



**HAL**  
open science

# Large scale Gaussian processes with Matheron's update rule and Karhunen-Loève expansion

Hassan Maatouk, Didier Rullière, Xavier Bay

► **To cite this version:**

Hassan Maatouk, Didier Rullière, Xavier Bay. Large scale Gaussian processes with Matheron's update rule and Karhunen-Loève expansion. A. Hinrichs, P. Kritzer, F. Pillichshammer (eds.). Monte Carlo and Quasi-Monte Carlo Methods 2022. Springer Verlag, inPress. hal-03909542v2

**HAL Id: hal-03909542**

**<https://hal.science/hal-03909542v2>**

Submitted on 7 Apr 2023

**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.

# Large scale Gaussian processes with Matheron's update rule and Karhunen-Loève expansion

Hassan Maatouk and Didier Rullière and Xavier Bay

**Abstract** Gaussian processes have become essential for nonparametric function estimation and are widely used in many fields, like machine learning. In this paper, large scale Gaussian process regression (GPR) is investigated. This problem is related to the simulation of high-dimensional Gaussian vectors truncated on the intersection of a set of hyperplanes. The main idea is to combine both Matheron's update rule (MUR) and Karhunen-Lovève expansion (KLE). First, by the MUR we show that simulating from the posterior distribution can be achieved without computing the posterior covariance matrix and its decomposition. Second, by splitting the input domain into smallest nonoverlapping subdomains, the KLE coefficients are conditioned in order to guarantee the correlation structure in the entire domain. The parallelization of this technique is developed and the advantages are highlighted. Through this, the computational complexity is drastically reduced. The mean-square global *block* error is computed. It provides accurate results when using a family of covariance functions with compact support. Some numerical examples to study the performance of the proposed approach are included.

---

Hassan Maatouk  
CY Cergy Paris Université, CY Tech, Laboratoire AGM, Site du Parc, 95011 Cergy-Pontoise,  
France, e-mail: hassan.maatouk@cyu.fr

Didier Rullière  
Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol,  
F-42023 Saint-Étienne, France e-mail: drulliere@emse.fr

Xavier Bay  
Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol,  
F-42023 Saint-Étienne, France e-mail: bay@emse.fr

## 1 Introduction

Due to their flexibility, Gaussian processes (GPs) are widely used in many fields like geostatistics [5, 9], finance [1, 7] and econometrics [4] and have become very popular in the context of machine learning [16, 17]. In this paper, large scale Gaussian process regression (GPR) is investigated. This problem is related to the simulation of high-dimensional Gaussian vectors constrained on a set of intersection of hyperplanes. The direct approach is based on computing the posterior distribution and using the location scale transformation of the posterior covariance matrix (i.e., the *scaling* matrix). After computing the *scaling* matrix, we sample from a standard multivariate normal (MVN) distribution [11]. Computing the *scaling* matrix is possible via for example eigendecomposition or Cholesky factorization. When the dimension of the Gaussian vector is high, this approach becomes numerically heavy. This is due to the fact that the computational complexity scales cubically with the dimension of the random vector [6].

The methodology presented in this paper is quite different. It is based on combining both Matheron's update rule (MUR) and Karhunen-Loève expansion (KLE). In the first hand, the MUR, initially discovered in geostatistics [9] and later in astrophysics [8] is developed. Contrarily to the direct approach, it is based on generating from the prior distribution and adding an update part to obtain the target posterior distribution. The advantage of this approach is that we sample before conditioning rather than after. Therefore, there is no need to compute the posterior covariance matrix and its decomposition. In [3], the MUR is used to efficiently sampling MVN distribution whose covariance (precision) matrix can be decomposed as a positive-definite matrix minus (plus) a low-rank symmetric matrix. The main idea is to sample from a block diagonal covariance matrix. Recently, in the context of machine learning, the authors in [17] use the MUR for GPR. This approach is denoted *pathwise conditioning*. In the second hand, the KLE can be seen as an efficient way to sample random fields, which is based on computing the eigendecomposition of the covariance operator [10]. In high dimensions, the eigendecomposition becomes numerically heavy. To address this problem, the idea is to split the input domain into smallest nonoverlapping subdomains and to condition the KLE coefficients in order to respect the given correlation in the entire domain [2, 13].

In the present paper, we investigate the advantage of the MUR in the context of GPR. Then, we develop the large scale KLE for sampling the prior distribution. The parallelization of this approach is studied, resulting in a significant reduction of computational complexity. Finally, the mean-square global *block* error introduced by the proposed approach is calculated.

The article is structured as follows: in Sect. 2 GPR is briefly reviewed. Section 3 is devoted to the MUR in the context of GPR. In Sect. 4, the large scale KLE is developed where the parallelization is investigated and the introduced errors are computed. In Sect. 5, the performance of the proposed approach is shown through numerical examples.

## 2 Gaussian process regression

For any  $\mathbf{x} \in \mathbb{R}^d$ , suppose  $(Z(\mathbf{x}))$  is a GP with mean function  $\mu$  and covariance function  $k$ , i.e.,  $Z \sim \mathcal{GP}(\mu, k)$ . Then

$$Z(\mathbf{x}) = \mu(\mathbf{x}) + Y(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^d,$$

where  $(Y(\mathbf{x}))$  is a zero-mean GP with covariance function  $k$ , i.e.,  $Y \sim \mathcal{GP}(0, k)$

$$k(\mathbf{x}, \mathbf{x}') = \text{Cov}(Y(\mathbf{x}), Y(\mathbf{x}')), \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d.$$

Given a training data set  $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$  of size  $n$ , where  $\mathbf{x}_i$  denotes an input vector of dimension  $d$  and  $y_i$  denotes a scalar output (data). The input vectors are aggregated in the  $n \times d$  design matrix  $\mathbb{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top$  and the data are collected in the vector  $\mathbf{y} = [y_1, \dots, y_n]^\top$ , so we can write  $D = \{(\mathbb{X}, \mathbf{y})\}$ . The following regression problem is considered

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{\text{noise}}^2),$$

where  $f$  is an unknown latent function generates the data and  $\epsilon_i$  is an additive independent identically distributed Gaussian noise with constant noise variance  $\sigma_{\text{noise}}^2$  estimated via the maximum likelihood [16]. A GP prior distribution on the unknown function  $f$  is assumed. Conditionally on the data  $\mathbf{y} = [y_1, \dots, y_n]^\top$ , the conditional process remains a GP

$$\{Y|Y(\mathbb{X}) + \epsilon = \mathbf{y}\} \sim \mathcal{GP}(\mu_c, c),$$

where  $\epsilon = [\epsilon_1, \dots, \epsilon_n]^\top$  is a zero-mean Gaussian noise vector and the conditional mean and covariance functions  $\mu_c$  and  $c$  are given as follows:

$$\begin{aligned} \mu_c(\mathbf{x}) &= \mathbb{E}[Y(\mathbf{x})|\mathbf{y}] = k(\mathbf{x}, \mathbb{X})^\top (k(\mathbb{X}, \mathbb{X}) + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} \mathbf{y}; \\ c(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbb{X})^\top (k(\mathbb{X}, \mathbb{X}) + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} k(\mathbf{x}', \mathbb{X}); \end{aligned} \quad (1)$$

with  $\mathbf{I}_n$  the  $n \times n$  identity matrix. Let us recall that  $k(\mathbb{X}, \mathbb{X})$  is the covariance matrix of  $Y(\mathbb{X})$  and  $k(\mathbf{x}, \mathbb{X})$  is the vector of covariance between  $Y(\mathbf{x})$  and  $Y(\mathbb{X})$ .

In the simple special case where the observations are noise-free [14], that is we know  $\{(\mathbf{x}_i, f_i) | i = 1, \dots, n\}$ , with  $f_i = f(\mathbf{x}_i)$ , the predictive equations for GPR (1) are conserved where we replace  $\sigma_{\text{noise}}^2$  by zero and  $\mathbf{y}$  by  $\mathbf{f}$ , with  $\mathbf{f} = [f_1, \dots, f_n]^\top$ .

Table 1 shows some popular covariance functions in one-dimensional case (i.e.,  $x, x' \in \mathbb{R}$ ). They are widely used in machine learning community [16], and ordered by decreasing degree of smoothness, where  $\theta$  is the correlation length parameter.

In the next section, the MUR is briefly reviewed, as well as, the advantages of this method for the GPR are highlighted.

**Table 1** Some popular covariance functions with their degree of smoothness [16]

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

### 3 Matheron's update rule for Gaussian process regression

As said in the introduction, the MUR presented in this section has first appeared in geostatistics [9]. Let us briefly recall this method. Suppose  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are jointly Gaussian random variables. Then, the random vector  $\mathbf{X}_1$  conditional on  $\mathbf{X}_2 = \mathbf{x}_2$  can be expressed as

$$\{\mathbf{X}_1 | \mathbf{X}_2 = \mathbf{x}_2\} \stackrel{d}{=} \mathbf{X}_1 + \Sigma_{\mathbf{X}_1, \mathbf{X}_2} \Sigma_{\mathbf{X}_2, \mathbf{X}_2}^{-1} (\mathbf{x}_2 - \mathbf{X}_2),$$

where  $\Sigma_{\mathbf{X}_1, \mathbf{X}_2} = \text{Cov}(\mathbf{X}_1, \mathbf{X}_2)$  is the covariance between  $\mathbf{X}_1$  and  $\mathbf{X}_2$ . As mentioned in [17], a key difference with the direct approach is that we now sample before conditioning, rather than after, which is the key of the main idea developed in the present paper.

In the following proposition, we give the well-known MUR result for GPR.

**Proposition 1 (MUR for GPR)**

Using previous notations,  $(Y(\mathbf{x}))$  is a zero-mean GP with covariance function  $k$ . Then,  $Y$  conditioned on data  $\{Y(\mathbb{X}) + \epsilon = \mathbf{y}\}$  can be expressed as follows:

$$\{Y | Y(\mathbb{X}) + \epsilon = \mathbf{y}\}(\cdot) \stackrel{d}{=} \underbrace{Y(\cdot)}_{\text{prior}} + \underbrace{k(\cdot, \mathbb{X})^\top (k(\mathbb{X}, \mathbb{X}) + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} (\mathbf{y} - Y(\mathbb{X}))}_{\text{update}}, (2)$$

where  $\mathbf{y} - Y(\mathbb{X})$  represents the residual.

The sampling scheme of the posterior distribution  $\{Y | Y(\mathbb{X}) + \epsilon = \mathbf{y}\}(\cdot)$  using the MUR is given in the following algorithm.

---

**Algorithm 1:** Sampling scheme by MUR of  $\{Y | Y(\mathbb{X}) + \epsilon = \mathbf{y}\}(\cdot)$

---

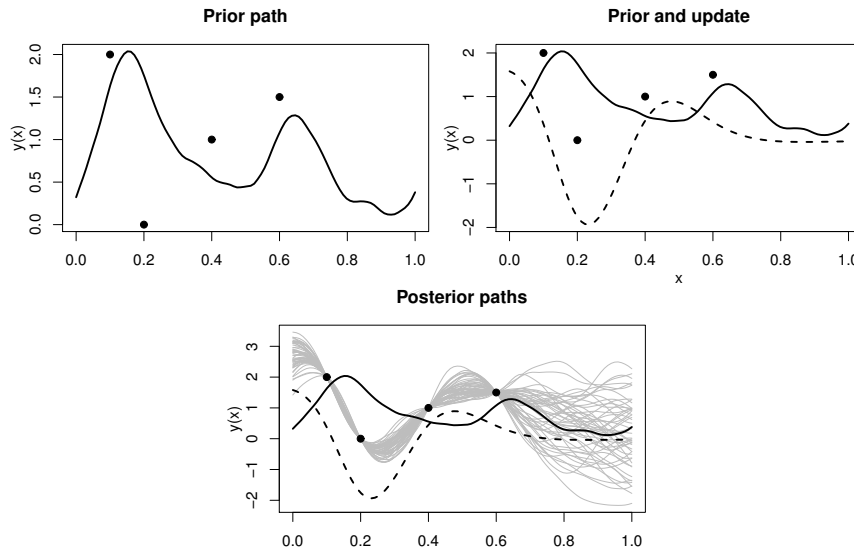
- sample  $Y(\cdot) \sim \mathcal{GP}(0, k)$ ;
  - return  $Y(\cdot) + k(\cdot, \mathbb{X})^\top (k(\mathbb{X}, \mathbb{X}) + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} (\mathbf{y} - Y(\mathbb{X}))$ .
- 

**Corollary 1** Suppose  $Y$  is simulated with Algorithm 1, then it is distributed as  $\{Y | Y(\mathbb{X}) + \epsilon = \mathbf{y}\}$ .

**Proof** The proof is a simple consequence of Proposition 1.

Let us give some remarks:

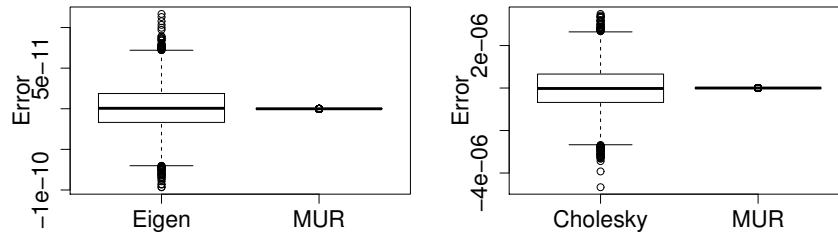
- By the MUR, we sample from the prior (first step in Algorithm 1) which is an advantage especially when the corresponding unconstrained (precision) matrix is diagonal or low-rank.
- By the MUR, the stationary property is preserved in the sampling procedure unlike the direct approach where the posterior covariance matrix must be computed which is not stationary anymore, cf. Sect. 4 below.
- The MUR provides numerical stability compared to standard approaches. This is because, after sampling the prior, we directly map it onto the set of observations, cf. Fig. 2 below.
- As demonstrated in Proposition 1, the MUR can be applied within the framework of GPR, whether the data is observed with or without noise.



**Fig. 1** Visual guide of the MUR. *Top left*: one path<sup>x</sup> of unconstrained Gaussian prior together with the observations (black dots). *Top right*: the corresponding update path from (2) has been added (black dashed-curve). *Bottom*: the posterior paths (gray solid curves) obtained by adding the prior and the update as in (2)

In Fig. 1, the visual guide of the MUR is presented. The simple case where the observations are noise-free is considered. The Matérn covariance function with regularity parameter  $\nu = 5/2$  and correlation length parameter  $\theta = 0.2$  (cf. Table 1) is used. *Top left*: the prior (black solid curve) zero-mean GP  $Y \sim \mathcal{GP}(0, k)$  together with the observations (black dots) have been shown. Let us mention that the prior does not interpolate the observations. *Top right*: the update part (black dashed-curve)

of the MUR (2) has been added. As expected, the update part follows the trend of the observations. *Bottom*: the posterior paths (gray solid curves) have been illustrated by adding the prior to the update as in (2). Each posterior sample path is obtained by adding a prior sample path to the update one. The posterior sample paths verify the interpolation conditions. One sample path of the prior and the update is shown for clarity. However, fifty sample paths of the posterior distribution are shown in the bottom panel of Fig. 1.



**Fig. 2** The numerical error of the residual  $Y(X) - \mathbf{f}$  using different approaches. For the Cholesky approach, a nugget effect of order  $10^{-12}$  has been added in order to avoid numerical problems

The stability of the MUR is now investigated. In Fig. 2, the boxplot of the numerical residual error  $Y(\mathbb{X}) - \mathbf{f}$  using the MUR and different standard approaches are shown for ten thousand replicates. Let us mention that a nugget effect of order  $10^{-12}$  was added to the Cholesky factorization to avoid numerical problems. As expected, the MUR provides accurate results. Additionally, based on numerical experiments, the MUR outperforms eigendecomposition and Cholesky factorization in terms of stability. This is because in the second step of the MUR algorithm, we map directly onto the set of observations.

When the input domain  $\mathcal{D}$  is large and a fine distretization is used, simulating the prior (first step in Algorithm 1) using standard approaches (eigendecomposition and Cholesky factorization) becomes numerically heavy. The computational complexity grows cubically with the dimension of the associated Gaussian vector [6]. To sidestep this problem, we show in the next section how the prior GP can be simulated efficiently using the KLE update.

#### 4 Karhunen-Loève expansion update

Let us briefly recall the standard KLE [10, 15]. In this section, the GP  $(Y(x))_{x \in \mathcal{D}}$  is assumed stationary. For simplicity, the input domain  $\mathcal{D}$  is supposed to be the unit interval  $[0, 1]$ .

#### 4.1 Standard Karhunen-Loève expansion

According to previous notations,  $(Y(x))_{x \in \mathcal{D}}$  is a zero-mean (GP), whose stationary covariance function is  $k(|x - x'|)$ , i.e.,  $Y \sim \mathcal{GP}(0, k)$ , where

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

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

$$\int_{\mathcal{D}} k(|x - x'|)\varphi_i(x)dx = \lambda_i\varphi_i(x'), \quad \forall i \in \mathbb{N}, \forall x, x' \in \mathcal{D}. \quad (3)$$

The deterministic functions  $\{\varphi_i\}$  and the coefficients  $\{\lambda_i\}$  are respectively the eigenfunctions and eigenvalues of the covariance function  $k(|\cdot|)$  on the domain  $\mathcal{D}$ . Let us recall that the eigenvalues are real and nonnegative since the covariance is symmetric and positive semi-definite:

$$\int_{\mathcal{D}} \int_{\mathcal{D}} k(|x - x'|)g(x)g(x')dx dx' \geq 0$$

for any  $g$  having finite  $L^2$  norm on  $\mathcal{D}$ . Let us recall also that the eigenfunctions  $\{\varphi_i(\cdot)\}$  form a complete orthonormal basis functions set [15]. This means that

$$\int_{\mathcal{D}} \varphi_i(x)\varphi_j(x)dx = \delta_{ij},$$

where  $\delta_{ij}$  represents the Kronecker delta, equal to 1 if  $i = j$  and 0 otherwise.

By the KLE, the GP  $(Y(x))$  can be written as (see e.g., [15] Sect. 1.2):

$$Y(x) = \sum_{i=1}^{+\infty} \sqrt{\lambda_i}\varphi_i(x)\zeta_i, \quad \forall x \in \mathcal{D},$$

where the KLE coefficients  $\{\zeta_i\}$  are zero-mean uncorrelated Gaussian random variables (thus independent) with unit variance, i.e.,  $\zeta_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ . The KLE coefficients are defined as the projection of the GP  $(Y(x))$  onto the KLE eigenfunctions:

$$\zeta_i = \frac{1}{\sqrt{\lambda_i}} \int_{\mathcal{D}} \varphi_i(x)Y(x)dx;$$

$$\mathbb{E}[\zeta_i] = 0 \quad \text{and} \quad \mathbb{E}[\zeta_i\zeta_j] = \delta_{ij}.$$

One can define an approximation of the GP  $(Y(x))$  on the domain  $\mathcal{D}$  using a truncated sum of  $p \geq 1$  terms, obtained from the KLE.

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



This approximation with a finite number of terms is called *truncated KLE*. Let us finish this section by recalling that the mean-square truncation error  $\epsilon_{\text{KL}}^2$  is related to the sum of the eigenvalues, as given by the following equation:

$$\epsilon_{\text{KL}}^2 = \frac{\mathbb{E} \left[ \int_{\mathcal{D}} (Y(x) - Y^p(x))^2 dx \right]}{\mathbb{E} \left[ \int_{\mathcal{D}} Y(x)^2 dx \right]} = 1 - \frac{\sum_{i=1}^p \lambda_i}{\sum_{i=1}^{+\infty} \lambda_i}.$$

This error decreases when the number of terms retained in the expansion increases.

When the domain  $\mathcal{D}$  is discretized into  $N$  equally spaced points, the eigendecomposition (3) leads to a  $N \times N$  eigenvalue problem. When the domain is huge and a fine discretization is used, the eigendecomposition becomes expensive with computational complexity of order  $\mathcal{O}(N^3)$ . The idea in the following sections is to split the input domain  $\mathcal{D}$  in smallest nonoverlapping subdomains. The KLE coefficients are conditioned so that  $(Y(x))$  follows the given correlation structure. By the stationary property of the GP  $(Y(x))_{x \in \mathcal{D}}$ , we show that only the eigendecomposition of the first subdomain is needed. The parallelization of this approach is studied in Sect. 4.3, resulting in a significant reduction of computational complexity.

## 4.2 Large scale Karhunen-Loève expansion

In this section, the GP  $(Y(x))_{x \in \mathcal{D}}$  is assumed stationary. The approach developed in [12, 13] is first considered. 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  $m \in \{1, \dots, M\}$ . The extension to subdomains with different lengths has been investigated (cf. Sect. 2.2 in [12]). The main idea is to generate  $M$  independent samples, each covering its corresponding subdomain, and then impose a correlation between the KLE coefficients of any two connected subdomains. We will denote by  $\bar{Y}$  the process constructed in this way and by  $\{\bar{\zeta}_i\}$  its corresponding KLE coefficients. For any  $m \in \{1, \dots, M\}$  and  $x \in \mathcal{D}_m$ , let

$$\bar{Y}_m(x) := \sum_{i=1}^{+\infty} \sqrt{\gamma_i} \phi_i(x - (m-1)S) \bar{\zeta}_i^{(m)}$$

be the proposed KLE covering the  $m^{\text{th}}$  subdomain  $\mathcal{D}_m$ , where the deterministic functions  $\{\phi_i(\cdot)\}$  and the coefficients  $\{\gamma_i\}$  are respectively the eigenfunctions and eigenvalues of the covariance function  $k(|\cdot|)$  on the first subdomain  $\mathcal{D}_1 = [0, S]$  and  $\{\bar{\zeta}_i^{(m)}\}$  are the conditional KLE coefficients. This implies that only the eigendecomposition of the first subdomain is necessary for the construction of the proposed approach. Before showing how the conditional coefficients  $\bar{\zeta}_i^{(m)}$  are constructed, let us give the following notation:

$$(\bar{Y}_m \cup \bar{Y}_{m'}) (x) = \bar{Y}_m(x) \mathbb{1}_{\mathcal{D}_m}(x) + \bar{Y}_{m'}(x) \mathbb{1}_{\mathcal{D}_{m'}}(x),$$

for any  $x \in \mathcal{D}$  and  $m, m' \in \{1, \dots, M\}$ , where  $\mathbb{1}_{\mathcal{D}_m}$  is the indicator function, equal to 1 if  $x \in \mathcal{D}_m$  and 0 otherwise. In practice, we suppose that  $p \geq 1$  terms are retained in the expansion:

$$\bar{Y}_m^p(x) = \sum_{i=1}^p \sqrt{\gamma_i} \phi_i(x - (m-1)S) \bar{\zeta}_i^{(m)}, \quad \forall x \in \mathcal{D}_m. \quad (4)$$

Let  $(Y_m^p(x))$  and  $(Y_{m+1}^p(x))$  be two independent processes covering respectively the  $\mathcal{D}_m$  and  $\mathcal{D}_{m+1}$  subdomains, for any  $m \in \{1, \dots, M-1\}$ . Thus,

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

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

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  $\zeta^{(m)} = \{\zeta_i^{(m)}\}_i$  and  $\zeta^{(m+1)} = \{\zeta_i^{(m+1)}\}_i$  are independently generated, the two GPs are uncorrelated:

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

for all  $x \in \mathcal{D}_m$  and  $t \in \mathcal{D}_{m+1}$ . Let us give the following result proved in [12].

**Proposition 2 (Distribution on blocks)**

Under the stationary property of the GP  $(Y(x))$ , we suppose that the  $m^{\text{th}}$  conditional coefficients set  $\bar{\zeta}^{(m)}$  is computed as follows

$$\bar{\zeta}^{(m)} = \mathbf{K}^\top \bar{\zeta}^{(m-1)} + \mathbf{L} \zeta^{(m)}, \quad \forall m \in \{2, \dots, M\}, \quad (5)$$

where  $\bar{\zeta}^{(1)} = \zeta^{(1)}$  and  $\mathbf{K}$  and  $\mathbf{L}$  are defined as

$$\mathbf{K}_{i,j} := \frac{1}{\sqrt{\gamma_i \gamma_j}} \int_0^S \int_0^S k(|x - x' - S|) \phi_i(x) \phi_j(x') dx dx'; \quad (6)$$

$$\mathbf{L} \mathbf{L}^\top := \mathbf{I}_p - \mathbf{K}^\top \mathbf{K};$$

with  $\mathbf{I}_p$  the  $p \times p$  identity matrix. Then

- the two processes  $Y$  and  $(\bar{Y}_{m-1} \cup \bar{Y}_m)$  have the same distribution on  $\mathcal{D}_{m-1} \cup \mathcal{D}_m$ , for any  $m \in \{2, \dots, M\}$ .
- For any  $m' \geq m$

$$\begin{cases} \text{Cov} \left( \bar{\zeta}^{(m)}, \bar{\zeta}^{(m')} \right) = \mathbf{K}^{m'-m} \\ \text{Cov} \left( \bar{\zeta}^{(m)} \right) = \mathbf{I}_p \end{cases}$$

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

**Proof** Let us consider the simple case where  $m = 2$ . In the first hand, we have for any  $(s, t) \in \mathcal{D}_1 \times \mathcal{D}_2$ ,

$$\text{Cov}(Y(s), Y(t)) = \mathbb{E}[Y(s)Y(t)] = k(|s - t|).$$

In the second one, we have

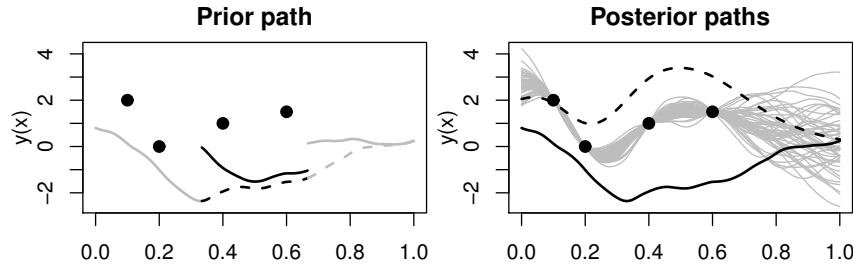
$$\begin{aligned} \mathbb{E}[\bar{Y}_1(s)\bar{Y}_2(t)] &= \sum_{i,j=1}^{+\infty} \sqrt{\gamma_i \gamma_j} \phi_i(s) \phi_j(t - S) \mathbb{E}[\bar{\zeta}_i^{(1)} \bar{\zeta}_j^{(2)}] \\ &= \sum_{i,j=1}^{+\infty} \phi_i(s) \phi_j(t - S) \int_{x=0}^S \int_{x'=S}^{2S} \phi_i(x) \phi_j(x' - S) k(|x - x'|) dx dx' \\ &= \int_0^S \int_S^{2S} \sum_{i=1}^{+\infty} \phi_i(s) \phi_i(x) \sum_{j=1}^{+\infty} \phi_j(t - S) \phi_j(x' - S) k(|x - x'|) dx dx' \\ &= \int_0^S \int_S^{2S} \delta(s - x) \delta(t - x') k(|x - x'|) dx dx' = k(|s - t|), \end{aligned}$$

where  $\delta(\cdot)$  is the Dirac delta function. Conversely, we have

$$\begin{aligned} \text{Cov} \left( \bar{\zeta}_i^{(1)}, \bar{\zeta}_j^{(2)} \right) &= \text{Cov} \left( \frac{1}{\sqrt{\gamma_i}} \int_0^S \phi_i(x) \bar{Y}_1(x) dx, \frac{1}{\sqrt{\gamma_j}} \int_S^{2S} \phi_j(t - S) \bar{Y}_2(t) dt \right) \\ &= \frac{1}{\sqrt{\gamma_i \gamma_j}} \int_{x=0}^S \int_{t=S}^{2S} k(|x - t|) \phi_i(x) \phi_j(t - S) dx dt \\ &= \frac{1}{\sqrt{\gamma_i \gamma_j}} \int_{x=0}^S \int_{x'=0}^S k(|x - x' - S|) \phi_i(x) \phi_j(x') dx dx'. \end{aligned}$$

The general case can be proved in a similar way. The proof of the second item of the proposition is obvious.

Let us give some comments on these results: From Eq. (5), the  $m^{\text{th}}$  conditional coefficient  $\bar{\zeta}^{(m)}$  is computed using the left hand side previous conditional coefficient  $\bar{\zeta}^{(m-1)}$  and the right hand side unconditional coefficient  $\zeta^{(m)}$ . The coupling matrix  $\mathbf{K}$  ensures that the distribution of the original GP ( $Y(x)$ ) is equal to that of the approximated process  $\bar{Y}_m$  at any two connected subdomains  $\mathcal{D}_{m-1} \cup \mathcal{D}_m$ . From the stationarity of the GP ( $Y(x)$ ), the matrix  $\mathbf{K}$  is computed once for any arbitrary number of subdomains. The approximation error between  $Y$  and  $\bar{Y}_m$  is studied in the following section (see, Proposition 4).



**Fig. 3** *Left*: GP sample paths prior where the domain is split in three subdomains. Solid curves (before) and dashed-curves (after) conditioning. *Right*: the posterior paths (gray solid curves) obtained using the MUR as in (2) by adding the prior (black solid curve) to the update (black dashed curve)

In Fig. 3, we illustrate the method presented in this section for sampling the prior (left panel) and we applied it to the MUR to get the target posterior distribution (right panel). The same parameters used in Fig. 1 are also used in this figure. The black dots represent the observations. In the left panel of Fig. 3, the domain  $\mathcal{D}$  is split in three subdomains, i.e.,  $M = 3$ . The solid curves represent the sample paths of the prior before conditioning which are uncorrelated (i.e.,  $Y_1$ ,  $Y_2$  and  $Y_3$ ). The dashed curves represent those after conditioning (i.e.,  $\bar{Y}_2$  and  $\bar{Y}_3$ ). Let us mention that the dashed curves follow the given correlation structure. In the right panel, the black solid curve represents the prior sample path obtained in the left panel. However, the black dashed curve represents the corresponding update sample path as in (2). As in Fig. 1, the update sample path follows the trend of the observations. The posterior sample paths (gray solid curves) have been obtained by applying the MUR to the prior as in (2). As expected, they verify the interpolation condition.

In the next section, the parallelization of the proposed approach is developed and the advantages are highlighted. Additionally, the mean-square global *block* error is computed.

### 4.3 Parallel computing large scale KLE

In this section, the parallelization of the technique presented in Sect. 4.2 developed in [13] is investigated. This section presents the main results of the paper, including the error incurred by the approach and the efficiency in special cases. Without loss of generality, suppose that the domain  $\mathcal{D}$  is split into  $M$  odd subdomains. The case when  $M$  is even is discussed at the end of this section, cf. Remark 1. Model (4) developed in the previous section is used again

$$\bar{Y}_m^p(x) = \sum_{i=1}^p \sqrt{\gamma_i} \phi_i(x - (m-1)S) \bar{\zeta}_i^{(m)}, \quad \forall x \in \mathcal{D}_m.$$

However, this section demonstrates how the sets of conditional KLE coefficients  $\bar{\zeta}^{(m)}$  can be generated using a parallelization technique. First, we sample independently  $M$  sets of KLE coefficients  $\zeta^{(1)}, \dots, \zeta^{(M)}$  following a standard MVN distribution. The KLE coefficients corresponding to an even part  $m$  is conditioned by the parts at the left and at the right

$$\bar{\zeta}^{(m)} = \mathbf{K}^\top \zeta^{(m-1)} + \mathbf{K} \zeta^{(m+1)} + \mathbf{H} \zeta^{(m)}, \quad (7)$$

where  $m \in \{2, 4, \dots, (M-1)\}$  and  $\mathbf{H}$  is the lower triangular matrix such that

$$\mathbf{I}_p - \mathbf{K}^\top \mathbf{K} - \mathbf{K} \mathbf{K}^\top = \mathbf{H} \mathbf{H}^\top. \quad (8)$$

From Eq. (7), one can deduce that the conditional coefficients sets  $\bar{\zeta}^{(m)}$  can be generated in parallel. To summarize: first, the prior KLE coefficients sets  $\zeta^{(1)}, \dots, \zeta^{(M)}$  are generated in parallel. Second, the  $M/2$  conditional coefficients  $\bar{\zeta}^{(m)}$  are computed in parallel too using Eq. (7). By this strategy, the conditional prior sample paths are generated by parallelization.

**Proposition 3** According to Eqs. (7) and (8),

- the two processes  $Y$  and  $(\bar{Y}_{m-1} \cup \bar{Y}_m)$  have the same distribution on  $\mathcal{D}_{m-1} \cup \mathcal{D}_m$ , for any  $m \in \{2, 4, \dots, (M-1)\}$ .
- Additionally, we have the following results:

$$\begin{aligned} \text{Cov}(\bar{\zeta}^{(m)}) &= \mathbf{I}_p, \quad \forall m \in \{2, 4, \dots, (M-1)\}; \\ \text{Cov}(\bar{\zeta}^{(m)}, \bar{\zeta}^{(m+2)}) &= \mathbf{K}^2 \quad \text{and} \quad \text{Cov}(\bar{\zeta}^{(m)}, \bar{\zeta}^{(m')}) = \mathbf{0}; \end{aligned}$$

for any  $m \in \{2, 4, \dots, (M-3)\}$  and any  $m' > m+2$  an even number.

**Proof** The proof of the first item is similar to the one given in Proposition 2, while the proof of the second item can be accomplished by a simple calculation.

Let us give some comments on these results:

- At any two connected subdomains, the proposed approach and the original random process have the same distribution.
- By the parallelization technique, the KLE coefficients corresponding to the odd parts are unconditioned. They are uncorrelated. This is because they are generated independently. Consequently, only  $(M-1)/2$  coefficients sets are conditioned. By this technique, the computational complexity of the sampling procedure is drastically reduced. However, as in Sect. 4.2, the eigendecomposition of the first subdomain is required to compute the coupling matrix  $\mathbf{K}$ .

- As we can see in Fig. 4, the domain  $\mathcal{D} = [0, 1]$  is split in  $M = 3$  subdomains. However, only one dashed conditioned part was used to follow the given correlation structure.
- The parallelization technique works well in the low correlation structure cases (i.e., when the correlation length parameter  $\theta$  is small enough). This is because the KLE sets corresponding to the odd part are uncorrelated by construction.

*Remark 1* When the number of subdomains  $M$  is even, only the KLE coefficients set corresponding to the last subdomain  $m = M$  is conditioned from the left-hand side (as in Sect. 4.2). In that case, the number of conditional KLE coefficients sets is equal to  $M/2$ .

In the following corollary, the correlation between blocks is computed. This leads to compare the approximated correlation obtained by the proposed model and the true correlation function.

**Corollary 2 (Correlation between blocks)** For any  $(x, x') \in \mathcal{D}_m \times \mathcal{D}_{m+1}$

$$\text{Cov}\left(\bar{Y}_m^p(x), \bar{Y}_{m+1}^p(x')\right) = \sum_{i,j=1}^p \sqrt{\gamma_i \gamma_j} \phi_i(x - (m-1)S) \phi_j(x' - mS) \mathbf{K}_{ij}, \quad (9)$$

for any  $m \in \{1, 2, \dots, M\}$ . However, for any  $m \in \{2, 4, \dots, M-3\}$  and any  $(x, x') \in \mathcal{D}_m \times \mathcal{D}_{m+2}$ , we have

$$\text{Cov}\left(\bar{Y}_m^p(x), \bar{Y}_{m+2}^p(x')\right) = \sum_{i,j=1}^p \sqrt{\gamma_i \gamma_j} \phi_i(x - (m-1)S) \phi_j(x' - (m+1)S) (\mathbf{K}^2)_{ij}.$$

The correlation is equal to zero in other situations, i.e., when the distance between blocks is greater than or equal to 3.

**Proof** The proof is a simple consequence of Proposition 3.

The following proposition computes the mean-square global *block* error, which will be used to compare the proposed approach with and without parallelization for different types of covariance functions.

**Proposition 4 (Mean-square global *block* error)**

In the setting of Proposition 3, we have the following global *block* error:

$$\begin{aligned} \epsilon_{B,M}^2 &= \frac{\mathbb{E}\left[\int_{\mathcal{D}} (Y(x) - \bar{Y}_{1:M}(x))^2 dG(x)\right]}{\mathbb{E}\left[\int_{\mathcal{D}} Y(x)^2 dG(x)\right]} \\ &= \frac{\text{Trace}\left((S_Y - S_{\bar{Y}_{1:M}})(S_Y - S_{\bar{Y}_{1:M}})^\top\right)}{\text{Trace}(S_Y S_Y^\top)}, \end{aligned} \quad (10)$$

where  $\bar{Y}_{1:M} = \cup_{m=1}^M \bar{Y}_m$  and  $(S_Y; S_{\bar{Y}_{1:M}})$  are the Cholesky matrices of the covariance functions of  $Y$  and  $\bar{Y}_{1:M}$  on the grid  $\mathcal{G} = \{x_1, \dots, x_N\}$  respectively and  $G$  is the cumulative distribution function (CDF) of the Uniform discrete random variable on  $\mathcal{G}$ .

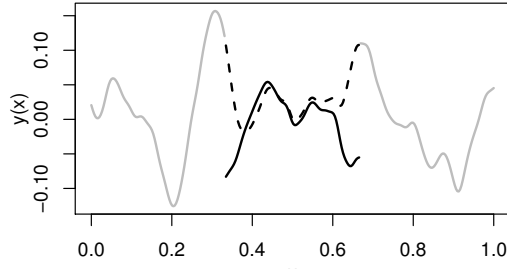
**Proof** We know from [11] that any zero-mean Gaussian vector can be written as

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

where  $\epsilon$  is a  $N$ -dimensional standard Gaussian vector chosen the same for  $Y$  and  $\bar{Y}_{1:M}$  to get a specific dependence structure. The two matrices  $S_Y$  and  $S_{\bar{Y}_{1:M}}$  are the Cholesky factorization of the covariance of  $Y$  and  $\bar{Y}_{1:M}$  on the grid  $\mathcal{G} = \{x_1, \dots, x_N\}$  respectively. If we denote by  $C$  the covariance matrix of the Gaussian vector  $[Y(x_1), \dots, Y(x_N)]^\top$ , then  $S_Y S_Y^\top = C$ . Thus,

$$\mathbb{E} \left[ \sum_{i=1}^N Y(x_i)^2 \right] = \mathbb{E}[\epsilon^\top S_Y^\top S_Y \epsilon] = \mathbb{E}[\epsilon^\top C \epsilon] = \text{Trace}(C) = \text{Trace}(S_Y S_Y^\top).$$

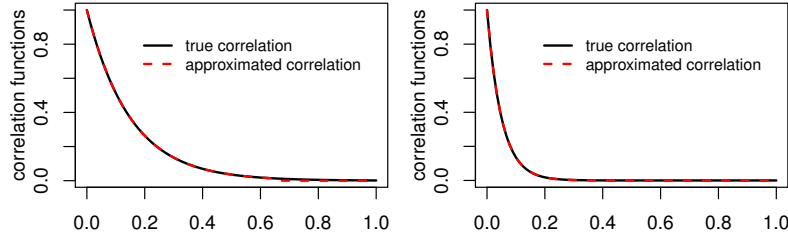
The result holds by following the same way.



**Fig. 4** GP sample path prior using the parallelization technique. The domain  $\mathcal{D}$  is split in three subdomains. The solid curves (resp. dashed curves) represent the paths before (resp. after) conditioning

Figure 4 shows one GP sample path prior using the Matérn covariance function with the regularity parameter  $\nu = 5/2$  by applying the parallelization technique described in this section. The correlation length parameter  $\theta$  is fixed at 0.05. The domain  $\mathcal{D}$  is split in three subdomains. However, only one conditional KLE sets (gray dashed curve) is needed to follow the correlation structure in the entire domain. The solid curves (resp. dashed curves) represent the GP sample paths before (resp. after)

conditioning. The black dashed curve follows the given correlation on the left and on the right hand sides as expected.



**Fig. 5** The correlation functions between  $t = 0$  and any  $t \in \mathcal{D}$  when the domain is split into three subdomains. The approximated correlation using the proposed approach is computed from Eq. (9). The true correlation is the Exponential function with  $\theta = 0.15$  (left) and  $0.05$  (right)

Figure 5 shows the correlation functions between  $t = 0$  and any  $t \in \mathcal{D}$  when the domain  $\mathcal{D}$  is split into three subdomains. The black solid curve represents the true Exponential covariance function (Table 1) with correlation length parameter  $\theta = 0.15$  (left panel) and  $\theta = 0.05$  (right panel). However, the red dashed curve represents the proposed correlation function obtained from Eq. (9). In the case where  $\theta = 0.15$ , the root-mean-square error (RMSE) between the true correlation function and the proposed one in the entire domain is of order  $1.68 \times 10^{-2}$ . When  $\theta = 0.05$ , the RMSE is even smaller, equal to  $1.24 \times 10^{-7}$ . This is an expected result as the correlation length parameter  $\theta$  was chosen to be ‘too small’ (right panel), leading to a rapidly decreasing correlation function. This is a suitable situation for the parallelization technique developed in this section, where the correlation between unconnected odd subdomains is zero by construction.

*Remark 2* The parallelization technique developed in this section can be applied to the family of covariance functions with compact support (cf. Sect. 4.2 in [16]). As said in [16], ‘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 which is suitable for the parallelization technique investigated in this section, since the correlation between unconnected odd parts is exactly zero, cf. Fig. 6 below. However, the approximation error still exists.

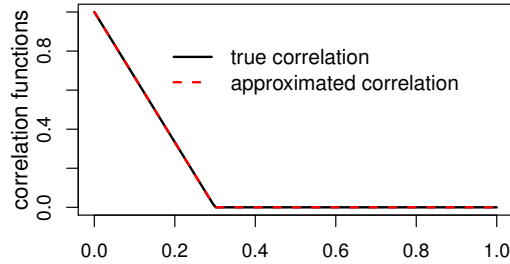
#### Example: compact support

In this example, the triangle correlation function (corresponding to the class of covariance functions with compact support) is used

$$k(|h|) = \max\left(1 - \frac{|h|}{\theta}; 0\right), \quad (11)$$



where  $\theta$  is the correlation length parameter. In Fig. 6,  $\theta$  is fixed at 0.3. This means that  $k(|h|)$  is equal to zero for any  $h \geq 0.3$ . As before, the domain  $\mathcal{D}$  is split into three subdomains. The black solid curve represents the true triangle correlation function (11) on  $\mathcal{D}$ . However, the red dashed curve represents the approximation correlation function using the proposed approach with the parallelization technique developed in this section. Let us mention that the RMSE on  $\mathcal{D}$  between these two functions is of order  $3.6 \times 10^{-16}$ .



**Fig. 6** The true correlation function and the approximated one on  $\mathcal{D}$  are shown together. The true one corresponding to the triangle correlation function in (11). The RMSE is equal to  $3.6 \times 10^{-16}$

**Table 2** The mean-square global *block* error for different type of covariance functions using the proposed approach with and without parallelization. The domain  $\mathcal{D}$  is split into  $M = 3$  subdomains

| covariance function         | Mean-square global <i>block</i> error    |  |
|-----------------------------|--|--|
|                             | without parallelization                  | with parallelization                     |
| Triangle, $\theta = 0.3$    | $3.82 \times 10^{-2}$                    | <b><math>1.79 \times 10^{-27}</math></b> |
| Triangle, $\theta = 0.05$   | $1.49 \times 10^{-5}$                    | <b><math>2.69 \times 10^{-28}</math></b> |
| Matérn 5/2, $\theta = 0.05$ | <b><math>2.99 \times 10^{-24}</math></b> | $3.95 \times 10^{-7}$                    |
| Matérn 3/2, $\theta = 0.05$ | <b><math>6.42 \times 10^{-27}</math></b> | $1.18 \times 10^{-7}$                    |

Table 2 shows the mean-square global *block* error defined in (10) for different type of covariance functions using the proposed approach with and without parallelization. The parallelization technique outperforms the case without parallelization when using a covariance function with compact support, according to the numerical experiments. This is because the correlation between odd subdomains is equal to zero by construction, unlike the classical approach without parallelization. However, for the Matérn family of covariance functions, the parallelization has no advantage over the proposed approach without parallelization in terms of mean-square global *block* error. For instance, with the Matérn covariance function and a regularity parameter of  $\nu = 5/2$ , the parallelization technique results in a mean-square global

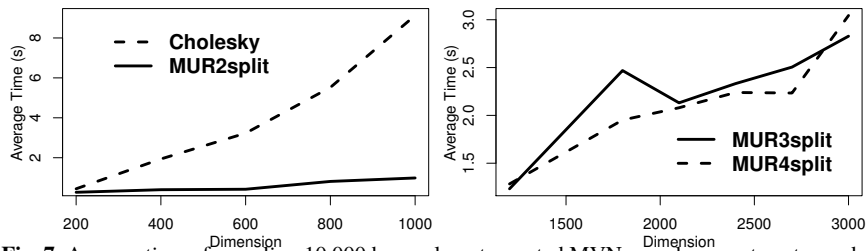
block error of  $9.27 \times 10^{-17}$  when the correlation length parameter is set to 0.03, and  $1.81 \times 10^{-28}$  when it is set to 0.01. So, it provides an accurate result. This is because, when the correlation length parameter  $\theta$  tends to zero, the Matérn covariance function decreases rapidly to zero. In that case, it can be considered to be almost similar to a correlation function with compact support.

## 5 Computational illustrations

In this section, the performance of the proposed approach is investigated. The problem of sampling an  $N$ -dimensional MVN vector  $\eta \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ , truncated on the intersection of  $n < N$  hyperplanes, is considered

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

where  $\boldsymbol{A} \in \mathbb{R}^{n \times N}$ ,  $\boldsymbol{y} \in \mathbb{R}^n$  and  $\text{rank}(\boldsymbol{A}) = n$ . This problem is called hyperplane-truncated MVN distribution [3, 11]. The unconditional covariance matrix  $\boldsymbol{\Gamma}$  is generated using the Matérn  $\nu = 5/2$  covariance function. In that case, the covariance matrix  $\boldsymbol{\Gamma}$  admits no special structure. The elements of  $\boldsymbol{y}$  and  $\boldsymbol{A}$  are generated from a standard normal distribution  $\mathcal{N}(0, 1)$ .



**Fig. 7** Average time of sampling 10,000 hyperplane-truncated MVN samples over twenty random trials, when the number of data dimension  $N$  increases and of observations is fixed at  $n = 10$

In Fig. 7, the computation time of sampling 10,000 hyperplane-truncated MVN distributions averaged over twenty random trials is shown. The number of observations is fixed at  $n = 10$  and the data dimension (i.e., dimension of  $\eta$ )  $N$  increases. *Left panel:* the Cholesky factorization was compared to the proposed approach when the domain is split in only two subdomains. It is evident that the proposed approach outperforms the Cholesky factorization method. Contrarily to the Cholesky factorization, the computation time of the proposed approach grows linearly with the dimension of the vector  $\eta$ . *Right panel:* the black solid curve (resp. dashed curve) represents the average time in second of sampling 10,000 hyperplane-truncated MVN distribution using the proposed approach when the domain is split in three subdo-

mains (resp. four subdomains). The average time is slightly different between these two approaches.

## 6 Conclusion

In this paper, a new methodology for sampling large scale Gaussian process regression is developed. This problem is related to the simulation of high-dimensional Gaussian vectors truncated on the intersection of hyperplanes. The main idea is to combine both Matheron's update rule (MUR) and Karhunen Loève expansion (KLE). First, by the MUR we sample the target distribution without computing the posterior covariance matrix. Second, by splitting the input domain into smallest nonoverlapping subdomains, the KLE coefficients are conditioned in order to follow the correlation structure in the entire domain. The parallelization of this approach has been developed. The mean-square global *block* error has been computed as well. The advantages of the proposed approach are demonstrated through numerical examples. Based on numerical experiments, the parallelization technique is particularly efficient when using a class of covariance functions with compact support.

## Acknowledgements

The authors would like to thank the editor and the anonymous reviewers for their constructive comments that improved the clarity of the paper. 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. M. Chataigner. *Some contributions of machine learning to quantitative finance : volatility, nowcasting, cva compression*. Thesis, Université Paris-Saclay, October 2021.
2. H. Cho, D. Venturi, and G. E. Karniadakis. Karhunen–Loève expansion for multi-correlated stochastic processes. *Probabilistic Engineering Mechanics*, 34:157–167, 2013.
3. Y. Cong, B. Chen, and M. Zhou. Fast simulation of hyperplane-truncated multivariate normal distributions. *Bayesian Analysis*, 12(4):1017 – 1037, 2017.
4. A. Cousin, H. Maatouk, and D. Rullière. Kriging of financial term-structures. *European Journal of Operational Research*, 255(2):631–648, 2016.
5. X. Emery. Simple and ordinary multigaussian kriging for estimating recoverable reserves. *Mathematical Geology*, 37(3):295–319, 2005.
6. G. Golub and C. F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 1996.
7. D. Gueye. *Some contributions to financial risk management*. Thesis, Université de Strasbourg, July 2021.

8. Y. Hoffman and E. Ribak. Constrained realizations of Gaussian fields: A simple algorithm. *The Astrophysical Journal*, 380:L5, October 1991.
9. A. G. Journel and C. J. Huijbregts. *Mining geostatistics*. Academic Press, 1976.
10. M. Loève. Elementary probability theory. In *Probability theory i*, pages 1–52. Springer, 1977.
11. H. Maatouk, X. Bay, and D. Rullière. A note on simulating hyperplane-truncated multivariate normal distributions. *Statistics & Probability Letters*, 191:109650, 2022.
12. H. Maatouk, D. Rullière, and X. Bay. Sampling large hyperplane-truncated multivariate normal distributions. working paper or preprint, August 2022.
13. 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.
14. J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.
15. L. Wang. *Karhunen-Loève expansions and their applications*. London School of Economics and Political Science (United Kingdom), 2008.
16. C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
17. 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.