scale random fields generation using localized Karhunen–Loève expansion. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):1–29, 2018.

[16] H. Rue. Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):325–338, 2001.

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

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

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

[20] S. Xiong. The reconstruction approach: From interpolation to regression. *Technometrics*, 63(2):225–235, 2021.

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