Large-scale constrained Gaussian processes for shape-restricted function estimation

Hassan Maatouk, Didier Rullière, Xavier Bay

Large-scale constrained Gaussian processes for shape-restricted function estimation

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

[1]EBInnov, School of Industrial Biology, 49 avenue des Genottes, Cergy-Pontoise, CS 90009, 95895, France.

[2*]Department of Mathematics, CY Tech, Site du Parc, Cergy-Pontoise, 95011, France.

[3]Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, Saint-Étienne, F-42023, France.

*Corresponding author(s). E-mail(s): h.maatouk@hubebi.com;
Contributing authors: drulliere@emse.fr; bay@emse.fr;

Abstract

In this paper, we revisit the problem of Bayesian shape-restricted function estimation. The finite-dimensional Gaussian process (GP) approximation proposed in Maatouk and Bay (2017) is considered, which admits an equivalent formulation of the shape constraints in terms of basis coefficients. This approximation satisfies a wide variety of shape constraints everywhere, whether applied alone, in combination, or sequentially. We propose a new, efficient, and fast algorithm for sampling from a large Gaussian vector extracted from a stationary GP. The proposed approach significantly improves the novel circulant embedding technique proposed in Ray et al (2020) for efficiently sampling from the resulting posterior constrained distribution. The main idea of the algorithm developed in the present paper is to divide the input domain into smaller subdomains and apply a cross-correlated technique to address the correlation structure in the entire domain. As the number of subdomains increases, the computational complexity is drastically reduced. The developed algorithm is accurate and efficient, as demonstrated through comparisons with competing approaches. The performance of the proposed approach has been evaluated within the context of shape-restricted function estimation.

Keywords: Elliptical slice sampling, nonparametric regression, shape constraints, smooth relaxation, Toeplitz

1 Introduction

Various real-world scenarios have been presented in fields such as nuclear physics (Zhou et al, 2019) and econometrics (Chataigner et al, 2021; Cousin et al, 2022, 2016; Crépey and Dixon, 2020), in which the data indicate that the underlying function satisfies specific shape constraints such as monotonicity, convexity and boundedness.

The utilization of a Bayesian framework provides a unified probabilistic approach to integrating different shape constraints, leading to a rich literature devoted to Bayesian shape constrained estimation. A common approach is to express the unknown function as a basis

expansion and convert the functional constraints into linear constraints in the space of coefficients. Examples of such basis expansions include the piecewise linear model (Cai and Dunson, 2007; Neelon and Dunson, 2004), Bernstein polynomials (Curtis and Ghosh, 2011), splines (Shively et al, 2011), and restricted splines (Meyer et al, 2011; Papp and Alizadeh, 2014). Gaussian processes (GPs) are a well-known nonparametric Bayesian framework for estimating functions (Williams and Rasmussen, 2006), and they have also been applied for shape-restricted inference (Tran et al, 2023). For example, the authors in Riihimäki and Vehtari (2010) impose the monotonicity constraint by including derivative information in a GP model. In Lin and Dunson (2014), the idea is to project the GP prior into the space of constraints. The authors in Pensoneault et al (2020) enforce nonnegativity constraints into GP models. They select a set of constraint points in the domain and impose the nonnegativity on the posterior GP at these points. Recently, an overview and survey detailing various strategies to incorporate shape constraints into a GP are provided in Swiler et al (2020).

In the present paper, we focus on the finite-dimensional GP approximation originally proposed in Maatouk and Bay (2017). Its performance has been demonstrated through several real-world data applications (Cousin et al, 2022, 2016; López-Lopera et al, 2018; Maradesa et al, 2022; Williams et al, 2023; Zhou et al, 2019). Through this approach (Maatouk and Bay, 2017), the generalization of the well-known Kimeldorf-Wahba correspondence (Kimeldorf and Wahba, 1970) between Bayesian estimation on stochastic processes and splines for constrained cases has been established (Bay et al, 2016; Grammont et al, 2022). The main idea in Maatouk and Bay (2017) is to approximate the samples of the GP prior by representing them in a finite-dimensional space of functions using an appropriate basis expansion. The advantage of this approach is that various functional shape constraints can be translated *equivalently* into linear inequality constraints on the basis coefficients. This leads to the need for sampling from a high-dimensional truncated multivariate normal (tMVN) distribution.

Sampling from a tMVN distribution is a challenging problem, especially in higher dimensions. Recently, several efficient McMC algorithms have been proposed for sampling from a tMVN distribution. These included Gibbs sampling (Taylor and Benjamini, 2016), Metropolis-Hastings (MH) (Murphy, 2018), and the minimax tilting method accept-reject sampler (Botev, 2017). Moreover, an efficient Hamiltonian Monte Carlo (HMC) algorithm was proposed in Pakman and Paninski (2014), which significantly improved the speed of sampling from tMVNs.

In the present paper, we focus on the strategy proposed in Ray et al (2020) for sampling from a tMVN distribution restricted to the positive orthant. Their main idea is based on a novel combination of elliptical slice sampling (denoted as ESS; (Murray et al, 2010)), circulant embedding techniques, and *smooth relaxation* of hard constraints. They also use Durbin's recursion to efficiently update hyperparameters within the covariance function of the parent GP. Therefore, sampling from the full conditional tMVN distribution is performed by sampling from a prior MVN distribution. In other words, they sample before conditioning rather than after. To accomplish this, they utilize the highly efficient sampler, based on the fast Fourier transform (FFT) for stationary GPs on a regular grid (Wood and Chan, 1994), denoted as samp.WC. In the present paper, we propose a new, efficient, and fast algorithm for sampling from a large MVN distribution extracted from a stationary one-dimensional GP on a regular grid. With the developed algorithm, generating a MVN distribution of size 1,000,000 takes around 0.25 second on a machine with a 8-Core processor and 8 GB RAM. We demonstrate its suitability and performance for the strategy outlined in Ray et al (2020); Zhou et al (2022) within the context of shape-restricted function estimation.

The article is structured as follows: in Section 2, we present the proposed methodology for generating samples from a large MVN distribution. We include a comparison with competing approaches. Section 3 is devoted to the application of shape-restricted function estimation. Firstly, we provide a brief recall of the efficient approach developed in Ray et al (2020). Secondly, we evaluate the performance of the method proposed in this paper in terms of computational complexity and prediction accuracy using the approach established in Ray et al (2020). Third, we explore efficient samplers to approximate both the posterior and prior distributions, including Hamiltonian Monte Carlo and the Fast Fourier Transform. Fourth, we investigate the

behavior of both the maximum and mean a posterior estimate (MAP) and (mAP) respectively in terms of prediction accuracy.

2 Prior sampling

Consider the problem of sampling from a prior distribution having the following form:

$$p(\boldsymbol{\xi}) \propto \exp\left\{-\frac{1}{2\tau^2}\boldsymbol{\xi}^\top \boldsymbol{K}^{-1}\boldsymbol{\xi} + \frac{1}{\tau^2}\boldsymbol{\xi}^\top \boldsymbol{K}^{-1}\boldsymbol{\mu}\right\}, \quad \boldsymbol{\xi} \in \mathbb{R}^N, \tag{1}$$

where $\boldsymbol{K}$ is a positive definite matrix, $\tau > 0$, and $\boldsymbol{\mu} \in \mathbb{R}^N$. Although our methodology is generally applicable to any such $\boldsymbol{K}$, we are specifically interested in situations where $\boldsymbol{K}$ results from evaluating a stationary covariance function on a regular grid. This means that $\boldsymbol{K}_{j,l} = k(u_j - u_l)$ for a positive definite function k and a set of regular grid points $\{u_j\}$, $j = 1, \ldots, N$ in $\mathbb{R}$. Some commonly used stationary covariance functions can be found in Williams and Rasmussen (2006). Throughout this paper, and without loss of generality, the Matérn family of covariance functions will be employed with a smoothness parameter $\nu > 0$ and a length-scale parameter $\ell > 0$.

The prior distribution (1) is widely used in Bayesian nonparametric regression problems, especially when employing techniques such as Metropolis-Hastings (MH) (Cotter et al, 2013; Neal, 1999), ESS (Murray et al, 2010), or Matheron's update rule (MUR) (Cong et al, 2017; Maatouk et al, 2023b; Wilson et al, 2021). Standard approaches such as eigendecomposition and Cholesky factorization involve computing a *scaling* matrix $\boldsymbol{S}$ such that $\boldsymbol{S}\boldsymbol{S}^\top = \boldsymbol{K}$. However, when N is high (i.e., $N \gg 1,000$), these approaches become numerically heavy, with a computational complexity of order $\mathcal{O}(N^3)$, see Golub and Van Loan (1996) Chapter 4.2.3. In the next section, we will present the proposed algorithm to overcome this issue.

2.1 Algorithm development

In this section, we present our methodology of sampling from a prior distribution as in (1). Hereafter, we shall assume that $\boldsymbol{K}_{j,l} = k(u_j - u_l)$ for a positive definite function k and a regular grid point $\{u_j\}$, $j = 1, \ldots, N$. Let $(Y(x))_{x\in\mathcal{D}}$ be a zero-mean GP with covariance function k, i.e., $Y \sim \mathcal{GP}(0, k)$.

Suppose that $\mathcal{D} = [0, 1]$ is discretized into $M \times N_1 = N$ equally spaced points $\{u_j\}$, for some $M \geq 1$. We consider the case where $\mathcal{D}$ is partitioned into $M \geq 2$ equally sized subdomains $\mathcal{D}_m$, i.e., $\mathcal{D} = \bigcup_{m=1}^{M} \mathcal{D}_m$. Then each subdomain contains N_1 equally spaced points, where N_1 is smaller than N. The algorithm developed in this section is also applicable for non-uniform subdivision of $\mathcal{D}$. In this paper, our focus is on the scenario where the Gaussian prior vector $\boldsymbol{\xi}$ is determined by the value of the GP prior Y on the grid $\{u_j\}$, i.e., $\boldsymbol{\xi}_j = Y(u_j)$, for any $j = 1, \ldots, N$. The main idea is to generate M independent samples $\{\boldsymbol{\xi}^{(m)}\}$, $m = 1, \ldots, M$, each covering the associated subdomain, and then impose a correlation between the random coefficients of the connected subdomains, to align with the given correlation structure. Let us mention that since the random samples are generated independently, the associated GPs $(Y_m(x))_{x\in\mathcal{D}_m}$ and $(Y_{m'}(x))_{x\in\mathcal{D}_{m'}}$ are uncorrelated for any $m \neq m'$, where, $\boldsymbol{\xi}_j^{(m')} = Y_{m'}(u_j)$, for some $j's$ in $\{1, \ldots, N\}$. The random samples $\{\boldsymbol{\xi}^{(m)}\}$ follow a MVN distribution, $\mathcal{N}_{N_1}(\boldsymbol{0}, \boldsymbol{K}_{11})$, where $\boldsymbol{K}_{11} = (k(u_j - u_l))_{j,l}$, for $j, l = 1, \ldots, N_1$, with $\boldsymbol{0}$ being the N_1-dimensional zero vector.

Proposition 1 (Construction of conditional coefficients)**.** *Suppose that* $\{\boldsymbol{\xi}^{(m)}\}$, $m = 1, \ldots, M$ *are independent samples from a* $\mathcal{N}_{N_1}(\boldsymbol{0}, \boldsymbol{K}_{11})$ *distribution, and* $\boldsymbol{K}_{12} := (\mathrm{Cov}(\xi_j, \xi_l))_{j,l} = (k(u_j - u_l))_{j,l}$*, where* $j \in \{1, \ldots, N_1\}$ *and* $l \in \{N_1 + 1, \ldots, 2N_1\}$*. Let* $\left\{\widetilde{\boldsymbol{\xi}}^{(m)}\right\}$ *be a sequence of conditional random vectors defined as follows:*

$$\begin{cases} \widetilde{\boldsymbol{\xi}}^{(1)} := \boldsymbol{\xi}^{(1)}; \\ \widetilde{\boldsymbol{\xi}}^{(m)} := \boldsymbol{C}\widetilde{\boldsymbol{\xi}}^{(m-1)} + \boldsymbol{L}\boldsymbol{\xi}^{(m)}, \quad m = 2, \ldots, M \end{cases} \tag{2}$$

where $\boldsymbol{C} := \boldsymbol{K}_{12}\boldsymbol{K}_{11}^{-1}$ *and* $\boldsymbol{L} := \underline{\boldsymbol{L}}(\overline{\boldsymbol{L}})^{-1}$*, with*

$$\boldsymbol{K}_{11} - \boldsymbol{K}_{12}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}^{\top} = \underline{\boldsymbol{L}}(\underline{\boldsymbol{L}})^{\top} \quad \textit{and} \quad \boldsymbol{K}_{11} = \overline{\boldsymbol{L}}(\overline{\boldsymbol{L}})^{\top}.$$

Then,

$$\begin{cases} \mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m)}\right) = \boldsymbol{K}_{11}, & \textit{for any } m = 1, \ldots, M; \\ \mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m-1)}, \widetilde{\boldsymbol{\xi}}^{(m)}\right) = \boldsymbol{K}_{12}, & \textit{for any } m = 2, \ldots, M. \end{cases}$$

More generally, for any $m < m'$*, we have* $\mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m)}, \widetilde{\boldsymbol{\xi}}^{(m')}\right) = \boldsymbol{K}_{12}(\boldsymbol{C}^{\top})^{m'-m-1}$*, where* $(\boldsymbol{C}^{\top})^0 = \mathbf{I}$*, the identity matrix.*

Before presenting the proof of Proposition 1, let us recall that $\boldsymbol{K}_{11} - \boldsymbol{K}_{12}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}^{\top}$ is a symmetric and positive definite matrix that represents the covariance matrix of the conditional MVN distribution $\{Y(\boldsymbol{u}_1)|Y(\boldsymbol{u}_2)\}$, where $Y(\boldsymbol{u}_1) = [Y(u_1), \ldots, Y(u_{N_1})]^{\top} \in \mathbb{R}^{N_1}$ and $Y(\boldsymbol{u}_2) = [Y(u_{N_1+1}), \ldots, Y(u_{2N_1})]^{\top} \in \mathbb{R}^{N_1}$.

Proof of Proposition 1. Let us first proof that $\mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m)}\right) = \boldsymbol{K}_{11}$, for any $m = 1, \ldots, M$. For example, when $m = 2$, we have

$$\begin{aligned} \mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(2)}\right) &= \mathrm{Cov}\left(\boldsymbol{C}\boldsymbol{\xi}^{(1)} + \boldsymbol{L}\boldsymbol{\xi}^{(2)}\right) = \boldsymbol{C}\boldsymbol{K}_{11}\boldsymbol{C}^{\top} + \boldsymbol{L}\boldsymbol{K}_{11}\boldsymbol{L}^{\top} \\ &= \boldsymbol{K}_{12}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}^{\top} + \underline{\boldsymbol{L}}(\underline{\boldsymbol{L}})^{\top} = \boldsymbol{K}_{11}. \end{aligned}$$

Now, by induction, for any $m > 2$, we have

$$\begin{aligned} \mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m)}\right) &= \mathrm{Cov}\left(\boldsymbol{C}\widetilde{\boldsymbol{\xi}}^{(m-1)} + \boldsymbol{L}\boldsymbol{\xi}^{(m)}\right) = \boldsymbol{C}\mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(m-1)}\right)\boldsymbol{C}^{\top} + \boldsymbol{L}\mathrm{Cov}\left(\boldsymbol{\xi}^{(m)}\right)\boldsymbol{L}^{\top} \\ &= \boldsymbol{C}\boldsymbol{K}_{11}\boldsymbol{C}^{\top} + \boldsymbol{L}\boldsymbol{K}_{11}\boldsymbol{L}^{\top} = \boldsymbol{K}_{11}. \end{aligned}$$

Additionally, for $m = 2$, we have

$$\mathrm{Cov}\left(\widetilde{\boldsymbol{\xi}}^{(1)}, \widetilde{\boldsymbol{\xi}}^{(2)}\right) = \mathrm{Cov}\left(\boldsymbol{\xi}^{(1)}, \boldsymbol{C}\boldsymbol{\xi}^{(1)} + \boldsymbol{L}\boldsymbol{\xi}^{(2)}\right) = \boldsymbol{K}_{11}\boldsymbol{C}^{\top} = \boldsymbol{K}_{12}.$$

Using induction, we can prove the general formula, thereby completing the proof of the proposition. □

Algorithm 1 Sampling scheme of $\boldsymbol{\xi} \sim \mathcal{N}_N(\boldsymbol{\mu}, \boldsymbol{K})$, where N is too high.

Initialization: $\boldsymbol{\mu}$, $\boldsymbol{K}$, M, N_1, and $N = M \times N_1$.

- Generate $\boldsymbol{\xi} \sim \mathcal{N}_N(\mathbf{0}, \boldsymbol{K})$:
  - compute $\boldsymbol{K}_{11}$ and $\boldsymbol{K}_{12}$;
  - sample independently $\boldsymbol{\xi}^{(m)} \sim \mathcal{N}_{N_1}(\mathbf{0}, \boldsymbol{K}_{11})$, for any $m = 1, \ldots, M$;
  - compute the matrices $\boldsymbol{C}$ and $\boldsymbol{L}$;
  - compute $\widetilde{\boldsymbol{\xi}}^{(m)}$ using Equation (2), for any $m = 1, \ldots, M$;
  - bind $\boldsymbol{\xi} = \left[\widetilde{\boldsymbol{\xi}}^{(1)}, \ldots, \widetilde{\boldsymbol{\xi}}^{(M)}\right]^{\top}$.
- Return $\boldsymbol{\mu} + \boldsymbol{\xi}$.

Algorithm 1 outlines the different steps for simulating a large MVN distribution $\boldsymbol{\xi} \sim \mathcal{N}_N(\boldsymbol{\mu}, \boldsymbol{K})$ (i.e., when $N \gg 1,000$) using the proposed approach when the domain $\mathcal{D}$ is

divided into M equally sized subdomains. Generally speaking, Algorithm 1 is denoted as *Fast Large-Scale* and abbreviated as Fast.LS throughout the paper.

2.2 Numerical performance

The aim of this section is to illustrate the proposed approach and demonstrate its efficiency by comparing it with competing approaches. To accomplish this, we will utilize the Matérn family of covariance functions with a smoothness parameter $\nu > 0$ and a length-scale parameter $\ell > 0$ (Williams and Rasmussen, 2006). Generally speaking, the length-scale parameter ℓ in the numerical examples was chosen such that the correlation at the maximum separation between covariates equals 0.05, unless otherwise specified.

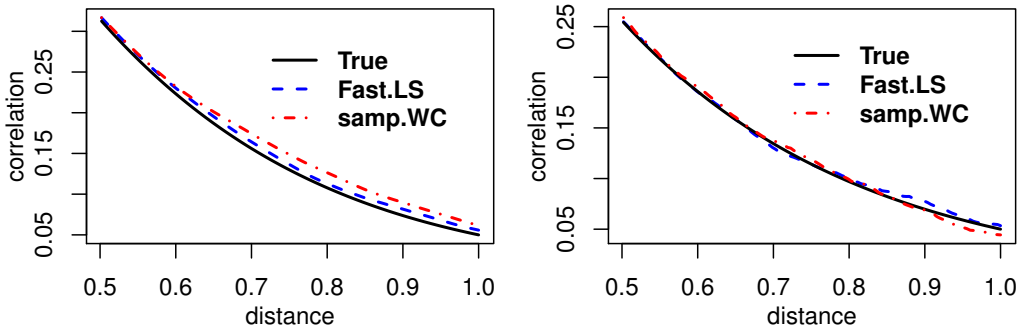

Fig. 1 The Matérn covariance function is plotted as a function of distance. We used 15,000 iterations repeated 25 times to compute the estimation of the covariance function for the two compared approaches: Fast.LS and samp.WC. In the left panel, ν is fixed at 1.5, with an average mean squared error (MSE) of 5.82×10^{-3} for Fast.LS and 27.30×10^{-3} for samp.WC. In the right panel, ν is set to 0.75, and the average MSE is 1.53×10^{-3} for Fast.LS, while it is 1.89×10^{-3} for samp.WC.

In Fig. 1, we compare the accuracy of the proposed approach Fast.LS with the fast Fourier transform (FFT), denoted as samp.WC (Wood and Chan, 1994). The Matérn covariance function with a smoothness parameter ν is plotted as a function of the distance $h \in [0.5, 1]$ (black slide curves). In the left panel, ν is fixed at 1.5, while in the right panel, it is fixed at 0.75. The blue dashed curve and the red dashed-dotted curve represent the numerical estimation of the Matérn covariance function based on 15,000 iterations repeated 25 times using the proposed Fast.LS and samp.WC methods, respectively. The mean square error (MSE) for both approaches is provided in the caption of Fig. 1. It is worth noting that the accuracy of both approaches decreases as the smoothness parameter ν increases. Let us mention that the MVN generated in this numerical example has a dimension of $N = 250$. When using the proposed approach, the number of subdomains M is fixed at 5, and the size of the first subdomain N_1 is set to 50.

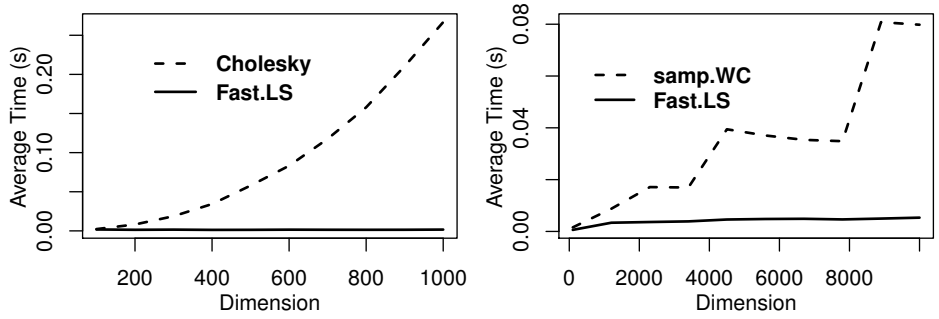

Fig. 2 Average running time of sampling a MVN over 25 replicates as a function of the dimension N. The proposed approach, Fast.LS, has been compared to Cholesky factorization in the left panel and to FFT samp.WC approach (Wood and Chan, 1994) in the right panel.

Figure 2 displays the average running time (in seconds) of sampling from a MVN distribution over 25 replicates as a function of the dimension N. The proposed approach exhibits a clear advantage over both Cholesky factorization and the FFT, as the dimension N of the MVN increases. Note that in our proposed approach, the value of N_1 is fixed at 50 in the left panel and 100 in the right panel, while the number of subdomains M varies from 2 to 20 in left panel and from 1 to 100 in right panel.

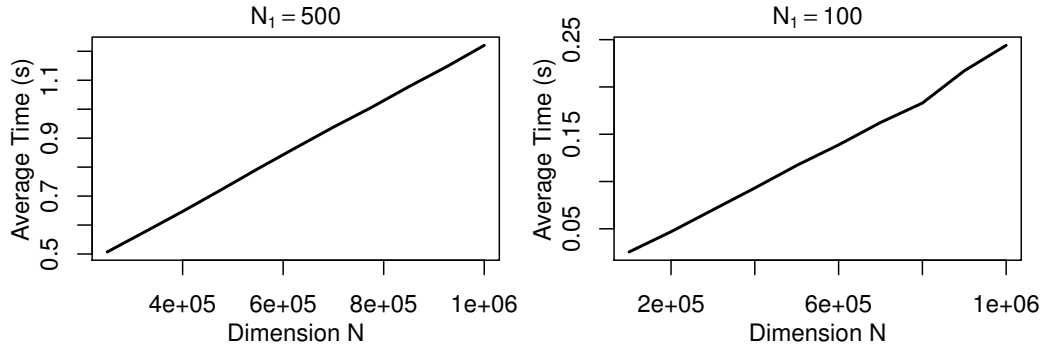

Fig. 3 Average running time (in seconds) of sampling a MVN over 25 replicates as a function of the dimension using the proposed approach Fast.LS. In the left panel, we fix the value of N_1 at 500, and the number of subdomains, M, varies from 500 to 2,000. In the right panel, we fix the value of N_1 at 100, and we vary the number of subdomains, M, from 1,000 to 10,000.

Figure 3 displays the average sampling times in seconds of a MVN over 25 replicates as a function of the dimension when using the proposed approach Fast.LS. The Matérn covariance function with a smoothness parameter of $\nu = 0.5$ was used. In the left panel, the first subdomain is discretized into $N_1 = 500$ equally spaced points, and the input domain $\mathcal{D}$ is split into M subdomains, varying from 500 to 2,000. While in the right panel, we fix the value of N_1 at 100, and we vary the number of subdomains, M, from 1,000 to 10,000. Using the algorithm developed in this paper, a Gaussian vector of dimension $N = 1,000,000$ can be generated in approximately 0.25 second. Let us mention that the running time is linear as a function of the dimension N.

Remark 1 (Problem in sample.WC for smoothness cases)**.** *The function sample.WC, which utilizes the FFT method described in Wood and Chan (1994), performs well for small smoothness parameter ν (i.e., $\nu < 2$). However, it encounters a problem when a higher value of ν is employed. For example, when the smoothness parameter ν and the length-scale parameter ℓ of the Matérn covariance function are fixed at 2.5 and 0.5, respectively, sample.WC fails to generate a Gaussian vector of dimension greater than 1,000. On the other hand, the proposed Fast.LS algorithm does not suffer from this issue. Nevertheless, when smoothness is higher, a nugget effect should be added to avoid numerical problems.*

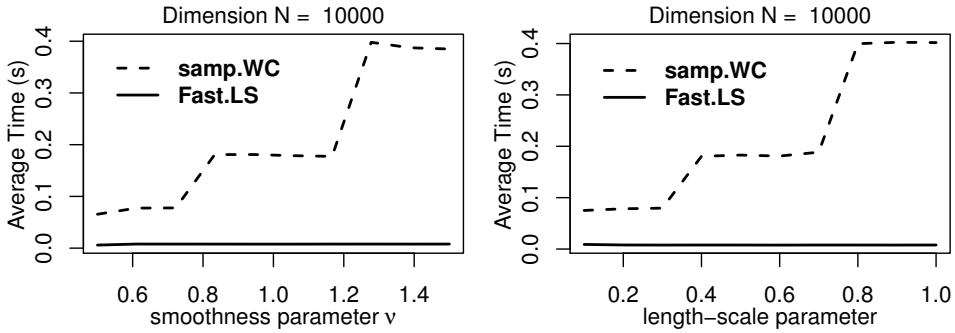

Fig. 4 Average running time (in seconds) of sampling a MVN over 50 replicates as a function of the Matérn smoothness parameter ν (left) and length-scale parameter ℓ (right), using the proposed approach Fast.LS (black solid curve) and samp.WC (black dashed curve). The dimension N is fixed at 10,000.

In Fig. 4, we examine the behavior of the proposed approach Fast.LS and the FFT samp.WC as the Matérn parameters ν and ℓ increase. We computed the average running time (in seconds) of sampling a MVN over 50 replicates as a function of ν (left panel) and ℓ (right panel). For the proposed approach, the average running time remains stable across all values of ν and ℓ, whereas for FFT samp.WC, it increases. Note that, in the proposed Fast.LS approach, the parameters N_1 and M are both fixed at 100.

We conclude this section with a comparison of computational running times using the recently developed efficient large-scale Karhunen-Loève approach, abbreviated as LS.KLE, and presented in Maatouk et al (2023b).

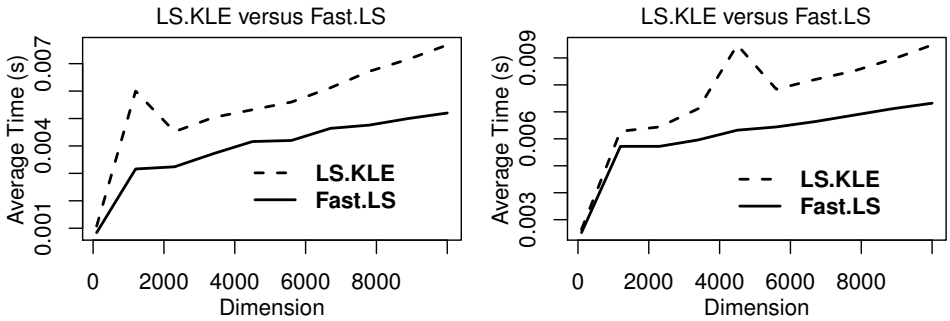

Fig. 5 Average running time of sampling a MVN over 25 replicates as a function of the dimension N. The proposed approach, Fast.LS, has been compared to the recently efficient LS.KLE approach developed in Maatouk et al (2023b). The Matérn covariance function is employed with a smoothness parameter $\nu = 0.5$ in the left panel and $\nu = 1.5$ in the right panel.

Figure 5 depicts the average running time for sampling a MVN distribution over 25 replicates, plotted against the dimension N. The Matérn covariance function was employed with a smoothness parameter $\nu = 0.5$ in the left panel and $\nu = 1.5$ in the right panel. The proposed Fast.LS approach is compared to the recently efficient LS.KLE approach developed in Maatouk et al (2023b). In both cases, the Fast.LS approach presented in this paper demonstrates faster performance compared to LS.KLE.

2.3 Non stationary cases

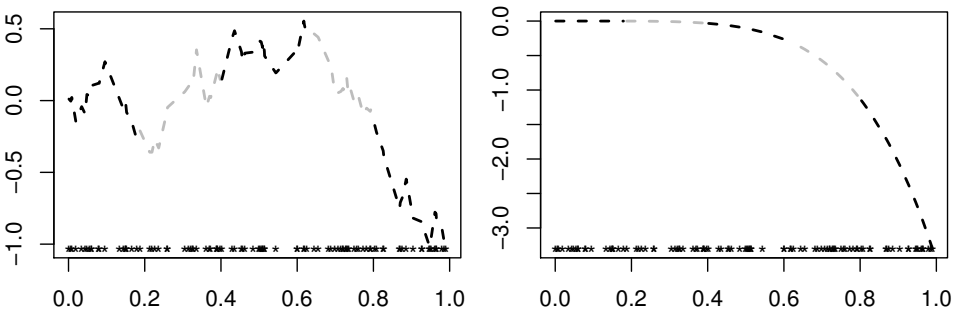

Fig. 6 A GP sample path was generated using the proposed approach with $M = 5$ and $N_1 = 20$. The Brownian motion covariance function was used in the left panel, while the polynomial covariance function was used in the right panel.

In Fig. 6, a GP sample path is generated using the proposed approach with $M = 5$ and $N_1 = 20$. The left panel utilizes the Brownian motion covariance function $k(x, x') = \min(x, x')$. On the other hand, the right panel employs the polynomial covariance function from Williams and Rasmussen (2006), defined as $k(x, x') = (xx' + \sigma_0^2)^p$, where $\sigma_0 = 0.1$ and $p = 5$. A non-uniform grid was used for the input domain $\mathcal{D} = [0, 1]$ in both cases. The locations were displayed using black crosses.

3 Application to shape-restricted function estimation

In this section, we explore the application of the algorithm developed in this paper to shape-restricted function estimation through GP regression, where the unknown function satisfies shape constraints such as nonnegativity, monotonicity, or convexity. Let $\{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, n\}$ be a training set of n observations, where $\boldsymbol{x}_i$ denotes the covariates and y_i represents the data. The following regression problem is considered

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \tag{3}$$

$i = 1, \ldots, n$, where f represents an unknown latent function that generates the data $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$. Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a covariate of dimension d, and ϵ_i is an additive independent and identically distributed (i.i.d.) zero-mean Gaussian noise with a constant variance of σ^2. We assume a GP prior distribution on the unknown function f (Williams and Rasmussen, 2006).

3.1 Finite-dimensional Gaussian process approximation

We first introduce some notations to define the finite-dimension GP approximation proposed in Maatouk and Bay (2017), which will be used in the present paper. For example, in the one-dimensional case (i.e., $d = 1$), we denote by $\{u_j,\ j = 1, \ldots, N\}$ the set of equally spaced knots on $\mathcal{D} = [0, 1]$ with spacing $\delta_N = 1/(N-1)$, where $u_j = (j-1)\delta_N$. Note that the methodology developed in the present paper is also applicable to non-uniform discretization of $\mathcal{D}$. We recall the three basis functions that will be used in three different models denoted as (M_h), (M_ϕ), and (M_φ):

$$h_j(x) := h\left(\frac{x - u_j}{\Delta_N}\right), \quad \phi_j(x) := \int_0^x h_j(t)\mathrm{d}t, \quad \varphi_j(x) := \int_0^x \int_0^t h_j(u)\mathrm{d}u\mathrm{d}t, \tag{4}$$

for $j \in \{1, \ldots, N\}$, where $h(x) := (1 - |x|)\mathbf{1}_{[-1,1]}(x)$ is the *hat* function on $[-1, 1]$. Let us recall that the *hat* functions $\{h_j\}$ have two nice properties. First, the value of any *hat* function at any knot is equal to Kronecker's delta function (i.e., $h_j(u_l) = \delta_{j,l}$), where $\delta_{j,l}$ is equal to one when $j = l$ and zero otherwise. Second, for any $x \in \mathcal{D}$, we have $\sum_{j=1}^N h_j(x) = 1$. As outlined in Maatouk and Bay (2017), any continuous function $f : \mathcal{D} \to \mathbb{R}$, that is, $f \in C^0(\mathcal{D}, \mathbb{R})$ can be approximated by a piecewise linear interpolating between the function values at the knots $\{u_j\}$,

$$\tilde{f}_N(x) = \sum_{j=1}^N f(u_j)h_j(x), \quad \forall x \in \mathcal{D}, \tag{5}$$

where $\{h_j\}$ are the basis functions defined in (4). Additionally, $\tilde{f}_N$ converges uniformly to f when N tends to infinity.

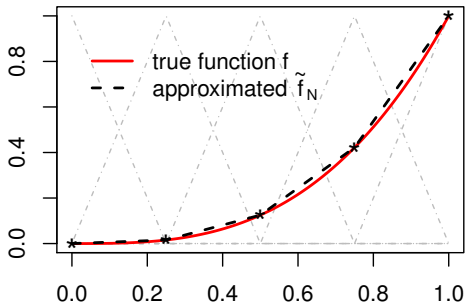

Fig. 7 Illustrative example of approximating a monotone (nondecreasing) function f (red solid curve) using a piecewise linear interpolating function $\tilde{f}_N$ (black dashed curve). A uniform subdivision is used with $N = 5$ hat functions represented by the gray triangles.

Figure 7 presents an illustrative example of approximating the monotone (nondecreasing) function $f(x) = x^3$ (red solid curve) with the piecewise linear interpolating function $\tilde{f}_N$ defined in (5), where N is fixed at 5. The gray triangles represent the five basis functions $\{h_j\}$, while the black crosses represent the values of $\tilde{f}_N$ at the knot points $\{u_j\}$.

Now, if the function f is continuous differentiable or twice continuously differentiable, then by the fundamental theorem of calculus, we get

$$f(x) - f(0) = \int_0^x f'(t)\mathrm{d}t;$$
$$f(x) - f(0) - xf'(0) = \int_0^x \left(\int_0^t f''(v)\mathrm{d}v \right) \mathrm{d}t.$$

Expending f' and f'' in the basis approximation (5) implies the following two approaches:

$$\tilde{f}_N(x) = f(0) + \sum_{j=1}^N f'(u_j)\phi_j(x) \quad \text{and} \quad \tilde{f}_N(x) = f(0) + f'(0)x + \sum_{j=1}^N f''(u_j)\varphi_j(x),$$

for any $x \in \mathcal{D}$. Now, we are ready to present the three different models:

$$Y^N(x) := \sum_{j=1}^N Y(u_j)h_j(x) = \sum_{j=1}^N \xi_j h_j(x), \tag{M_h}$$

$$Y^N(x) := Y(0) + \sum_{j=1}^N Y'(u_j)\phi_j(x) = \xi_0 + \sum_{j=1}^N \xi_j \phi_j(x), \tag{M_ϕ}$$

$$Y^N(x) := Y(0) + Y'(0)x + \sum_{j=1}^N Y''(u_j)\varphi_j(x) = \xi_0 + \xi_0^* x + \sum_{j=1}^N \xi_j \varphi_j(x), \tag{M_φ}$$

for any $x \in \mathcal{D}$, where ξ_j is equal to $Y(u_j)$ in (M_h), $Y'(u_j)$ in (M_ϕ), and $Y''(u_j)$ in (M_φ), for any $j = 1, \ldots, N$. Since $(Y(x))_{x\in\mathcal{D}}$ is supposed a zero-mean GP with covariance function k, then from (M_h), the vector of coefficient $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^\top$ is zero-mean Gaussian with covariance matrix $\boldsymbol{K}$ such that

$$\boldsymbol{K}_{j,l} = k(u_j - u_l), \quad \forall j, l = 1, \ldots, N.$$

The covariance matrix $\boldsymbol{K}$ preserves the stationary property and exhibits a Toeplitz structure. Before presenting the following equivalent results in Proposition 2, let us introduce some notations to simplify the presentation of the problem. Let $\mathcal{C}$ be the convex set of functions that verify some shape constraints, such as monotonicity, convexity, and boundedness. The non-convex case has been investigated in Maatouk et al (2023a). For instance,

$$\mathcal{C} = \begin{cases} \mathcal{C}_b := \left\{ f \in C^0(\mathcal{D}, \mathbb{R}) \text{ s.t. } f(x) \geq 0, \ \forall x \in \mathcal{D} \right\} \\ \mathcal{C}_m := \left\{ f \in C^1(\mathcal{D}, \mathbb{R}) \text{ s.t. } f'(x) \geq 0, \ \forall x \in \mathcal{D} \right\} \\ \mathcal{C}_c := \left\{ f \in C^2(\mathcal{D}, \mathbb{R}) \text{ s.t. } f''(x) \geq 0, \ \forall x \in \mathcal{D} \right\} \end{cases} \tag{6}$$

which corresponds to boundedness (nonnegativity), monotonicity (nondecreasing), and convexity constraints, respectively. Here, $C^0(\mathcal{D}, \mathbb{R})$, $C^1(\mathcal{D}, \mathbb{R})$, and $C^2(\mathcal{D}, \mathbb{R})$ represent the sets of continuous, continuously differentiable, and twice continuously differentiable functions from $\mathcal{D}$ to $\mathbb{R}$, respectively. The authors in Maatouk and Bay (2017) have shown the advantages of using the basis functions defined in (4). Indeed, they demonstrated an equivalent between the functional constraints $Y^N \in \mathcal{C}$ and a finite set of linear constraints on the basis coefficients $\{\xi_j\}$ for various shape constraints. To be more precise, for the functional constraints given in (6), we have

$$Y^N \in \mathcal{C} \quad \Leftrightarrow \quad \boldsymbol{\xi} \in \mathcal{E}, \tag{7}$$

where $\mathcal{E} = \{\boldsymbol{\xi} \in \mathbb{R}^N \ : \ \xi_j \geq 0, \ j = 1, \ldots, N\}$ representing the positive orthant in $\mathbb{R}^N$. Let us now recall the following results provided in Maatouk and Bay (2017):

Proposition 2 (Equivalent constraints)**.**

- *Boundedness constraints: If Y^N is defined as in (M_h), then $Y^N \in \mathcal{C}_b$ if and only if $\boldsymbol{\xi} \in \mathcal{E}$.*
- *Monotonicity constraints: If Y^N is defined as in (M_ϕ), then $Y^N \in \mathcal{C}_m$ if and only if $\boldsymbol{\xi} \in \mathcal{E}$.*
- *Convexity constraints: If Y^N is defined as in (M_φ), then $Y^N \in \mathcal{C}_c$ if and only if $\boldsymbol{\xi} \in \mathcal{E}$.*

Here, $\mathcal{C}_b$, $\mathcal{C}_m$, and $\mathcal{C}_c$ represent the sets of functional constraints provided in (6), and $\mathcal{E}$ denotes the positive orthant in $\mathbb{R}^N$.

Hereafter, we focus on the boundedness (nonnegativity) constraints using model (M_h). The other shape constraints can be handled similarly. The Model (M_h), considering both noisy observations and shape constraints, is expressed as follows:

$$Y^N(x) = \sum_{j=1}^{N} \xi_j h_j(x) \quad \text{s.t.} \quad \begin{cases} Y^N(x_i) + \epsilon_i = y_i & \text{(noisy observations)}, \\ Y^N \in \mathcal{C}_b & \text{(boundedness constraints)}, \end{cases} \tag{8}$$

where $x_i \in \mathcal{D}$ is the covariate, $y_i \in \mathbb{R}$ is the response and $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, with σ^2 the noise variance. Following the equivalent in (7), the conditional distribution (8) can be written in a matrix form as follows:

$$\mathbf{H}\boldsymbol{\xi} + \boldsymbol{\epsilon} = \boldsymbol{y}, \quad \boldsymbol{\xi} \in \mathcal{E}, \tag{9}$$

where $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$ is the vector of data, $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_n]^\top$ is the noise Gaussian vector, $\mathbf{H}$ is the $n \times N$ design matrix defined by $\mathbf{H}_{i,j} := h_j(x_i)$, and $\mathcal{E}$ is the positive orthant of $\mathbb{R}^N$. With only noisy observations $\{\mathbf{H}\boldsymbol{\xi} + \boldsymbol{\epsilon} = \boldsymbol{y}\}$, we have $\{\boldsymbol{\xi}|\boldsymbol{y}\}$ follows a MVN distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (see, for example, Williams and Rasmussen (2006), Section 2.1.1), where

$$\begin{cases} \boldsymbol{\mu} = (\mathbf{H}^\top \mathbf{H}/\sigma^2 + \boldsymbol{K}^{-1}/\tau^2)^{-1} \mathbf{H}^\top \boldsymbol{y}/\sigma^2; \\ \boldsymbol{\Sigma} = (\mathbf{H}^\top \mathbf{H}/\sigma^2 + \boldsymbol{K}^{-1}/\tau^2)^{-1}. \end{cases} \tag{10}$$

A tMVN prior is placed on $\boldsymbol{\xi}$ as $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \tau^2 \boldsymbol{K})$ restricted to the positive orthant $\mathcal{E}$ of $\mathbb{R}^N$, where $\boldsymbol{K}$ is a positive definite matrix defined as $\boldsymbol{K}_{j,l} = (k(u_j - u_l))$, with $k(\cdot)$ representing the stationary Matérn covariance function with a smoothness parameter $\nu > 0$ and length-scale parameter $\ell > 0$. In this paper, we adopt the efficient MCMC sampler developed in Ray et al (2020) to sample from the joint posterior $(\boldsymbol{\xi}, \sigma^2, \tau^2, \nu, \ell)$. Our contribution lies in the sampling of the high-dimensional Gaussian prior $\boldsymbol{\xi}$ restricted to (9) using the efficient circulant embedding technique introduced in Ray et al (2020), along with Algorithm 1 proposed in Section 2.1. In the next section, we provide a brief overview of this technique.

3.2 Mean and Maximum a posteriori estimates (mAP) and (MAP)

Before presenting the new circulant embedding technique developed in Ray et al (2020), let us first define the two posterior estimates, namely maximum a posterior (MAP) and mean a posterior (mAP). We denote $\boldsymbol{\mu}^*$ as the mode of the posterior distribution (9). It maximizes the following posterior probability density function (pdf):

$$\boldsymbol{\mu}^* := \underset{\boldsymbol{\xi} \text{ s.t. } \boldsymbol{\xi} \in \mathcal{E}}{\arg\max} \left\{ -[\boldsymbol{\xi} - \boldsymbol{\mu}]^\top \boldsymbol{\Sigma}^{-1} [\boldsymbol{\xi} - \boldsymbol{\mu}] \right\}, \tag{11}$$

where $\mathcal{E}$ is the positive orthant of $\mathbb{R}^N$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the conditional mean and covariance, respectively, as given in (10). This is a quadratic optimization problem subject to linear inequality constraints (Boyd and Vandenberghe, 2004; Goldfarb and Idnani, 1983). It is equivalent to

$$\boldsymbol{\mu}^* := \underset{\boldsymbol{\xi} \text{ s.t. } \boldsymbol{\xi} \in \mathcal{E}}{\arg\min} \left\{ \boldsymbol{\xi}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} - 2\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} \right\}. \tag{12}$$

Now, we are ready to define the following two posterior estimates:

Definition 1 (MAP estimate)**.** *The Maximum a posteriori (MAP) estimate of* Y^N *conditionally on shape constraints and noisy observations is defined as*

$$M^N(x) := \sum_{j=1}^{N} \mu_j^* h_j(x) = \boldsymbol{h}(x)^\top \boldsymbol{\mu}^*, \quad x \in \mathcal{D},$$

where $\boldsymbol{\mu}^* = [\mu_1^*, \ldots, \mu_N^*]^\top \in \mathbb{R}^N$ *is the posterior mode computed by* (12) *and* $\boldsymbol{h}(x) = [h_1(x), \ldots, h_N(x)]^\top$.

The MAP estimate is independent of the sampling process and is determined only by solving a quadratic optimization problem with linear inequality constraints (see Equation (12)). However, it does rely on the noise standard deviation parameter σ. To address this, we propose using the average of the estimated values of σ obtained from the joint posterior presented in Ray et al (2020). The asymptotic behavior of this estimate has been discussed in both Bay et al (2016) and Grammont et al (2022), where they established a generalization of the Kimeldorf-Wahba correspondence (Kimeldorf and Wahba, 1970) for constrained cases.

Definition 2 (mAP estimate)**.** *The mean a posteriori (mAP) estimate of* Y^N *conditionally on shape constraints and noisy observations is defined as*

$$m^N(x) := \mathrm{E}\left[Y^N(x) \middle| \boldsymbol{y}, Y^N \in \mathcal{C}_b\right] = \boldsymbol{h}(x)^\top \overline{\boldsymbol{\mu}}, \quad x \in \mathcal{D},$$

where $\overline{\boldsymbol{\mu}} := \mathrm{E}\left[\boldsymbol{\xi} | \boldsymbol{y}, \boldsymbol{\xi} \in \mathcal{E}\right]$ *is the posterior mean which is computed from simulations and* $\boldsymbol{h}(x) = [h_1(x), \ldots, h_N(x)]^\top$.

3.3 Constrained posterior sampling

The main results outlined in the present paper are motivated by a recent novel approach proposed in Ray et al (2020). This approach is based on an efficient circulant embedding technique for simulating a MVN distribution restricted to the positive orthant, which avoids the use of the full conditional tMVN (14). In this section, we provide a brief overview of this technique for simulating from a constrained posterior distribution with the following form:

$$p(\boldsymbol{\xi}) \propto \underbrace{\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{\xi}\|^2\right)}_{\text{likelihood function}} \underbrace{\exp\left(-\frac{1}{2\tau^2}\boldsymbol{\xi}^\top \boldsymbol{K}^{-1}\boldsymbol{\xi}\right)}_{\text{(untruncated) MVN prior}} \underbrace{\mathbf{1}_{\mathcal{E}}(\boldsymbol{\xi})}_{\text{constraints}}, \quad \boldsymbol{\xi} \in \mathbb{R}^N, \tag{13}$$

where $\boldsymbol{y} \in \mathbb{R}^n$, $\mathbf{H} \in \mathbb{R}^{n \times N}$, $\mathcal{E}$ is the positive orthant in $\mathbb{R}^N$, and $\boldsymbol{K}$ is a positive definite matrix. This posterior pdf in (13) is proportional to the product of a likelihood function, an (untruncated) MVN prior, and an indicator function representing the set of constraints. The methodology generally applies to any such $\boldsymbol{K}$. However, our specific interest lies in situations where $\boldsymbol{K}$ is derived from the evaluation of a stationary covariance function on a regular grid. Equation (13) represents the pdf of the posterior constrained distribution (9). A simple calculation yields that $p(\boldsymbol{\xi})$ in (13) is the density function of the following MVN distribution (see e.g., Williams and Rasmussen (2006), Section 2.1.1):

$$\mathcal{N}\left(\left(\frac{\mathbf{H}^\top\mathbf{H}}{\sigma^2} + \frac{\boldsymbol{K}^{-1}}{\tau^2}\right)^{-1} \frac{\mathbf{H}^\top \boldsymbol{y}}{\sigma^2}, \left(\frac{\mathbf{H}^\top\mathbf{H}}{\sigma^2} + \frac{\boldsymbol{K}^{-1}}{\tau^2}\right)^{-1}\right) \tag{14}$$

truncated to the positive orthant $\mathcal{E}$ in $\mathbb{R}^N$. Sampling from the above tMVN distribution can be achieved using the fast HMC approach developed in Pakman and Paninski (2014) and implemented in the `R package` *tmg*. However, in situations when N is large, $(\mathbf{H}^\top\mathbf{H}/\sigma^2 + \boldsymbol{K}^{-1}/\tau^2)$ keeps changing over each McMC iteration due to updates in σ and τ. This necessitates an $N \times N$ matrix inversion at every iteration. Moreover, within a large McMC algorithm, updating the unknown covariance function parameters involves computing the inversion of an $N \times N$ matrix at each step.

The novel approach described in Ray et al (2020) for sampling from the density in (13) can be summarized in three steps that avoid matrix inversions and direct sampling from the full conditional posterior tMVN distribution in (14).

First step: smooth relaxation of the constraint

In the first step, the authors approximate the indicator function in Equation (13) by employing a smooth approximant through the sigmoid function. Specifically, they use the logistic sigmoid function $1/(1+e^{-x})$, which is the cumulative distribution function (cdf) of the logistic distribution. In other words, they use scaled logistic sigmoid approximations $\mathbf{1}_{(0,+\infty)}(x) \approx (1+e^{-\eta x})^{-1}$ for large $\eta > 0$ to derive a smooth approximation called $\mathbb{J}_\eta(\cdot)$ for $\mathbf{1}_{\mathcal{E}}(\boldsymbol{\xi})$ as follows:

$$\mathbf{1}_{\mathcal{E}}(\boldsymbol{\xi}) \approx \mathbb{J}_\eta(\boldsymbol{\xi}) = \prod_{j=1}^{N} \frac{1}{1+e^{-\eta \xi_j}}, \tag{15}$$

where $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]^\top \in \mathbb{R}^N$. By substituting the indicator function $\mathbf{1}_{\mathcal{E}}(\cdot)$ with its approximation $\mathbb{J}_\eta(\cdot)$ in (13), we obtain the following density approximation of p, where the smooth relaxation of the constraints $\mathbb{J}_\eta$ has been absorbed into the likelihood:

$$\tilde{p}(\boldsymbol{\xi}) \propto \underbrace{\left[\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y}-\mathbf{H}\boldsymbol{\xi}\|^2\right)\prod_{j=1}^{N}\frac{1}{1+e^{-\eta\xi_j}}\right]}_{\textit{redefined likelihood}\text{ function}} \underbrace{\exp\left(-\frac{1}{2\tau^2}\boldsymbol{\xi}^\top \boldsymbol{K}^{-1}\boldsymbol{\xi}\right)}_{\text{(untruncated) MVN prior}}. \tag{16}$$

Let us recall that the parameter η controls the quality of the approximation, which increases with the value of η. The approximate posterior pdf $\tilde{p}(\boldsymbol{\xi})$ in (16) is proportional to the product of a *redefined likelihood* function (the quantity within the square brackets) and an (untruncated) zero-mean MVN prior $\boldsymbol{\xi}$.

Second step: circulant embedding techniques

In the second step, as described in Ray et al (2020), the authors utilize the efficient ESS method developed in Murray et al (2010) to sample from the approximate posterior distribution in (16). Let us give a brief overview of the ESS. The ESS is a general technique employed for sampling from distributions with the following form:

$$p(\boldsymbol{\xi}) \propto L(\boldsymbol{\xi})\mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \boldsymbol{K}). \tag{17}$$

The above density (17) is proportional to a product of a general likelihood function $L(\cdot)$ and a zero-mean MVN prior, where L represents the *redefined likelihood* function in (16). In this context, sampling from (17) can be performed using Metropolis-Hastings proposals:

$$\boldsymbol{\xi}' = \rho\boldsymbol{\nu} + \sqrt{1-\rho^2}\boldsymbol{\xi}, \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}), \tag{18}$$

where $\rho \in [-1, 1]$ is a step-size parameter, $\boldsymbol{\xi}$ is the current state, and $\boldsymbol{\xi}'$ is the proposal state. These proposals are known to possess good empirical properties (Neal, 1999), as well as theoretical properties (Cotter et al, 2013). This proposal maintains the zero-mean MVN prior $\mathcal{N}(\mathbf{0}, \boldsymbol{K})$. In other words, if $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$ and $\boldsymbol{\xi}'|\boldsymbol{\xi} \sim \mathcal{N}(\sqrt{1-\rho^2}\boldsymbol{\xi}, \rho^2\boldsymbol{K})$ is drawn as described above, then the marginal distribution of $\boldsymbol{\xi}'$ remains $\mathcal{N}(\mathbf{0}, \boldsymbol{K})$. Using this fact, it can be observed that the Metropolis-Hastings acceptance ratio $\alpha = \min\left\{1, L(\boldsymbol{\xi}')/L(\boldsymbol{\xi})\right\}$ only relies on the likelihood ratio and is independent of ρ. The main idea is to generate u uniformly on $[0, 1]$ and then accept the proposal state $\boldsymbol{\xi}'$ in (18) when $u < \frac{L(\boldsymbol{\xi}')}{L(\boldsymbol{\xi})}$. This is equivalent to

$$\log(L(\boldsymbol{\xi})) + \log u < \log(L(\boldsymbol{\xi}')).$$

Let us recall that this method is simple to implement and can be immediately applied to a much wider variety of models with Gaussian priors. As reported in Neal (1999), the Metropolis-Hastings proposals performed better than Gibbs sampling for GP classification.

The ESS offers an adaptive and automated approach for tuning the step-size parameter ρ in (18), ensuring acceptance at each step. This technique is founded on the parametrization $\rho = \sin(\theta)$:

$$\boldsymbol{\xi}' = \sin(\theta)\boldsymbol{\nu} + \cos(\theta)\boldsymbol{\xi}, \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}), \tag{19}$$

where the angle θ is uniformly generated from a $[\theta_{\min}, \theta_{\max}]$ interval which is shrunk exponentially fast until an acceptable state is reached. For each respective θ, a uniform random number is generated and compared against the likelihood ratio $L(\boldsymbol{\xi}')/L(\boldsymbol{\xi})$. In the event that the proposal $\boldsymbol{\xi}'$ is not acceptable, one shrinks the bracket of θ, and continues this process until acceptance. The authors in Murray et al (2010) provide detailed explanations on how to shrink the bracket.

Third step: efficient sampling from the prior MVN distribution

In the final step, as described in Ray et al (2020), the authors employed a highly efficient sampler based on the fast Fourier transform (FFT) developed in Wood and Chan (1994) to sample $\boldsymbol{\nu}$ in (19) from the prior MVN distribution $\mathcal{N}(\mathbf{0}, \tau^2\boldsymbol{K})$. The last step will not be covered in detail in this paragraph since it will be replaced by the proposed algorithm developed in Section 2.1.

Since the covariance matrix $\boldsymbol{K}$ is obtained from a stationary covariance function on a regular grid, sampling $\boldsymbol{\nu}$ from the prior MVN distribution $\mathcal{N}(\mathbf{0}, \tau^2\boldsymbol{K})$ is equivalent to generating a stationary GP in a regular grid. For this reason, the authors in Ray et al (2020) utilize the efficient FFT algorithm developed in Wood and Chan (1994). The efficiency of this algorithm depends on the dimension N of the MVN prior and on both the smoothness parameter ν and the length-scale parameter ℓ of the covariance function k, as demonstrated in Section 2.2 (refer to Figures 2 and 4). Our contribution in this paper is to apply the proposed Fast.LS algorithm developed in Section 2.1 (i.e., Algorithm 1) to the methodology described in this section, enabling us to efficiently sample from the posterior approximate distribution (16).

Algorithm 2 details the different steps for sampling from the approximate posterior pdf in (16) using the proposed approach known as large-scale ESS, abbreviated as LS-ESS. It is worth mentioning that the posterior mode $\boldsymbol{\mu}^*$ in (12) can be used as a starting point in the ESS step. From Equation (16), the logarithm of the *redefined likelihood* is

$$\log(L(\boldsymbol{\xi})) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{\xi}\|^2 - \sum_{j=1}^{N}\left[\log\left(1 + e^{-\eta\xi_j}\right)\right]. \tag{20}$$

In that case, the computational complexity of computing the log-likelihood in (20) is $\mathcal{O}(nN)$, where n is the number of samples and N is the dimension of the MVN prior $\boldsymbol{\xi}$. Similar to the MUR technique (Maatouk et al, 2023b; Wilson et al, 2021), the proposed approach involves sampling before conditioning rather than after. This approach can be advantageous since it preserves the stationary property during the sampling procedure. Nevertheless, sampling $\boldsymbol{\nu}$ in (19) from the prior MVN distribution $\mathcal{N}(\mathbf{0}, \tau^2\boldsymbol{K})$ generally has a computational complexity of order $\mathcal{O}(N^3)$ (Golub and Van Loan, 1996). This complexity corresponds to only one McMC sampler iteration. Through Algorithm 1, this complexity is reduced to $\mathcal{O}(N_1^3)$, where $N_1 \ll N$. Consequently, the computational complexity of Algorithm 2 becomes $\mathcal{O}(\min\{nN, N_1^3\})$, with N_1 representing the size of the first split subdomain (see Section 2.1).

In the next section, we empirically demonstrate the computational complexity and prediction accuracy of the proposed LS-ESS approach when applied to shape-restricted function estimation. Furthermore, we compare it with the strategies outlined in Ray et al (2020); Zhou et al (2022).

Algorithm 2 Sampling scheme from the approximate posterior pdf in (16) using the proposed LS-ESS approach for N_c McMC iterations.

Input: current state $\boldsymbol{\xi} = \boldsymbol{\mu}^*$ in (12), N, τ, $\boldsymbol{K}$, $\boldsymbol{y}$, $\mathbf{H}$, η, and N_c McMC iterations.

- For i from 1 to N_c, **do**
  - Generate $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \tau^2 \boldsymbol{K})$ using Algorithm 1.
  - Compute the new state $\boldsymbol{\xi}'_{(i)}$ in (19) using the ESS in Murray et al (2010):
    1. Compute the log-likelihood functions:

$$\log(L(\boldsymbol{\xi})) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{\xi}\|^2 + \eta \sum_{j=1}^{N} \xi_j - \sum_{j=1}^{N} \left[\log\left(1 + e^{\eta \xi_j}\right)\right];$$
$$\log y = \log(L(\boldsymbol{\xi})) + \log u, \quad u \sim \mathcal{U}([0,1]).$$

    2. Define an initial bracket $[\theta_{\min}, \theta_{\max}]$:

$$\theta \sim \mathcal{U}([0, 2\pi]);$$
$$\theta_{\min} = \theta - 2\pi \quad \text{and} \quad \theta_{\max} = \theta.$$

    3. Compute the proposal

$$\boldsymbol{\xi}'_{(i)} = \sin(\theta)\boldsymbol{\nu} + \cos(\theta)\boldsymbol{\xi}.$$

    4. **While** $\log(L(\boldsymbol{\xi}'_{(i)})) \leq \log y$;
       - shrink the bracket $[\theta_{\min}, \theta_{\max}]$ as in Murray et al (2010) and generate $\theta \sim \mathcal{U}([\theta_{\min}, \theta_{\max}])$;
       - compute:

$$\boldsymbol{\xi}'_{(i)} = \sin(\theta)\boldsymbol{\nu} + \cos(\theta)\boldsymbol{\xi}.$$

  - Reinitialization: $\boldsymbol{\xi} = \boldsymbol{\xi}'_{(i)}$
- **EndFor**

Output: return $\left[\boldsymbol{\xi}'_{(1)}, \ldots, \boldsymbol{\xi}'_{(N_c)}\right]^\top$.

3.4 Run-time comparison

The purpose of this section is to demonstrate the computational complexity of the proposed approach when applied to shape-restricted function estimation. A comparison is conducted with the efficient circulant embedding technique developed in Ray et al (2020); Zhou et al (2022). As explained in Section 3.3, the circulant embedding technique proposed in Ray et al (2020) for simulating a tMVN relies on the FFT technique developed in Wood and Chan (1994) to sample from the (untruncated) prior distribution. However, our proposed approach LS-ESS replaces the step of sampling the (untruncated) prior with Algorithm 1 outlined in Section 2.1.

3.4.1 Monotone function estimation

In this section, we consider the monotone (nondecreasing) function $f(x) = 3/[1 + \exp(-10x + 2.1)]$, with $x \in [0, 1]$, as discussed in Maatouk et al (2023a); Zhou et al (2022). This function is nondecreasing and exhibits approximate flatness between 0.7 and 1. We will use it as a basis to evaluate the performance of the proposed approach in terms of computational running time and prediction accuracy.

Figure 8 illustrates the monotone (nondecreasing) function estimation using the GP approximation from Model (M_ϕ) and the circulant embedding technique presented in Section 3.3. We fix $N = 100$ and randomly generate $n = 100$ samples from (3) using the true nondecreasing function f (black solid curve) and a noise standard deviation of $\sigma = 0.5$. The covariates $\{x_i\}$, $i = 1, \ldots, n$ are uniformly generated between 0 and 1. This dataset is then randomly divided

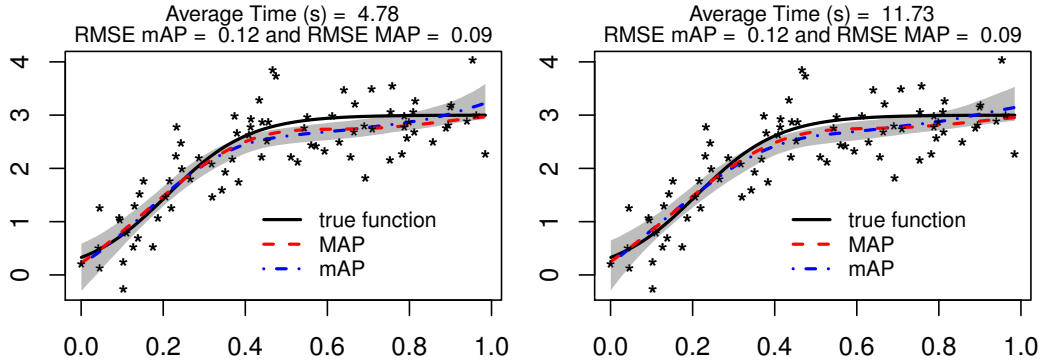

Fig. 8 Monotone function estimation using the efficient circulant embedding technique. The proposed approach (Fast.LS) is used in the left panel, while the FFT method (sample.WC) is employed in the right panel. The average computing time (in seconds) over 25 replicates of running the McMC sampler for 6,000 iterations is displayed in the main part of each panel. The dimension of the prior is fixed at $N = 100$.

into a training set of size 80 and a testing set of size 20. In Fig. 8, the training samples are represented by the black crosses. For the covariance function, we use the Matérn function with a smoothness parameter of $\nu = 1.5$. The length-scale parameter ℓ is chosen such that the correlation at the maximum possible separation between the covariates equals 0.05. The parameters σ and τ are updated at each McMC step using the inverse gamma distribution. The blue dashed-dotted curve represents the mean of the posterior sample paths (i.e., mAP estimate in Definition 2), while the red dashed curve represents the maximum of the posterior sample paths (i.e., MAP estimate in Definition 1). The black solid curve represents the true nondecreasing function, and the gray shaded area represents the posterior 95% confidence interval. The Fast.LS method is used to generate the *working prior* in the left panel with $N_1 = 20$ and $M = 5$, while the FFT sample.WC approach is employed in the right panel. The average computing time (in seconds) over 25 replicates of running the McMC sampler for 6,000 iterations is displayed in the main panel of each part, where the first 1,000 iterations are discarded as burn-in. In that case, and under the same setting, our approach using Fast.LS is approximately three times faster than the approach using the FFT sample.WC. Furthermore, the average root-mean-square error (RMSE) over 25 replicates for both mAP and MAP estimates is displayed in the main of each panel. According to the Bayesian analysis developed in Maatouk et al (2023a), the MAP estimate outperforms the mAP estimate in terms of prediction accuracy.

3.4.2 Bounded function estimation

In this section, we use the nonnegative function introduced in Pensoneault et al (2020) to evaluate the performance of the proposed approach in terms of computational complexity and prediction accuracy. The function is defined as follows:

$$f(x) = \frac{1}{[1 + (10x)^4]} + \frac{1}{2}\exp\left[-100(x - 0.5)^2\right], \quad x \in [0, 1]. \tag{21}$$

This is a challenging situation because the underlying function f is nonnegative, flat, and approximately zero on the interval $[0.7, 1]$.

Figure 9 illustrates the nonnegative function estimation using the GP approximation from Model (M_h). We fix $N = 100$ and randomly generate $n = 100$ samples from (3) using the true nonnegative function f defined in (21) and a noise standard deviation of $\sigma = 0.1$. This dataset is then randomly divided into an 80% training set and a 20% testing set. As in the numerical example of Section 3.4.1, the Fast.LS method is used to generate the *working prior* in the left panel with $N_1 = 20$ and $M = 5$, while the FFT sample.WC approach is employed in the right panel. The average computing time (in seconds) over 25 replicates of running the McMC sampler for 6,000 iterations is displayed in the main panel of each part, where the first 1,000 iterations are discarded as burn-in. Again, the proposed approach is approximately three times faster than the FFT approach.

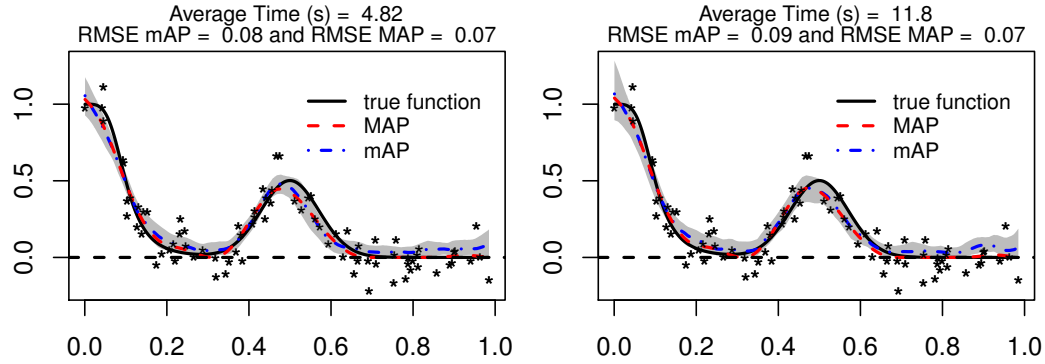

Fig. 9 Nonnegative function estimation using the efficient circulant embedding technique. The proposed approach (Fast.LS) is used in the left panel, while the FFT method (sample.WC) from Wood and Chan (1994) is employed in the right panel. The average computing time (in seconds) of running the McMC samplers for 6,000 iterations and the RMSE for both mAP and MAP estimates over 25 replicates are displayed in the main part of each panel. The dimension of the prior is fixed at $N = 100$.

3.4.3 Comparison with alternative McMC samplers

The aim of this section is to illustrate the computational complexity of the proposed approach compared to the highly efficient HMC sampler developed in Pakman and Paninski (2014) and implemented in the `R package` *tmg*. The monotonicity (nondecreasing) and boundedness (nonnegativity) constraints considered in Sections 3.4.1 and 3.4.2 are employed. In both cases, the covariates are generated uniformly on $[0, 1]$. The parameter η is fixed at 50 and the number of knots $N = \lceil \frac{n}{8} \rceil$. The Matérn covariance function is employed, with the smoothness parameter ν fixed at 1.5, and the length-scale parameter ℓ chosen to achieve a correlation of 0.05 at the maximum possible separation between the covariates. The other parameters σ and τ are updated using the joint posterior distribution as in Sections 3.4.1 and 3.4.2.

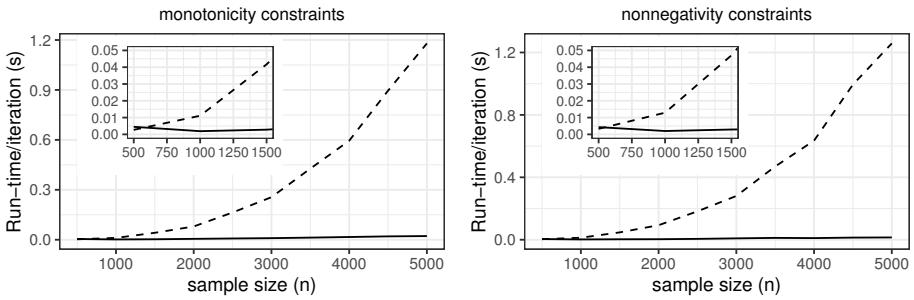

Fig. 10 Running time per McMC iteration (in seconds) plotted against the sample size (n) for two McMC samplers: the proposed approach denoted LS.ESS (black solid curve) and the HMC *tmg* (black dashed curve). The monotonicity nondecreasing constraint is considered in the left panel, while the nonnegative constraint is considered in the right panel. In both cases, the dimension N is fixed at $\lceil \frac{n}{8} \rceil$.

In Fig. 10, the average run-time per iteration (in seconds) over 10 replicates is plotted against the sample size n (varied between 500 and 5,000). The monotonicity constraint is considered in the left panel, while the nonnegativity constraint is applied in the right panel. The large-scale ESS (LS-ESS) approach developed in the present paper is compared to the efficient HMC sampler developed in Pakman and Paninski (2014). The proposed approach is represented by the black solid curve, while the HMC sampler is represented by the black dashed curve. The HMC performs well in low dimensions, specially when $N \leq 125$ (i.e., when $n \leq 1,000$). However, as both the sample size n and the dimension N of the MVN prior $\boldsymbol{\xi}$ increase, the proposed LS-ESS approach outperforms the HMC sampler. Unlike the HMC method, the LS-ESS approach grows linearly as a function of the sample size n and of the dimension N.

4 Conclusion

This paper introduces a new highly efficient approach for sampling from a large multivariate normal distribution. The key concept is to partition the input domain into smaller subdomains and utilize cross-correlation techniques among the connected subdomains to capture the correlation structure across the entire domain. The proposed approach is simple to implement and can be applied to a covariance function extracted from both stationary and non-stationary Gaussian processes. This strategy significantly reduces computational complexity, especially as the number of subdomains increases. The developed algorithm demonstrates superior efficiency compared to other existing methods. Its performance has been evaluated using the efficient circulant embedding technique proposed in Ray et al (2020); Zhou et al (2022) for shape-restricted function estimation.

References

Bay X, Grammont L, Maatouk H (2016) Generalization of the Kimeldorf-Wahba correspondence for constrained interpolation. Electron J Statist 10(1):1580–1595

Botev ZI (2017) The normal law under linear restrictions: simulation and estimation via minimax tilting. J R Stat Soc Series B Stat Methodol 79(1):125–148

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press

Cai B, Dunson DB (2007) Bayesian multivariate isotonic regression splines: applications to carcinogenicity studies. J Amer Statist Assoc 102(480):1158–1171

Chataigner M, Cousin A, Crépey S, et al (2021) Beyond surrogate modeling: Learning the local volatility via shape constraints. SIAM J Financ Math 12(3):SC58–SC69

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

Cotter SL, Roberts GO, Stuart AM, et al (2013) MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster. Stat Sci 28(3):424 – 446

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

Cousin A, Deleplace A, Misko A (2022) Gaussian process regression for swaption cube construction under no-arbitrage constraints. Risks 10(12):232

Crépey S, Dixon MF (2020) Gaussian process regression for derivative portfolio modeling and application to credit valuation adjustment computations. J Comput Finance 24(1)

Curtis SM, Ghosh SK (2011) A variable selection approach to monotonic regression with Bernstein polynomials. J Appl Stat 38(5):961–976

Goldfarb D, Idnani A (1983) A numerically stable dual method for solving strictly convex quadratic programs. Math Program 27(1):1–33

Golub G, Van Loan CF (1996) Matrix computations. The Johns Hopkins University Press

Grammont L, Maatouk H, Bay X (2022) Equivalent between constrained optimal smoothing and Bayesian estimation, preprint

Kimeldorf G, Wahba G (1970) A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. Ann Math Stat pp 495–502

Lin L, Dunson DB (2014) Bayesian monotone regression using Gaussian process projection. Biometrika 101(2):303–317

López-Lopera AF, Bachoc F, Durrande N, et al (2018) Finite-dimensional Gaussian approximation with linear inequality constraints. SIAM/ASA Journal on Uncertainty Quantification 6(3):1224–1255

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

Maatouk H, Rullière D, Bay X (2023a) Bayesian analysis of constrained Gaussian processes, URL https://hal.science/hal-04084865, under revision (round 2)

Maatouk H, Rullière D, Bay X (2023b) Sampling large hyperplane-truncated multivariate normal distributions. To appear in Comput Stat https://doi.org/10.1007/s00180-023-01416-7

Maradesa A, Py B, Quattrocchi E, et al (2022) The probabilistic deconvolution of the distribution of relaxation times with finite Gaussian processes. Electrochim Acta 413:140119

Meyer MC, Hackstadt AJ, Hoeting JA (2011) Bayesian estimation and inference for generalised partial linear models using shape-restricted splines. J Nonparametr Stat 23(4):867–884

Murphy KP (2018) Machine learning: A probabilistic perspective (adaptive computation and machine learning series)

Murray I, Adams R, MacKay D (2010) Elliptical slice sampling. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp 541–548

Neal RM (1999) Regression and classification using Gaussian process priors. Bayesian statistics pp 475–501

Neelon B, Dunson DB (2004) Bayesian isotonic regression and trend analysis. Biometrics 60(2):398–406

Pakman A, Paninski L (2014) Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. J Comput Graph Stat 23(2):518–542

Papp D, Alizadeh F (2014) Shape-constrained estimation using nonnegative splines. Journal of Computational and Graphical Statistics 23(1):211–231

Pensoneault A, Yang X, Zhu X (2020) Nonnegativity-enforced Gaussian process regression. Theor App Mech Lett 10(3):182–187

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

Riihimäki J, Vehtari A (2010) Gaussian processes with monotonicity information. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp 645–652

Shively TS, Walker SG, Damien P (2011) Nonparametric function estimation subject to monotonicity, convexity and other shape constraints. J Econom 161(2):166–181

Swiler LP, Gulian M, Frankel AL, et al (2020) A survey of constrained Gaussian process regression: Approaches and implementation challenges. JMLMC 1(2)

Taylor J, Benjamini Y (2016) RestrictedMVN: multivariate normal restricted by affine constraints. R package version 1

Tran TT, Fradi A, Samir C (2023) Learning, inference, and prediction on probability density functions with constrained Gaussian processes. Inf Sci 642:119068

Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning, vol 2. MIT press Cambridge, MA

Williams NJ, Osborne C, Seymour ID, et al (2023) Application of finite Gaussian process distribution of relaxation times on SOFC electrodes. Electrochem commun p 107458

Wilson JT, Borovitskiy V, Terenin A, et al (2021) Pathwise conditioning of Gaussian processes. JMLR 22(105):1–47

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

Zhou S, Giulani P, Piekarewicz J, et al (2019) Reexamining the proton-radius problem using constrained Gaussian processes. Phys Rev C 99:055202

Zhou S, Ray P, Pati D, et al (2022) A mass-shifting phenomenon of truncated multivariate normal priors. J Amer Statist Assoc 0(ja):1–37