INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Gaussian Process Controlled B-Spline Surface

Yongxiang Li, Yu Tian, Huadong Mo, Shichang Du

To cite this article:
Yongxiang Li, Yu Tian, Huadong Mo, Shichang Du (2026) Gaussian Process Controlled B-Spline Surface. INFORMS Journal on Data Science

Published online in Articles in Advance 27 Jan 2026

. https://doi.org/10.1287/ijds.2024.0061

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Articles in Advance, pp. 1–19
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

Gaussian Process Controlled B-Spline Surface

Yongxiang Li,[a] Yu Tian,[a] Huadong Mo,[b] Shichang Du[a,*]

[a] Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai 200240, P.R. China; [b] School of Systems & Computing, University of New South Wales, Canberra, Australian Capital Territory 2600, Australia

*Corresponding author

Contact: yongxiangli@sjtu.edu.cn, https://orcid.org/0000-0003-0618-6857 (YL); tianyu_202308@sjtu.edu.cn (YT); huadong.mo@unsw.edu.au, https://orcid.org/0000-0002-7782-2884 (HM); lovbin@sjtu.edu.cn, https://orcid.org/0000-0003-2408-722X (SD)

Received: December 23, 2024
Revised: September 20, 2025; December 1, 2025
Accepted: December 8, 2025
Published Online in Articles in Advance: January 27, 2026

https://doi.org/10.1287/ijds.2024.0061

Abstract. We propose a Gaussian process controlled B-spline surface (GPBSS), which integrates the flexibility of B-spline basis functions into the probabilistic framework of Gaussian processes. By leveraging the sparsity inherent in B-spline bases, GPBSS achieves a linear time complexity, making it particularly effective for large-scale data sets in low-dimensional spaces. Compared with current benchmark approximations of the standard Kriging model, GPBSS offers a unique balance between computational efficiency and prediction accuracy. Furthermore, we extend the application of the GPBSS model to Bayesian optimization, enabling efficient optimization of black box functions. To validate the performance of GPBSS, we conduct a regression study on four large-scale data sets and an optimization study on three complex objective functions. The results demonstrate that our proposed model not only significantly enhances computational efficiency but also excellently balances its prediction accuracy. Its favorable tradeoff makes GPBSS a valuable tool for data-intensive regression and optimization tasks in low-dimensional scenarios such as medical imaging, geospatial analysis, and additive manufacturing, where data are sampled at high rates or over long intervals.

History: Eunshin Byon served as the senior editor for this article.

Funding: This work was funded by the National Natural Science Foundation of China [Grants 72471142, 72101147, 52275499, and 92467101].

Supplemental Material: The code capsule is available at https://github.com/Yongxiang-Li/GPBSS and in the e-companion to this article (available at https://doi.org/10.1287/ijds.2024.0061).

Keywords: kriging • computer experiments • Bayesian optimization • uncertainty quantification • inducing points method

1. Introduction

Gaussian process (GP) models (Sacks et al. 1989, Cressie 1993, Santner et al. 2003), also known as Kriging models, have gained widespread recognition as powerful tools for modeling and optimizing complex systems across diverse domains, including machine learning (Rasmussen and Williams 2006), engineering design (Su et al. 2017), materials design (Zhang et al. 2020), signal processing (Li et al. 2024b), and environmental modeling (Sun et al. 2024). The core strength of GP lies in its probabilistic regression framework, which not only provides predictions but also quantifies the uncertainty associated with these predictions (Bilionis and Zabaras 2012, Che et al. 2024). This characteristic makes it highly effective for further tasks such as Bayesian optimization (BO; Snoek et al. 2012), where the goal is to efficiently identify the optimal solution to a black-box optimization problem.

Despite its versatility, the standard Kriging model faces significant challenges when applied to fields that require the analysis of large-scale data sets (Li et al. 2024a), such as medical imaging (Salimi-Khorshidi et al. 2011), geospatial data (Atkinson and Lloyd 1998, Boer et al. 2001, Son et al. 2019), and additive manufacturing data (Xu et al. 2024). The computational complexity of the Kriging model, which scales cubically with the number of data points, renders it computationally impractical for large-scale data sets (Schulz et al. 2018). This limitation has spurred the development of various approximation methods (Kleijnen 2009, Liu et al. 2020, Fuhg et al. 2021) that aim to reduce the computational burden of GPs while maintaining acceptable prediction accuracy.

Approximation methods often rely on simplifying either likelihood formulations or covariance structures. Composite likelihood methods (Stein et al. 2004, Eidsvik et al. 2014) approximate the full likelihood of the GP by the product of several marginal or conditional likelihoods (Lindsay 1988, Lindsay et al. 2011, Varin et al. 2011). Another line of work is the Vecchia approximation (Vecchia 1988), which represents the joint distribution as an ordered product of valid conditional distributions, each conditioning only on a few nearby observations. Inducing-point methods (Titsias 2009, Wilson and Nickisch 2015) are the most common approaches that utilize

simplified covariance structures. Covariance tapering (Furrer et al. 2006, Kaufman et al. 2008), another popular method, simplified the covariance structure by multiplying the full covariance matrix by a tapering function, typically a compactly supported correlation function, thus improving computational efficiency for large-scale data sets. These methods obtain a scalability-accuracy trade-off either by enforcing sparsity to capture local dependencies at the expense of losing long-range information (e.g., composite likelihood methods and covariance tapering) or by using inducing points to approximate the full covariance for computational efficiency.

Although these approaches have substantially improved the computational scalability, they approximate the full GP by sacrificing certain dependency information, thereby leaving room for improvement in the scalability-accuracy tradeoff. To achieve a unique tradeoff, we introduce the GP controlled B-spline surface (GPBSS), which is neither a pseudo-likelihood approximation nor a low-rank approximation of the covariance. GPBSS models a small number of B-spline control points via GP to retain long-range correlation and utilizes the computationally efficient sparse B-spline basis to capture local dependencies.

To enhance the model's adaptability, we introduce a sequential knot number selection (SKNS) technique, which provides effective guidance for selecting the optimal number of control points in GPBSS. Additionally, we integrate GPBSS with BO, using GPBSS as a surrogate model, to enable efficient optimization of complex black-box functions. Numerical examples demonstrate that the GPBSS model is significantly efficient for data-intensive regression and optimization tasks with no more than four input dimensions.

The main novelties of this work are as follows. First, GPBSS achieves a linear time complexity, which significantly reduces the computational costs compared with the standard Kriging model. Second, unlike existing approximation methods, GPBSS exploits the geometric properties of B-splines to achieve a more favorable position on the scalability-accuracy Pareto frontier in low-dimensional numerical examples. Third, the SKNS technique offers a faster and more efficient method for selecting knots, setting it apart from traditional knot selection approaches.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive review of related work, covering the partial spline models, the standard Kriging model, and current benchmark approximations. Section 3 introduces the GPBSS model, detailing model construction, parameter estimation, knot number selection, model prediction, and the GPBSS-based BO. Numerical experiments on regression and optimization are presented in Sections 4 and 5, demonstrating the efficacy of GPBSS in practical applications. Finally, Section 6 concludes the paper. The code for reproducing our results is available at https://github.com/Yongxiang-Li/GPBSS.

2. Literature Review

2.1. Partial Spline

Partial spline models form a foundational class of semiparametric regression methods, providing a unified framework that seamlessly accommodates both linear and nonlinear components (Powell 1994, Ruppert 2003). A standard one-dimensional partial spline model (Gu 2013) can be formulated as

$$y(x) = \boldsymbol{f}(x)^T \boldsymbol{\beta} + \eta(x) + \epsilon(x), \tag{1}$$

where $\boldsymbol{f}(x)$ represents the set of covariates with corresponding coefficients $\boldsymbol{\beta}$, $\eta(x)$ is the nonparametric component represented by spline functions, and $\epsilon(x)$ is an independently and identically distributed (i.i.d.) noise term. When dealing with multiple predictors, tensor product spline constructions can be employed to effectively capture interactions among variables (Gu 2013).

A common strategy for estimating the unknown function $\eta(x)$ in Equation (1) is to frame the problem as minimizing a penalized least squares criterion in an appropriate Hilbert space. More specifically, let $\mathcal{H} = \{\eta : J(\eta) < \infty\}$ be a Hilbert space defined by a quadratic functional $J(\eta)$ that measures the smoothness of $\eta(x)$. The estimation procedure involves solving the following optimization problem:

$$\min_{\eta} \frac{1}{n} \sum_{i=1}^{n} \big(y_i - \eta(x_i)\big)^2 + \lambda J(\eta), \tag{2}$$

where λ is a penalty parameter that balances the tradeoff between the goodness-of-fit and smoothness of $\eta(x)$. The choice of $J(\eta)$ and its associated Hilbert space structure can be guided by established theoretical frameworks (Wahba 1990, Gu 2013, Ma et al. 2015).

The Hilbert space $\mathcal{H}$ can be decomposed into the direct sum of the null space $\mathcal{N}_J$ and its orthogonal complement $\mathcal{H}_J$:

$$\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J,$$

where $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$ is the null space of $J(\eta)$. $\mathcal{N}_J$ is commonly assumed to be a finite-dimensional linear subspace of $\mathcal{H}$ with basis $\{\xi_i : i = 1, \ldots, s\}$, where $s = \dim(\mathcal{N}_J)$. The orthogonal complement $\mathcal{H}_J$ inherits the structure of a reproducing kernel Hilbert space (RKHS) with its reproducing kernel $R_J(\cdot, \cdot)$.

Building on the work of Wahba (1990), the minimizer of the optimization problem in Equation (2) over $\mathcal{H}$ can be expressed as

$$\eta(x) = \sum_{k=1}^{s} \alpha_k \xi_k(x) + \sum_{i=1}^{n} c_i R_J(x_i, x). \tag{3}$$

This representation leverages the direct sum decomposition of $\mathcal{H}$, where the basis functions ξ_i span the null space $\mathcal{N}_J$, and $R_J(\cdot, \cdot)$ serves as the reproducing kernel for the orthogonal complement $\mathcal{H}_J$.

Although partial spline models effectively balance interpretability and flexibility (Ruppert 2003, Gu 2013), they fall short in assessing the prediction uncertainty. This limitation particularly hinders certain applications, such as BO, which relies on uncertainty quantification to make informed decisions about where to sample next (Morris et al. 1993), so there is a need for models that retain the interpretability and flexibility of partial splines while incorporating robust uncertainty quantification.

2.2. Gaussian Process

GP (or Kriging) models are powerful and flexible nonparametric methods that are widely used for regression tasks (Santner et al. 2003, Rasmussen and Williams 2006). For $\boldsymbol{x} \in \mathbb{R}^p$, the GP model is commonly defined as

$$y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{\beta} + z(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{4}$$

where $z(\boldsymbol{x})$ follows a GP with a zero mean function and a covariance function $K_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{x}')$, and $\epsilon(\boldsymbol{x})$ represents i.i.d. Gaussian noise. The standard Kriging model typically employs maximum likelihood estimation (MLE) to estimate the model parameters $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ (Santner et al. 2003).

The core advantage of GP models lies in their probabilistic regression framework that captures uncertainty in predictions, making them highly effective for further tasks such as BO. However, the standard Kriging model faces significant challenges when dealing with large-scale data sets due to its $\mathcal{O}(n^3)$ computational complexity, where n is the number of sample points. This limitation has motivated the development of various approximation methods that aim to reduce computational costs while preserving prediction accuracy.

The earliest attempt to address the computational issues of GP involved composite likelihood methods. These methods typically partitioned the entire data set into several subsets and defined a marginal or conditional likelihood for each subset (Besag 1975, Lindsay 1988). The full likelihood was often approximated by the product of these composite likelihoods (Lindsay et al. 2011, Varin et al. 2011). For example, composite conditional likelihood methods approximated the full likelihood of a GP by using the product of the conditional likelihoods on each subset of the data (Stein et al. 2004). Similarly, composite marginal likelihood methods (Heagerty and Lele 1998, Caragea and Smith 2007, Eidsvik et al. 2014) approximated the full likelihood by using the product of the marginal likelihoods on each subset of the data.

In contrast, the Vecchia approximation (Vecchia 1988) replaces the joint distribution with a product of univariate conditional distributions, each conditioning only on a small set of nearby observations in a given ordering. To enhance the Vecchia approximation, Guinness (2018) introduced a new grouping scheme and showed that ordering choices beyond the default coordinate-based orderings can improve approximation accuracy. Katzfuss and Guinness (2021) proposed a general framework for Vecchia approximations, which was extended by Katzfuss et al. (2020) to GP prediction, achieving an $\mathcal{O}(n)$ computational complexity for both parameter estimation and prediction.

Another attempt is using simplifies covariance structures for the GP to address computational issues. Covariance tapering approaches employ a sparse covariance matrix to accelerate the GP modeling (Furrer et al. 2006, Kaufman et al. 2008). In covariance tapering, sparsity is achieved by multiplying the original covariance function by a compactly supported covariance function (the taper function), which forces the resulting covariance matrix to be sparse. An alternative approach using simplified covariance structures, known as the inducing-point method (Hensman et al. 2013, Liu et al. 2020) or the low-rank method (Cressie and Johannesson 2008, Stein 2008), employs a low-rank covariance function to approximate the original covariance function. The computational complexity can be reduced to $\mathcal{O}(nm^2 + m^3)$, where m is the number of inducing points. For example, the sparse pseudo-input GP (SPGP; Snelson and Ghahramani 2006) introduced a small set of pseudo-input points to represent the entire data set, thereby reducing the effective sample size. GP regression reconstruction (GPRR; Xiong 2021) employed an interpolation-based method with finite knots to efficiently approximate GPs, offering

reduced computational complexity. Titsias (2009) proposed a variational sparse GP framework to jointly optimize inducing inputs and kernel hyperparameters through an evidence lower bound. Building on inducing-point methods, KISSGP further enhances computational efficiency by employing kernel interpolation techniques (Wilson and Nickisch 2015), reducing the computational complexity to $\mathcal{O}(n)$.

These approximation methods for scalable GP can generally be viewed as different strategies for balancing long-range and local dependencies, offering a tradeoff between computational efficiency and prediction accuracy (Sang and Huang 2012, Katzfuss 2017). Local dependencies typically lead to small or even sparse matrices that are computationally easy to handle, whereas long-range dependencies grow with the data set size and are often dense, posing a significant computational burden. Composite likelihood methods and covariance tapering accelerate computation by largely discarding long-range dependencies, relying primarily on local structure for prediction. Inducing-point methods mainly incorporate long-range dependencies through a correlation function defined on inducing and data points to approximate the full correlation for computational acceleration. However, the low-rank structure may limit their ability to capture local dependencies. Despite a substantial improvement in scalability, these approximation methods sacrifice certain dependency information, thereby leaving room for improvement in the scalability-accuracy tradeoff.

Recently, Ghosh et al. (2021) proposed a GP-controlled B-spline (GPBS) model, which combines multivariate GPs (MGPs; Fricker et al. 2013, Chen et al. 2023) and B-spline curves to flexibly model multivariate profiles of low-emission glasses over a one-dimensional input space. The model is formulated within a linear mixed-effects framework focusing on multioutput problems. A key innovation of the GPBS model is that it models the B-spline control points as an MGP, allowing joint modeling of within- and between-profile correlations in a probabilistic framework. GPBS exhibits considerable computational efficiency and numerical stability over conventional MGPs. Although the GPBS model presents a structured alternative to conventional GP models, it is validated only in one-dimensional and multioutput problems. Moreover, it does not explicitly exploit the sparsity of the B-spline basis, which could offer scalable computational advantages.

3. GPBSS

To address the aforementioned issues, this study proposes the GPBSS model. By combining the flexibility of B-spline basis functions with the probabilistic regression framework of GPs, this study creates a scalable and efficient surrogate model that maintains a high prediction accuracy. This section details the GPBSS modeling process, focusing on parameter estimation, knot number selection, and model prediction. The final section presents an application of GPBSS to BO.

3.1. Model Formulation

This study proposes the GPBSS model, an extension of GPBS, for surface modeling in higher dimensions. Unlike GPBS, which was validated only for one-dimensional problems, the GPBSS model integrates B-spline surfaces, rather than B-spline curves, into the GP framework. Additionally, this study leverages the sparsity of the B-spline basis matrix to improve computational efficiency during model fitting, a benefit not fully exploited by GPBS. The remainder of this section details the formulation of the GPBSS model.

B-spline surfaces in the bivariate input space can be expressed as

$$\mathcal{S}(x_1, x_2) = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} u_{i,d}(x_1) u_{j,d}(x_2) \Gamma_{i,j},$$

where $u_{i,d}(x)$ denotes the ith B-spline basis function of degree d, m_k denotes the number of B-spline basis functions in the kth input dimension for $k = 1, 2$, and $\Gamma_{i,j}$ is the (i,j)th element of the coefficient matrix $\mathbf{\Gamma}$. The matrix form of the B-spline surface is

$$\mathcal{S}(\boldsymbol{x}) = (\mathrm{u}(x_1)^T \otimes \boldsymbol{u}(x_2)^T)\boldsymbol{\gamma},$$

where $\boldsymbol{x} = [x_1, x_2]^T$, and $\boldsymbol{\gamma} = \mathrm{Vec}(\mathbf{\Gamma})$ is the vectorization of the coefficient matrix.

It is a common practice to treat the coefficients $\{\Gamma_{i,j}\}$ as control points (Li et al. 2025). Specifically, the coefficients $\{\Gamma_{i,j}\}$ constitute a control lattice, an equally spaced grid of control points that governs the shape of the B-spline surface. The indices i and j correspond to the coordinates of each control point $\Gamma_{i,j}$ in the logical grid defined by the B-spline basis functions along the two input directions. These control points act as representative points and capture the essential structure of the entire data set, similar to the inducing points in GPRR (Xiong 2021), SPGP (Snelson and Ghahramani 2006), and KISSGP (Wilson and Nickisch 2015).

The B-spline surface can be extended beyond the bivariate scenario to higher-dimensional cases, where the input vector $\boldsymbol{x}$ is denoted as $\boldsymbol{x} = [x_1, x_2 \cdots, x_p]^T$, where $p \geq 2$ denotes the dimension of the input space. The B-spline surface $\mathcal{S}(\boldsymbol{x})$ is extended to higher dimensions by combining all univariate B-spline basis functions. Specifically, the multidimensional basis function of $\mathcal{S}(\boldsymbol{x})$ is the Kronecker-based structure of the univariate B-spline basis functions for each dimension. Let $\boldsymbol{u}(x_k)$ represent the B-spline basis functions for the kth dimension. The multidimensional basis function vector $\boldsymbol{u}(\boldsymbol{x})$ is then given by

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(x_1) \otimes \boldsymbol{u}(x_2) \otimes \cdots \otimes \boldsymbol{u}(x_p),$$

resulting in a comprehensive set of basis functions that span the multidimensional input space. Thus, the surface $\mathcal{S}(\boldsymbol{x})$ can be expressed as

$$\mathcal{S}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})^T \boldsymbol{\gamma} = (\boldsymbol{u}(x_1)^T \otimes \boldsymbol{u}(x_2)^T \otimes \cdots \otimes \boldsymbol{u}(x_p)^T)\boldsymbol{\gamma}, \tag{5}$$

where $\boldsymbol{\gamma}$ is the vectorization of the B-spline coefficient tensor $\boldsymbol{\Gamma}$.

Based on the B-spline surface $\mathcal{S}(\boldsymbol{x})$, the proposed GPBSS model is

$$y(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{\beta} + \boldsymbol{u}(\boldsymbol{x})^T \boldsymbol{\gamma} + \epsilon(\boldsymbol{x}), \tag{6}$$

where $\boldsymbol{f}(\boldsymbol{x})$ and $\epsilon(\boldsymbol{x})$ are defined as in Equation (4). Unlike conventional B-spline regression, which assumes independent coefficients and thus neglects spatial-temporal correlation, GPBSS assumes that the coefficient tensor $\boldsymbol{\Gamma}$ follows a GP with zero mean and a separable covariance function defined as

$$\mathrm{Cov}(\Gamma_{i_1,i_2,\ldots,i_p}, \Gamma_{i'_1,i'_2,\ldots,i'_p}) = \sigma^2 \prod_{k=1}^{p} \varphi_{\theta_k}(i_k, i'_k), \tag{7}$$

where $i_k, i'_k \in \{1, 2, \ldots, m_k\}$ denote the indices of $\Gamma_{i_1,i_2,\ldots,i_p}$ and $\Gamma_{i'_1,i'_2,\ldots,i'_p}$ along the kth dimension and $\varphi_{\theta_k}(i_k, i'_k) = \exp(-\theta_k^2(i_k - i'_k)^2/m_k^2)$. Modeling the coefficients as GP allows GPBSS to account for spatial-temporal correlation.

The separable covariance function in Equation (7) is a standard choice that has been widely used in the GP literature. An alternative is to employ nonseparable kernels (Cao et al. 2021, Hristopulos 2024), which can capture more complex cross-dimensional dependencies. However, we conducted some preliminary simulation studies and found that the gain in predictive accuracy from using nonseparable kernels in GPBSS is marginal, whereas the computational burden increases substantially. This is because the B-spline basis in GPBSS has a Kronecker-based structure, which may limit the potential benefits of nonseparable kernels. Therefore, we adopt the separable covariance function in Equation (7) in GPBSS throughout this study.

Then, the correlation matrix of $\boldsymbol{\gamma}$ can be expressed as

$$\boldsymbol{R} = \boldsymbol{R}_1 \otimes \boldsymbol{R}_2 \otimes \cdots \otimes \boldsymbol{R}_p.$$

Here,

$$\boldsymbol{R}_k = \{\exp(-\theta_k^2(i_k - i'_k)^2/m_k^2)\}_{1 \leq i_k, i'_k \leq m_k} \tag{8}$$

is the correlation matrix corresponding to the kernel $\varphi_{\theta_k}(\cdot, \cdot)$. Thus, the covariance function of the proposed GPBSS model is

$$\mathrm{Cov}(y(\boldsymbol{x}), y(\boldsymbol{x}')) = \sigma^2 \boldsymbol{u}(\boldsymbol{x})^T \boldsymbol{R} \boldsymbol{u}(\boldsymbol{x}'),$$

which indicates that GPBSS is itself a valid GP model with a well-defined covariance function. Compared with partial splines, which map the input $\boldsymbol{x}$ into RKHS and represent the model as a deterministic linear combination of basis functions and the reproducing kernel, the proposed approach uses kernel functions to map the regression coefficients $\boldsymbol{\gamma}$ into a high-dimensional space.

GPBSS incorporates B-splines into the probabilistic regression framework of GPs, providing uncertainty quantification of complex surfaces. Note that we recommend using GPBSS primarily for low-dimensional problems (i.e., $p \leq 4$) due to the Kronecker-based structure of the B-spline surface $\mathcal{S}(\boldsymbol{x})$.

3.2. Parameter Estimation

In this section, we elaborate on the parameter estimation procedure for the GPBSS model. Denote the n design points by $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, and their corresponding responses are denoted by $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$. The matrix form of the GPBSS model can be expressed as

$$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{U}\boldsymbol{\gamma} + \epsilon,$$

where $\boldsymbol{F} = \left[\boldsymbol{f}(\boldsymbol{x}_1), \ldots, \boldsymbol{f}(\boldsymbol{x}_n)\right]^T$ is the regression matrix, $\boldsymbol{U} = [\boldsymbol{u}(\boldsymbol{x}_1), \ldots, \boldsymbol{u}(\boldsymbol{x}_n)]^T$ is the B-spline basis matrix, and $\epsilon = [\epsilon(\boldsymbol{x}_1), \ldots, \epsilon(\boldsymbol{x}_n)]^T$. This study assumes that $\epsilon \sim \mathcal{N}\left(0, \sigma^2\delta^2\boldsymbol{I}_n\right)$, and thus the covariance matrix of $\boldsymbol{y}$ is $\sigma^2\boldsymbol{\Sigma}$, where

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^T + \delta^2 I_n. \tag{9}$$

To facilitate efficient computation, the matrix inversion lemma (also known as the Sherman-Morrison-Woodbury formula; Sherman and Morrison 1950) is employed, and thus the inverse of $\boldsymbol{\Sigma}$ can be simplified as

$$\boldsymbol{\Sigma}^{-1} = \frac{\boldsymbol{I}_n}{\delta^2} - \frac{\boldsymbol{U}\boldsymbol{\Xi}^{-1}\boldsymbol{U}^T}{\delta^2}, \tag{10}$$

where $\boldsymbol{\Xi} = \delta^2\boldsymbol{R}^{-1} + \boldsymbol{U}^T\boldsymbol{U}$, in which $\boldsymbol{R}^{-1} = \boldsymbol{R}_1^{-1} \otimes \boldsymbol{R}_2^{-1} \otimes \cdots \otimes \boldsymbol{R}_p^{-1}$, and using the B-spline basis structure, we have

$$\boldsymbol{U}^T\boldsymbol{U} = \sum_{i=1}^{n} \boldsymbol{u}(\boldsymbol{x}_i)\boldsymbol{u}^T(\boldsymbol{x}_i) = \sum_{i=1}^{n} (\boldsymbol{u}(x_{i,1})\boldsymbol{u}^T(x_{i,1})) \otimes \cdots \otimes (\boldsymbol{u}(x_{i,p})\boldsymbol{u}^T(x_{i,p})).$$

It is worth noting that the matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Xi}$ remain positive definite even if $n < m$, which is critical for the implementation of GPBSS in BO, particularly during the early stages when n is sometimes less than m.

Similarly, according to Sylvester determinant theorem (Akritas et al. 1996), we have

$$|\boldsymbol{\Sigma}| = \delta^{2(n-m)}|\boldsymbol{R}|\,|\boldsymbol{\Xi}|, \tag{11}$$

where $m = m_1 \cdots m_p$ and

$$|\boldsymbol{R}| = \prod_{i=1}^{p} |\boldsymbol{R}_i|^{\frac{m}{m_i}}.$$

Although the matrix inversion lemma and the Sylvester determinant theorem are also utilized in conventional approximation approaches such as inducing-point methods or low-rank methods, the computational complexity of the proposed method is much lower. Note that each row of the basis matrix $\boldsymbol{U}$ contains only $q = (d+1)^p$ nonzero entries, where d is typically small (e.g., $d = 2, 3,$ or 4). Consequently, the sparse basis matrix $\boldsymbol{U}$ reduces the computational complexity of $\boldsymbol{\Sigma}^{-1}$ and $|\boldsymbol{\Sigma}|$ from $\mathcal{O}(nm^2 + m^3)$ to $\mathcal{O}(n + m^3)$. This sparse property and its associated computational acceleration were not fully recognized in the GPBS modeling. Furthermore, this sparsity allows GPBSS to achieve a better balance between computational efficiency and prediction accuracy compared with other inducing-point methods. In practice, we recommend setting $m_i < n^{1/p}$ on average to preserve this tradeoff.

The likelihood function, up to a constant, is

$$\ell(\boldsymbol{\phi}, \boldsymbol{\beta}, \sigma^2) = -\frac{\log\left|\sigma^2\boldsymbol{\Sigma}\right|}{2} - \frac{(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})}{2\sigma^2}, \tag{12}$$

where $\boldsymbol{\phi} = \left\{\theta_1, \ldots, \theta_p, \delta\right\}$. By setting the derivatives of the likelihood function to zero, the parameter estimates of $\boldsymbol{\beta}$ and σ^2 are

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{F}_U^T\boldsymbol{\Xi}^{-1}\boldsymbol{F}_U)^{-1}(\boldsymbol{F}^T\boldsymbol{y} - \boldsymbol{F}_U^T\boldsymbol{\Xi}^{-1}\boldsymbol{y}_U) \tag{13}$$

and

$$\hat{\sigma}^2 = \frac{\|\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}\|_2^2 - (\boldsymbol{y}_U - \boldsymbol{F}_U\hat{\boldsymbol{\beta}})^T\boldsymbol{\Xi}^{-1}(\boldsymbol{y}_U - \boldsymbol{F}_U\hat{\boldsymbol{\beta}})}{n\delta^2}, \tag{14}$$

where $\boldsymbol{F}_U = \boldsymbol{U}^T\boldsymbol{F}$ and $\boldsymbol{y}_U = \boldsymbol{U}^T\boldsymbol{y}$. Note that computing $\boldsymbol{U}^T\boldsymbol{U}$ in $\boldsymbol{\Xi}$ only has a $\mathcal{O}(n)$ complexity, and thus computing $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ costs $\mathcal{O}(n + m^3)$ time. Substituting Equations (13) and (14) into Equation (12), the profile likelihood, up to a constant, becomes

$$\hat{\ell}(\boldsymbol{\phi}) = -\frac{n\log\hat{\sigma}^2 + (n-m)\log\delta^2 + \log|\boldsymbol{R}|\,|\boldsymbol{\Xi}|}{2}. \tag{15}$$

Computing this profile likelihood $\hat{\ell}(\boldsymbol{\phi})$ also costs $\mathcal{O}(n + m^3)$ time. Maximizing the profile likelihood $\hat{\ell}(\boldsymbol{\phi})$ with respect to $\boldsymbol{\phi}$ gives the parameter estimates:

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}}\{-\hat{\ell}(\boldsymbol{\phi})\}. \tag{16}$$

This optimization problem is solved using numerical techniques such as gradient descent methods. The covariance structure of GPBSS, as described earlier in Equation (9), facilitates computational efficiency and allows

scalable modeling of large-scale data sets. By exploiting the inherent sparsity of the B-spline basis, the computational complexity of evaluating the covariance matrix and performing matrix operations is significantly reduced, making the GPBSS model computationally feasible for large-scale data sets.

3.3. Sequential Knot Number Selection

Knot selection plays a pivotal role in determining the effectiveness of the GPBSS model by balancing overfitting and underfitting (Yuan et al. 2013). Thus, this study focuses specifically on knot number selection. For readers interested in knot position selection, please refer to He et al. (2001).

In the traditional knot number selection (TKNS), the knot numbers for all dimensions are selected simultaneously. This approach requires complex optimization in a high-dimensional space where the knot number for each dimension is varied simultaneously, making the process less efficient. To address these issues, we propose the SKNS method, which employs the Akaike information criterion (AIC; Akaike 1998) to sequentially optimize the number of knots for each dimension.

Specifically, the number of control points for the ith dimension is determined by

$$\hat{m}_i = \arg\min_{m_i \in \mathcal{M}_i} 2\left(\prod_{j=1}^{i-1} \hat{m}_j\right) m_i \left(\prod_{j=i+1}^{p} m_j^0\right) - \hat{\ell}(\hat{\boldsymbol{\phi}}_0) \tag{17}$$

for $i = 1, \ldots, p$, where $\mathcal{M}_i$ is the set of all candidate numbers for ith dimension, and $\hat{\boldsymbol{\phi}}_0$ is pre-estimated only once according to Equation (16) given $\{m_i^0\}_{i=1\ldots p}$, for the sake of computational efficiency in knot number selection. Here, m_i^0 is a user-specified initial value in $\mathcal{M}_i$, typically chosen as the median of the candidate values in $\mathcal{M}_i$ for $i = 1, \ldots, p$. Then, the optimized knot number in the ith dimension is $\hat{m}_i + d + 1$ as shown in Prautzsch et al. (2002). It is worth noting that the knot locations are not optimized, but rather the grid points in the knot space.

SKNS performs sequential optimization, starting with the first dimension and proceeding sequentially. That is, m_i is optimized conditional on $\{\hat{m}_1, \ldots, \hat{m}_{i-1}\}$ determined in the previous steps. This method eliminates the need to optimize $m_1, \ldots, m_p$ simultaneously, offering a more efficient and scalable solution for knot selection in high-dimensional spaces. To illustrate the efficacy of SKNS, we present the following example.

Example 1. In this example, we perform model fitting and prediction on the built-in two-dimensional (2D) function peaks in MATLAB.[1] We first generate the training data set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ using Latin hypercube design (LHD; Park 1994). Both the SKNS and TKNS are employed to predict the data under different n values, and the experiments are repeated 10 times. The average root mean square error (RMSE), the computation time of knots selection for both methods, and the total computation time are recorded. The RMSE is calculated on $2n$ testing data points, which are also generated by LHD. Note that we also report the knot selection time for each dimension in SKNS.

The results are shown in Table 1, and the RMSE for both the SKNS and TKNS methods is very similar, with TKNS being slightly smaller. Nevertheless, the knot-selection time required by the SKNS method is almost 1/10th of that required by the TKNS method.

Table 1. Average RMSE and Computing Time Using Different Knot Selection Methods

	RMSE		Computing time (s)				
			SKNS			TKNS	
			Knot selection time				
n	SKNS	TKNS	m_1	m_2	Total time	Knot selection time	Total time
10,000	0.01594	0.01585	1.0189	0.9824	2.1998	18.5375	18.8527
20,000	0.01198	0.01150	2.1318	2.0885	4.5335	41.2410	41.8709
30,000	0.01040	0.00936	3.2880	3.1666	6.9985	64.2007	65.2289
40,000	0.00863	0.00822	4.5617	4.3957	9.3794	87.7423	87.9958
50,000	0.00795	0.00761	5.6219	5.5549	11.7910	110.1739	110.5093
60,000	0.00698	0.00680	6.7060	6.6997	14.1419	130.4400	131.9203
70,000	0.00652	0.00652	7.9946	7.7410	16.5969	155.2484	156.0388
80,000	0.00652	0.00622	9.5003	9.4278	19.5299	183.7709	184.4952
90,000	0.00589	0.00591	11.5172	11.0389	23.4761	217.8534	219.8178
100,000	0.00553	0.00556	12.4564	12.4688	25.2279	234.9568	236.6990

It is worth noting that, although using more knots may introduce a potential risk of overfitting, such risk is not severe for the following three reasons. First, as indicated by Equation (5), the high-dimensional B-spline surface is essentially a mesh grid of several one-dimensional B-spline curves. Each one-dimensional B-spline curve has only $m_i + d + 1$ knots, which is considerably smaller than n. This mesh grid structure makes the proposed GPBSS model less susceptible to severe overfitting. Second, the design points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ are often randomly generated by LHD, and their responses $\boldsymbol{y}$ usually contain observational noises. This inherent randomness in the sample data makes it difficult to overfit the data using the mesh-grid surface. Third, unlike traditional linear regression using the B-spline, the control points in GPBSS are regulated by a GP, which enforces the smoothness of the B-spline and thus mitigates the risk of overfitting. These points are further illustrated in Example 2.

Example 2. Continuing with the peaks function, we perform model fitting using $n = 900$ sample points and prediction using $10n$ sample points. The number of control points in two dimensions is increased from 10 to 30 in increments of 5, and the process is repeated 100 times. The RMSE results are provided in Table 2. It can be observed that the optimal RMSE occurs at $m_1 = 30$ and $m_2 = 25$. When the number of control points continues to increase (for example, at the setting of $m_1 = 30$ and $m_2 = 30$), the prediction RMSE exhibits a slight rise, implying a potential risk of overfitting when the number of control points becomes excessively large. However, this RMSE increase is not substantial, and the prediction accuracy remains acceptable, indicating that using more control points does not necessarily lead to severe overfitting. Overall, these results suggest that GPBSS can be effectively applied as a surrogate model in BO, even when the sample size n is smaller than m during the early stages of optimization.

Although SKNS costs substantially less time than TKNS, it still multiplies the computational burden of GPBSS, because more than 90% of the total computational time is consumed by SKNS, as shown in Table 1. Thus, we recommend a two-step approach for a compromise choice: Users may first apply SKNS to a relatively small subset of the data set to determine an initial knot selection, and these preliminary knots can be then used to fit the GPBSS model on the whole data set. In addition, additional control points can be added to enhance model accuracy if computational resources or budgets permit.

3.4. Model Prediction

After parameter estimation, the next crucial step in the GPBSS model is to make predictions at new or unobserved design points. This section details the prediction of GPBSS using the estimated parameters. The conditional distribution of $\boldsymbol{\gamma}$ given $\boldsymbol{y}$ is

$$\boldsymbol{\gamma} \mid \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{R}\boldsymbol{U}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\beta}), \hat{\sigma}^2(\boldsymbol{R} - \boldsymbol{R}\boldsymbol{U}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{U}\boldsymbol{R})).$$

By simplifying the expression, the estimation of $\boldsymbol{\gamma}$ is

$$\hat{\boldsymbol{\gamma}} = \boldsymbol{\Xi}^{-1}(\boldsymbol{y}_U - \boldsymbol{F}_U\hat{\boldsymbol{\beta}}),$$

and its covariance matrix is $\hat{\sigma}^2(\boldsymbol{I} - \boldsymbol{\Xi}^{-1}\boldsymbol{U}^T\boldsymbol{U})\boldsymbol{R}$.

For a new design point $\boldsymbol{x}_*$, the prediction of $y(\boldsymbol{x}_*)$ is given by

$$\hat{y}(\boldsymbol{x}_*) = f(\boldsymbol{x}_*)^T\hat{\boldsymbol{\beta}} + \boldsymbol{u}(\boldsymbol{x}_*)^T\,\hat{\boldsymbol{\gamma}},$$

and the prediction variance is

$$\hat{\sigma}^2_{y(\boldsymbol{x}_*)} = \hat{\sigma}^2\boldsymbol{u}(\boldsymbol{x}_*)^T(I - \boldsymbol{\Xi}^{-1}\boldsymbol{U}^T\boldsymbol{U})\boldsymbol{R}\boldsymbol{u}(\boldsymbol{x}_*).$$

That is, $y(\boldsymbol{x}_*) \mid \boldsymbol{y} \sim \mathcal{N}(\hat{y}(\boldsymbol{x}_*), \hat{\sigma}^2_{y(\boldsymbol{x}_*)})$.

Table 2. RMSE of GPBSS for Different Numbers of Control Points

	m_2				
m_1	10	15	20	25	30
10	0.2870	0.1526	0.1392	0.1367	0.1361
15	0.2436	0.0927	0.0846	0.0820	0.0814
20	0.2347	0.0650	0.0529	0.0522	0.0524
25	0.2373	0.0625	0.0512	0.0495	0.0496
30	0.2399	0.0625	0.0493	0.0484	0.0886

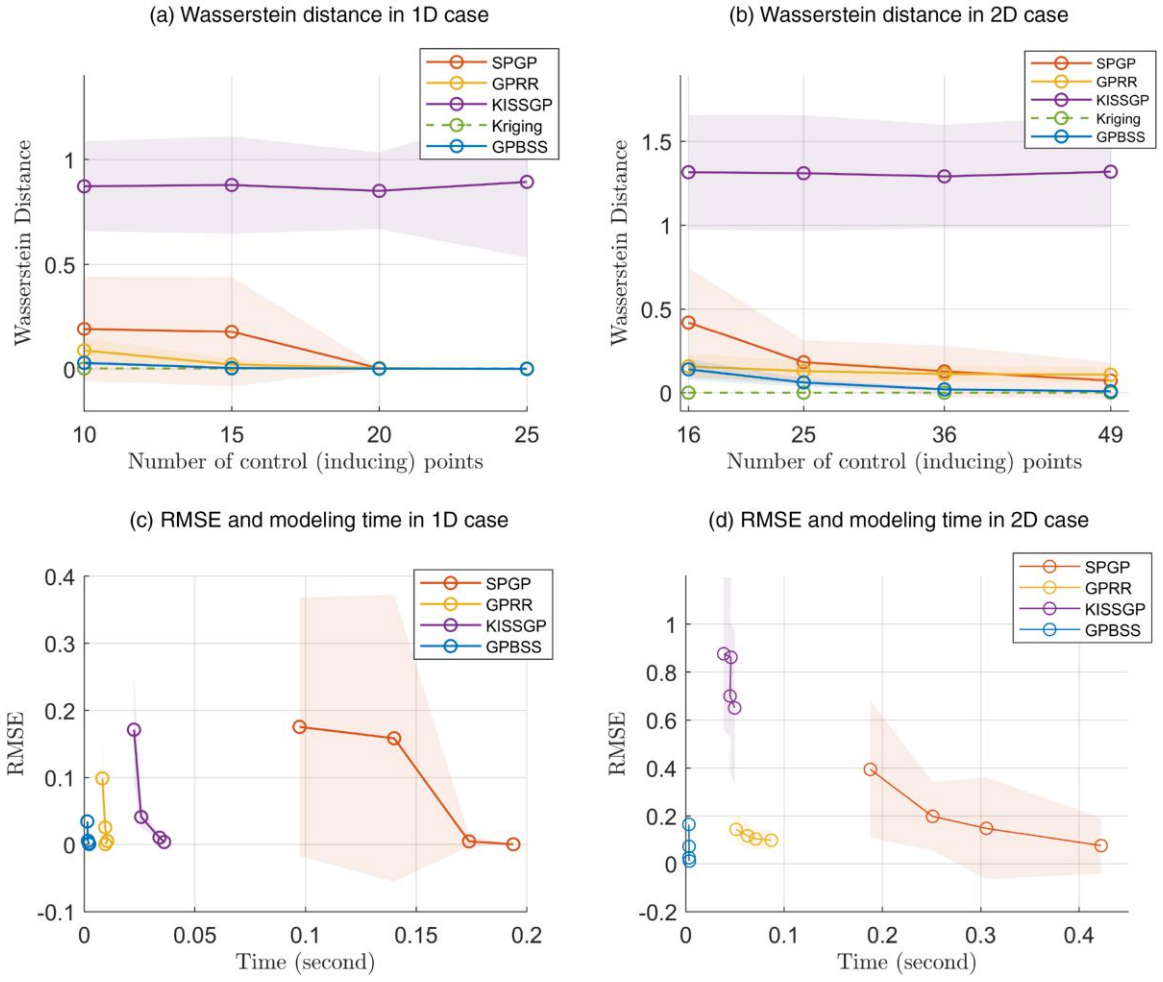
This prediction approach, based on the GPBSS model, provides efficient predictions for new design points by exploiting the structure of the B-spline basis and the sparsity it introduces into the model. Thus, the proposed method allows for scalable modeling even with large-scale data sets, ensuring computational efficiency in both the parameter estimation and model prediction phases.

Example 3. This example evaluates the uncertainty quantification capability of the proposed GPBSS method in both 1D and 2D settings and compares it against SPGP, GPRR, and KISSGP. The evaluation metric is the Wasserstein distance (Vaserstein 1969, Villani 2009, Mallasto and Feragen 2017) between the predictive distribution and the reference distribution. Training and test data are sampled from a Kriging model, with $\theta_1 = 5$ in the 1D setting and $\theta_1 = \theta_2 = 5$ in the 2D setting, which serve as roughness parameters in the Gaussian kernel $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') = \prod_{k=1}^{p} \exp\{-\theta_k^2 (x_k - x_k')^2\}$. Each experiment uses 200 training points and 500 (1D) or 2000 (2D) test points. The reference distribution at each test point is computed from the conditional distribution of the aforementioned Kriging model given the training points.

All methods are trained on the same training data, and the Wasserstein distance is evaluated for varying numbers of control (or inducing) points. Specifically, the numbers of inducing points for SPGP, GPRR, and KISSGP are 10, 15, 20, and 25 in the 1D setting and 16, 25, 36, and 49 in the 2D setting, respectively. Correspondingly, we set $m_i = 10, 15, 20,$ and 25 for GPBSS in the 1D setting and $m_i = 4, 5, 6,$ and 7 in the 2D setting where $i = 1, 2$. We also investigate the tradeoff between prediction accuracy and computational time for all methods. All experiments are repeated 30 times, and results are presented as mean curves with shaded areas indicating standard errors.

As shown in Figure 1, (a) and (b), Kriging achieves near-zero Wasserstein distance as expected because it perfectly fits all training points. Among the methods, GPBSS consistently achieves the lowest Wasserstein distance.

Figure 1. (Color online) Comparison of Prediction Accuracy and Uncertainty Quantification Across SPGP, GPRR, and KISSGP as the Number of Control (Inducing) Points Varies in 1D and 2D Cases



Moreover, GPBSS generally lies on the Pareto frontier, achieving a better trade-off between prediction accuracy and computational efficiency, as shown in Figure 1, (c) and (d).

3.5. GPBSS-Based Bayesian Optimization Framework

BO is a powerful framework for identifying the optimal design that maximizes or minimizes a given black-box function, particularly in applications where evaluations are costly or time-consuming. At the core of BO lies a surrogate model that predicts function values while simultaneously quantifying the associated uncertainty, which enables principled decision-making through acquisition functions. In this subsection, we use the proposed GPBSS model as the surrogate model and develop a GPBSS-based BO framework.

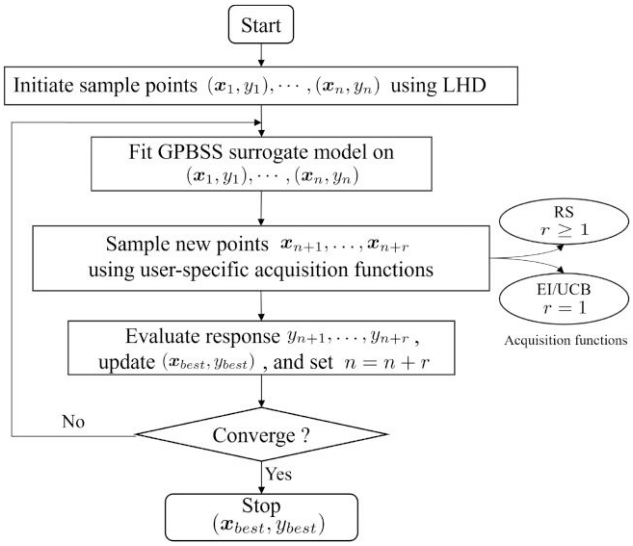
In the GPBSS-based BO, we consider various acquisition functions, including random search (RS; Wang et al. 2023), expected improvement (EI; Jones et al. 1998), and upper confidence bound (UCB; Srinivas et al. 2010). RS enables batch sampling where points are drawn using Markov chain methods based on improvement probabilities. EI and UCB, on the other hand, are single-point sampling strategies that are typically optimized using global optimization techniques, such as the genetic algorithms (Mühlenbein et al. 1991), implemented by the GA toolbox in MATLAB. This flexibility allows users to select acquisition strategies based on task characteristics. RS is ideal for parallelizable or data-intensive tasks, while EI and UCB are more effective for sequential optimization requiring precise guidance.

The process flowchart is shown in Figure 2. The BO process begins with using LHD to generate an initial sample set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. A GPBSS model is then fitted to these initial samples. In each iteration, the acquisition function selects a set of new points $\{x_{n+1}, x_{n+2}, \ldots, x_{n+r}\}$ for evaluation. The number of points r depends on the chosen strategy. RS supports batch sampling ($r \geq 1$), whereas EI and UCB typically select one point per iteration ($r = 1$). The corresponding responses $\{y_{n+1}, y_{n+2}, \ldots, y_{n+r}\}$ are then evaluated using the objective function. This resampling process iterates, with the GPBSS model being updated at each step, until the computational budget is reached.

4. Numerical Examples on Regression

In this section, we compare GPBSS with three approximation methods (SPGP, GPRR, and KISSGP). We use four different examples to evaluate the performance of each method. For a fixed sample size n, we vary the number of control points in GPBSS and the number of inducing points in GPRR, SPGP, and KISSGP. The objective is to study how well each method balances prediction accuracy and modeling time. It is worth noting that the standard Kriging model (Matheron 1963, Cressie 1993) is not included in the comparison due to its computational infeasibility with the large sample sizes used in these examples ($n \geq 50{,}000$). Similarly, the composite likelihood methods are not compared in this study because they also have prohibitive computational requirements for large-scale data sets in the numerical examples.

Figure 2. Flowchart for the GPBSS-Based Bayesian Optimization



4.1. Regression Problems

4.1.1. Regression Case 1: Six-Hump Camel Function. The six-hump camel function, a classic benchmark function, is widely used to evaluate the performance of algorithms on complex functions due to its multiple local extrema and two global minima (Lee et al. 2018). Defined in a two-dimensional space, its mathematical representation is

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2,$$

subject to $-3 \le x_1 \le 3$ and $-2 \le x_2 \le 2$.

We compared GPBSS, GPRR, SPGP, and KISSGP models on this function using data sets of 50,000 and 100,000 points. The number of control points in GPBSS is drawn from the range [5, 5] to [31, 31], that is, $m_i \in [5, 31]$, whereas GPRR, SPGP, and KISSGP used $[10, 200]$, $[10, 130]$, and $\left[10^2, 310^2\right]$ inducing points, respectively. We evaluated the models based on the RMSE calculated over 50,000 testing points and their computation time, as shown in Figure 3. Each RMSE was averaged over 100 repetitions to ensure robust results, and the standard errors were also reported. Notably, GPBSS demonstrated remarkable efficiency, handling larger 2D data sets in less time. For example, even with 1,000,000 data points in Case 1, GPBSS achieved an RMSE of 10^{-2} in less than three seconds.

4.1.2. Regression Case 2: Canopy Height Prediction. Canopy height is a key indicator of ecosystem functions such as carbon storage, biodiversity, and habitat suitability (Finney 1998, Hurtt et al. 2004, Klein et al. 2015). Understanding canopy structure is crucial for studying the impact of climate change on forest ecosystems and developing mitigation strategies. Consequently, there is an increasing demand for models that can predict canopy height. Although current light detection and ranging (LiDAR) systems are capable of mapping large areas of forest canopy, they cannot collect data at every location. This means that the canopy data have a sparse sampling design. Therefore, it is critical to build efficient and accurate regression models to predict canopy height in locations where data are lacking.

Case 2 utilizes a data set from the Bonanza Creek Experimental Forest (BCEF; Finley et al. 2022) in interior Alaska, which has 188,717 locations with collected percent tree cover (PTC) data and forest canopy height (FCH) measurements. The GPBSS, GPRR, SPGP, and KISSGP models are used to predict the canopy height. Specifically, we use 50,000 and 100,000 data points, respectively, to fit these models. The number of control points in GPBSS is drawn from the range [10, 10, 10] to [21, 21, 21], that is, $m_i \in [10, 21]$, while GPRR, SPGP, and KISSGP use $[10, 200]$, $[10, 60]$, and $\left[5^3, 104^3\right]$ inducing points, respectively. The RMSE of 50,000 testing points and the modeling time of each method are provided in Figure 4, where the standard error of the RMSEs is calculated by repeating the simulation 100 times.

4.1.3. Regression Case 3: Schwefel Function Prediction. The Schwefel function is a widely used test function for regression problems (Li et al. 2024a). This function is provided to demonstrate the performance of the proposed

Figure 3. (Color online) RMSE and Modeling Time of GPBSS, SPGP, GPRR, and KISSGP on Case 1

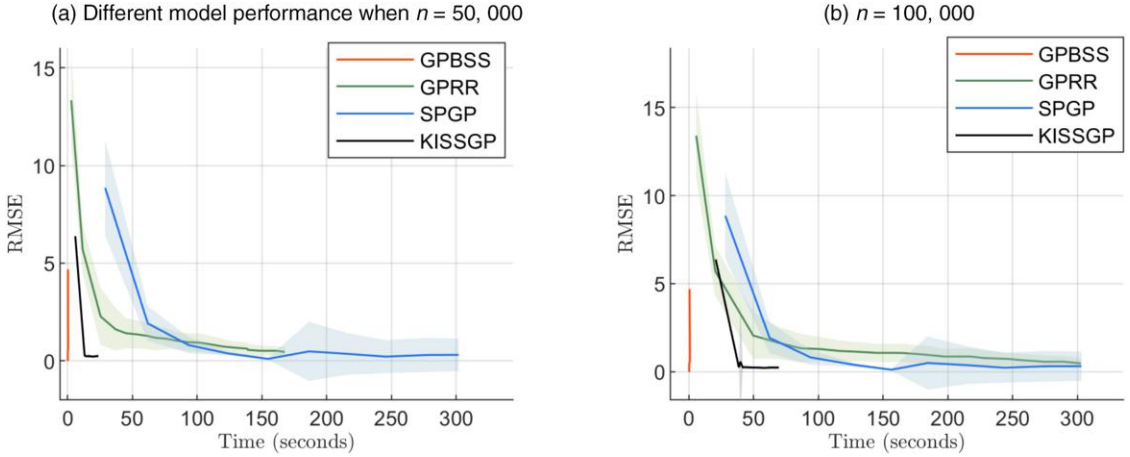
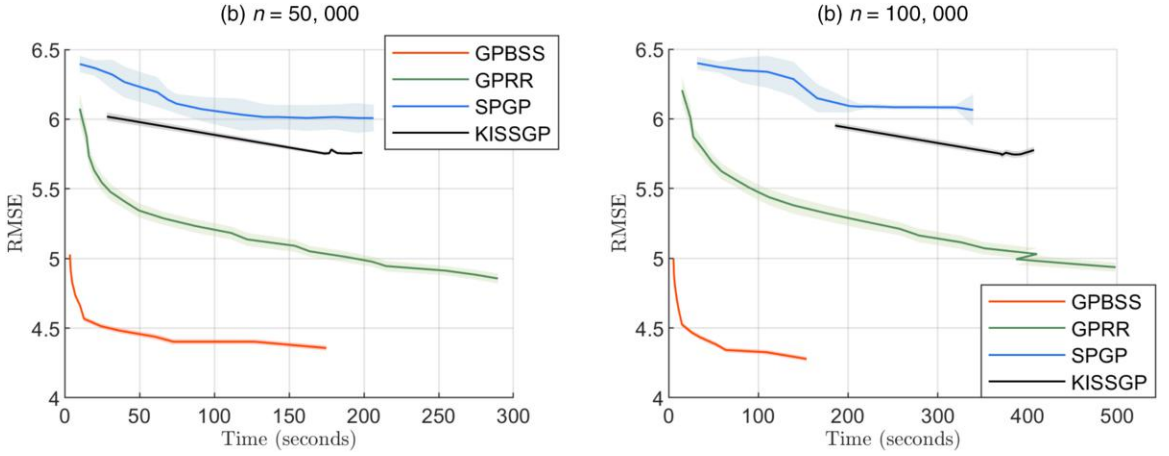


Figure 4. (Color online) RMSE and Modeling Time of GPBSS, SPGP, GPRR, and KISSGP on Case 2



method for large-scale interpolations. The Schwefel function used in this study is given by

$$f(\boldsymbol{x}) = -\sum_{i=1}^{4} x_i \sin\sqrt{|1{,}000x_i|},$$

where $-1 < x_i < 1$.

Case 3 employs the Schwefel function to evaluate the GPBSS, SPGP, KISSGP, and GPRR models. We use 100,000 and 200,000 data points, respectively, to fit these models. The number of control points in GPBSS is drawn from the range [5, 5, 5] to [11, 11, 11], that is, $m_i \in [5,11]$, whereas GPRR, SPGP, and KISSGP use $[100,1000]$, $[10,200]$, and $\left[4^4, 23^4\right]$ inducing points, respectively. The RMSE of 50,000 testing points and the modeling time of each method are provided in Figure 5, where the standard error of the RMSE is calculated by repeating the simulation 100 times.

4.1.4. Regression Case 4: Friedman Function Prediction. The Friedman function (Friedman 1991) is a widely used benchmark in regression analysis, defined as

$$f(\boldsymbol{x}) = 10 \cdot \sin(\pi x_1 x_2) + 20 \cdot (x_3 - 0.5)^2 + 10 \cdot x_4 + 5 \cdot x_5,$$

where $-1 < x_i < 1$, for $i = 1,2,\ldots,5$. This function is known for its nonlinear structure and strong interaction between variables.

In Case 4, we assess the performance of GPBSS, SPGP, and GPRR using 100,000 and 200,000 training points. It is worth noting that KISSGP is excluded from this comparison because it failed to run on such large-scale data;

Figure 5. (Color online) RMSE and Modeling Time of GPBSS, SPGP, GPRR, and KISSGP on Case 3

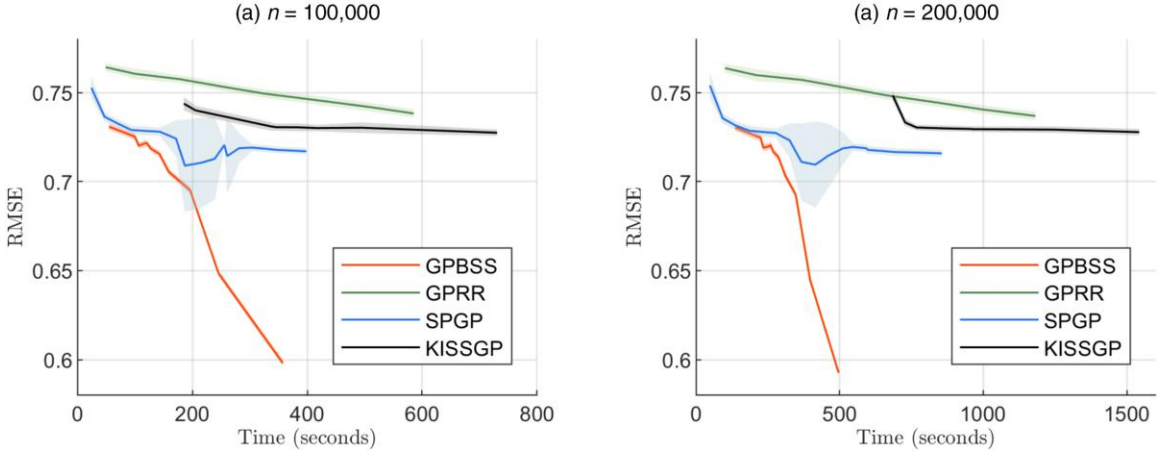
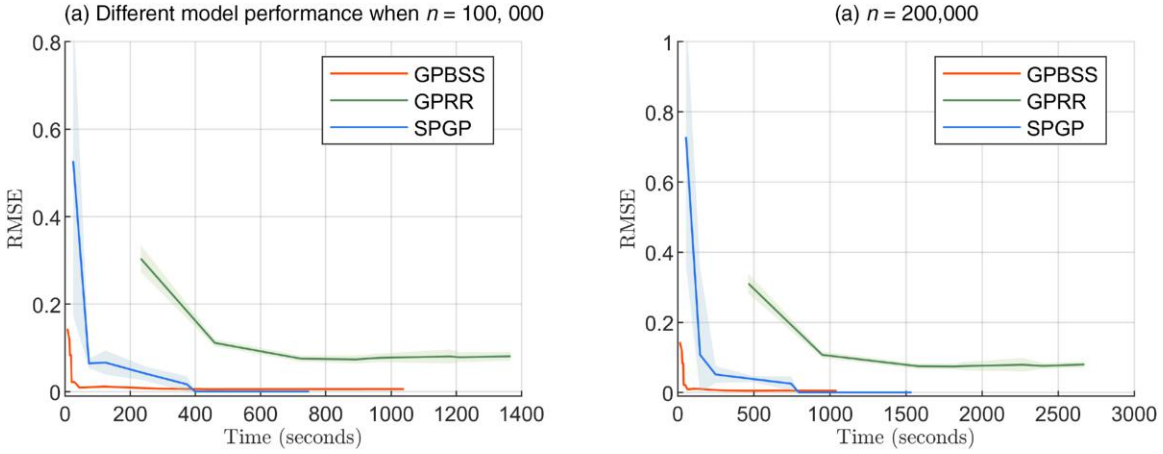


Figure 6. (Color online) RMSE and Modeling Time of GPBSS, SPGP, and GPRR on Case 4



this observation aligns with the recommendation of Wilson and Nickisch (2015) to avoid its use beyond four dimensions owing to its reliance on grid inducing points. For GPBSS, the number of control points in each dimension varies between 3 and 10 ($m_i \in [3, 10]$), whereas GPRR and SPGP use [100, 800] and [50, 250] inducing points, respectively. Model performance is evaluated using RMSE on 50,000 test points, and computational time is also reported. The standard error of the RMSE is estimated from 100 independent runs, as illustrated in Figure 6.

4.2. Result Analysis

The results from the four cases provide a comprehensive comparison of the GPBSS method against the GPRR, SPGP, and KISSGP methods.

First, compared with other methods, GPBSS achieves a better balance between computational efficiency and prediction accuracy in low-dimensional cases (i.e., Cases 1–3). Although GPBSS is not always the most accurate or time-efficient method in every scenario, it consistently performs well on the Pareto front, with the exception of Case 4 ($p = 5$), where GPBSS does not remain on the Pareto front.

Second, GPBSS provides smaller standard errors of RMSE when varying the number of control points, compared with the standard errors observed when changing the number of inducing points in GPRR and SPGP. This difference arises from the random selection of inducing points in SPGP and GPRR modeling, indicating that the way of selecting the inducing points significantly affects their performance. In contrast, GPBSS is not subject to this limitation because the knots and control points in GPBSS are evenly spaced across the design space.

Third, by comparing panels (a) and (b) in Figures 3–6, we observe that the superiority of GPBSS becomes more pronounced as the sample size n increases, because GPBSS, which has linear time complexity, is particularly suitable for large sample sizes. As n increases, the computing time of GPRR, SPGP, and KISSGP rises considerably, whereas the computing time of GPBSS remains relatively stable.

In conclusion, the GPBSS method achieves a more favorable position on the scalability–accuracy Pareto frontier in low-dimensional regression applications ($p \leq 4$). The results in the above examples clearly demonstrate that GPBSS is a valuable tool for data-intensive regression tasks in low-dimensional spaces.

5. Numerical Examples on BO

This section aims to evaluate the performance of the GPBSS-based BO on three benchmark problems. The proposed method is compared with the standard Kriging model (Redivo-Zaglia and Rodriguez 2012) in terms of optimization accuracy, computational efficiency, and predictive performance. The comparison is conducted under three widely used acquisition strategies: EI, UCB, and RS.

5.1. Optimization Problems

5.1.1. Optimization Case 1: Multi-Hills Function. The multi-hills function is a mathematical function commonly used to test and evaluate the performance of optimization algorithms (Sun et al. 2014). It has multiple local optima and a single global optimum, effectively assessing the ability of optimization algorithms to navigate

complex search spaces. The multi-hills function is defined as

$$g(x_1, x_2) = \frac{10\sin(0.05\pi x_1)}{2^{2((x_1-90)/50)^2}} + \frac{10\sin^6(0.05\pi x_2)}{2^{2((x_2-90)/50)^2}}.$$

This function contains two variables, x_1 and x_2, both of which are in the range [0, 100]. The construction of the function involves two sine functions, each of which is combined with a decay factor. It features up to 25 local maxima within the given range, as illustrated in Figure 7.

We employed SKNS and subsequently determined $m_i = 30$ as the number of control points. It is noteworthy that at the beginning of the Bayesian iterations, the number of control points significantly exceeded the number of sample points. However, this did not affect the model fitting, which is consistent with the observation in Example 2 of Section 3.

The GPBSS- and Kriging-based BOs are employed to optimize the multi-hills function, following the BO process depicted in Figure 2. BO starts with 100 initial sample points for all methods. For the RS acquisition function, 20 iterations are run with 40 resampled points per batch for a total of 800 points. To ensure a fair comparison, EI and UCB are each run for up to 800 iterations. The experiment is repeated 10 times, recording the optimization trajectories and time for both surrogate models during the iterative process.

5.1.2. Optimization Case 2: Unmanned Aerial Vehicle Controller. Unmanned aerial vehicles (UAVs; Austin 2011) are controlled either by radio remote control devices or by their own programs (Valavanis and Vachtsevanos 2014). A proportional-integral-derivative (PID) controller is often used to control the UAV's pitch angle, ensuring stable flight and precise trajectories (Blevins 2012). The parameters of the PID controller (proportional gain K_p, integral gain K_i, and differential gain K_d) are crucial for the UAV's performance (Cetin and Iplikci 2015).

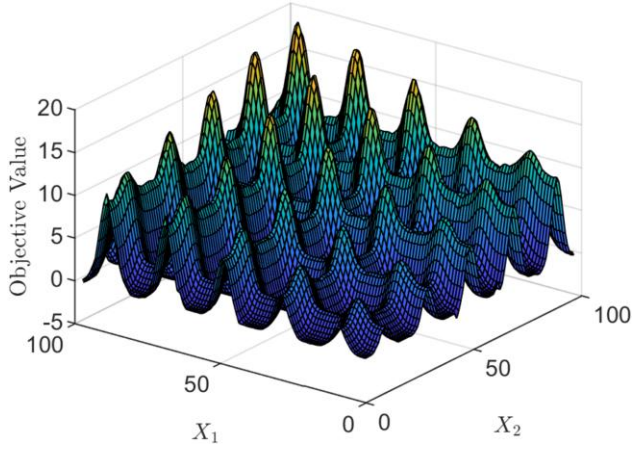
Optimizing the PID controller's response time is particularly important for UAVs because a shorter response time allows the UAV to respond more quickly and accurately to unexpected situations, thereby improving its flight safety and mission execution. The tuning expression for the PID is

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt},$$

where $u(t)$ is the control output, $e(t)$ is the system error, that is, the difference between the set value and the actual value. The proportional gain K_p adjusts the effect of the current errors. The integral gain K_i adjusts the effect of cumulative errors and eliminates steady state errors. The differential gain K_d adjusts the change rate of the errors, thereby reducing system oscillations.

In this study, we model the PID controller's response time based on its parameters K_p, K_i, and K_d, using the GPBSS (i.e., $m_i = 5$ selected by the SKNS technique) and Kriging models. Using the same acquisition settings as in the multi-hills experiment, we compare the performance of RS, EI, and UCB in the GPBSS and Kriging-based BOs. This experiment is also repeated 10 times, recording the optimization trajectories and time for both surrogate models.

Figure 7. (Color online) Surface Plot of the Multi-Hills Function



5.1.3. Optimization Case 3: Rastrigin Function. The Rastrigin function (Mühlenbein et al. 1991) is a nonconvex function that serves as a widely used benchmark in global optimization. In this case, we consider a five-dimensional Rastrigin function with a global minimum of zero, defined as

$$f(\mathbf{x}) = 50 + \sum_{i=1}^{5} \left(x_i^2 - 10\cos(2\pi x_i)\right),$$

where $-2 < x_i < 2$, to evaluate GPBSS in higher-dimensional settings ($p > 4$).

The GPBSS model is constructed using $m_i = 4$ control points selected by the SKNS technique, and its optimization performance is compared against the standard Kriging model. All variants of the GPBSS- and Kriging-based BOs are initialized with 100 initial sample points. RS is performed over 10 iterations with 40 resampled points per iteration, resulting in a total of 400 evaluations. For a fair comparison, EI and UCB are each executed for up to 400 iterations.

5.2. Result Analysis

Figure 8 depicts the optimization trajectories of the GPBSS-based and the Kriging-based BO under three acquisition functions, respectively. The lines represent the average optimized values across 10 repeated experiments, whereas the shaded areas indicate one standard deviation from the mean. Figure 9 reports runtime results for each experiment.

First, the GPBSS-based BO is faster than the Kriging-based BO in low-dimensional settings (i.e., $p \leq 4$), as shown in Figure 9, (a)–(f).This improvement is primarily due to the difference in time complexity: the Kriging model has a cubic time complexity with respect to the number of sample points, whereas the GPBSS model has a linear time complexity. However, GPBSS-based BO becomes less suitable for higher-dimensional problems

Figure 8. (Color online) Search Paths for GPBSS and Kriging Using Three Acquisition Functions (RS, EI, UCB) in Three Optimization Cases: Multi-Hills ((a)–(c)), UAV ((d)–(f)), and Rastrigin ((g)–(i))

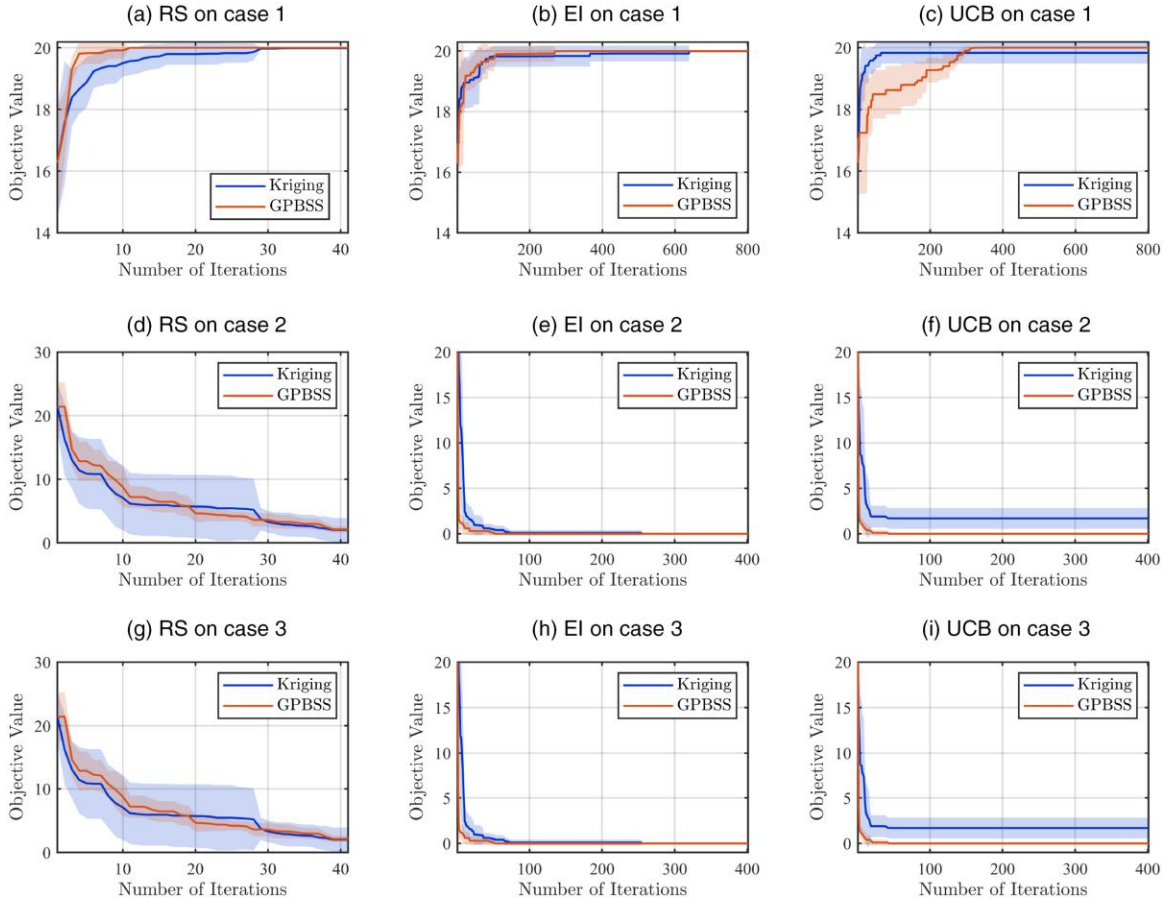
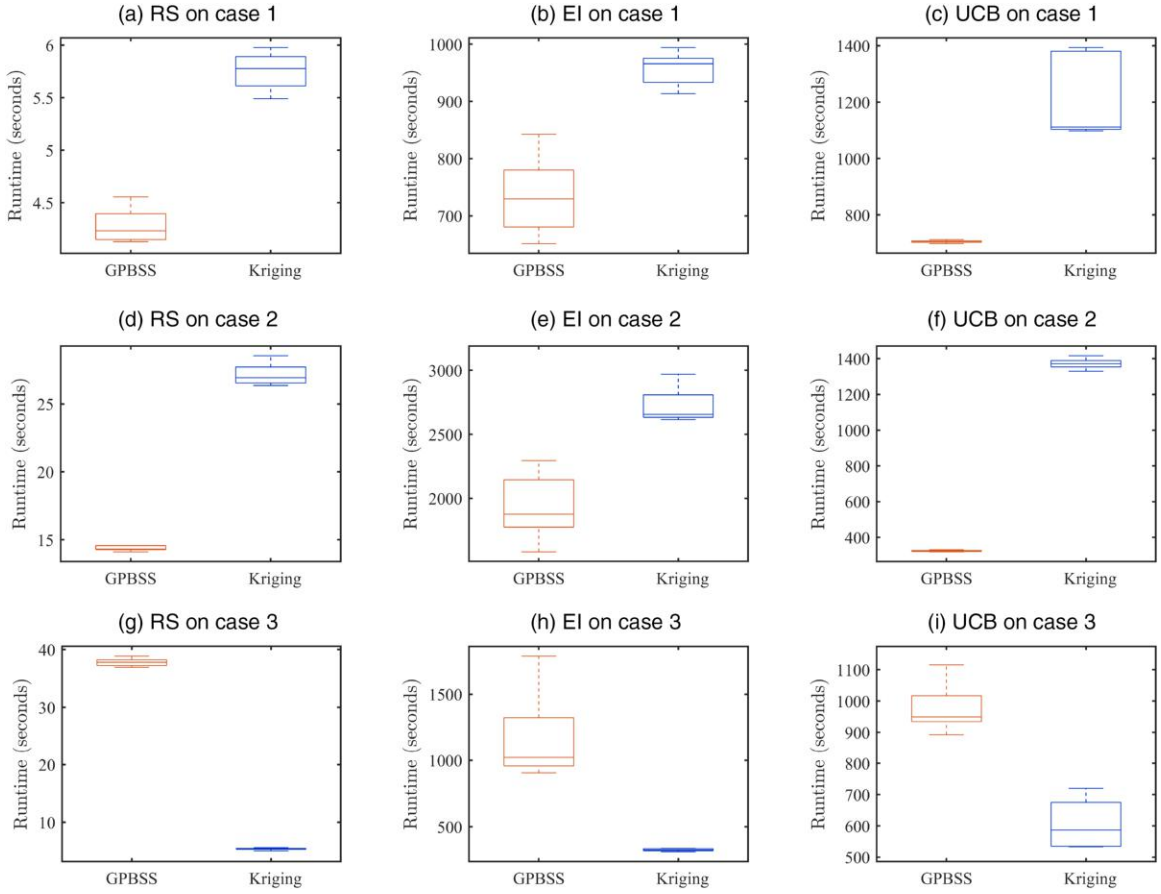


Figure 9. (Color online) Boxplots of Runtime for GPBSS and Kriging Using Three Acquisition Functions (RS, EI, and UCB)



($p > 4$). As demonstrated in Figure 9, (g)–(i), GPBSS is slower than Kriging for Case 3. This is because the number of control points (e.g., $m = 4^5$ in Case 3) used in GPBSS increases exponentially with p, resulting in a substantial computational burden. In addition, the RS acquisition function runs faster than other methods. This efficiency is largely attributed to its ability to support batch sampling, highlighting its advantages in parallel computation.

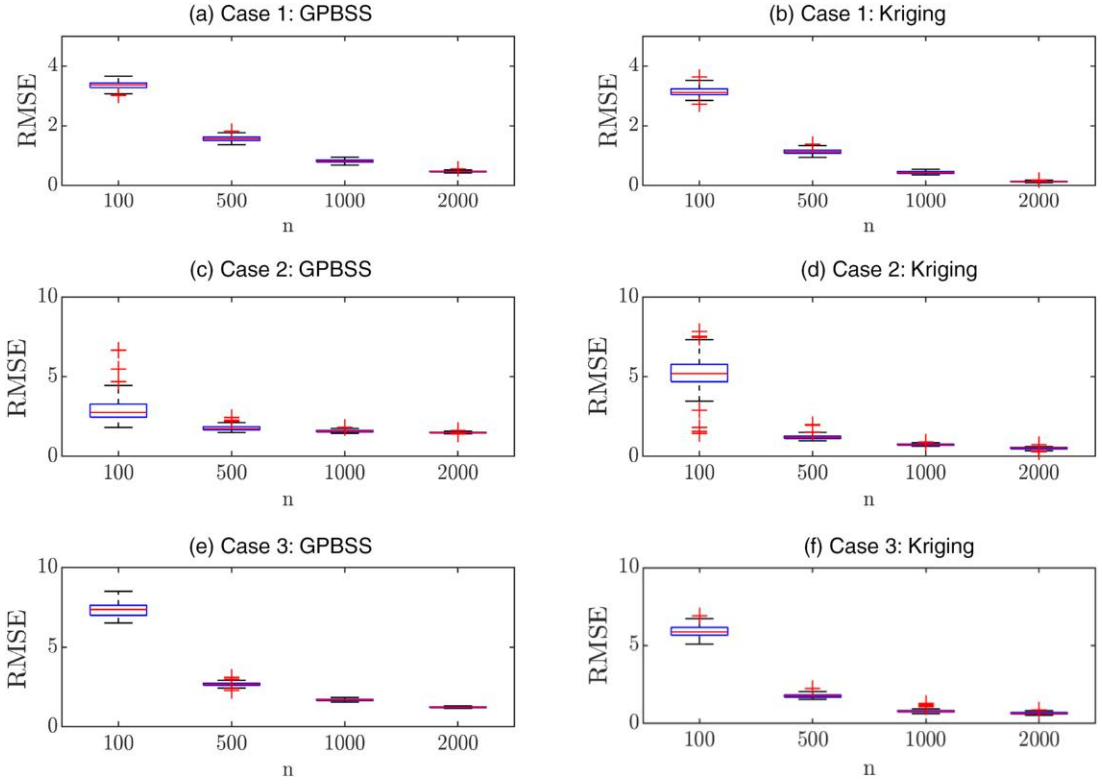
Second, the GPBSS-based method demonstrates optimization performance comparable to the Kriging-based method in most cases, as illustrated in Figure 8. The primary reason for this is that the Kriging and GPBSS models achieve comparable prediction accuracy. To illustrate this, we evaluate their RMSE in all cases by varying the sample size n from 100 to 2,000. The RMSE is calculated based on 5,000 testing points. As shown in Figure 10, the two models exhibit comparable prediction accuracy, with GPBSS yielding slightly higher RMSE. Therefore, both models can accurately approximate the objective function and converge to similar optimal solutions.

6. Conclusion

This study introduces a novel framework for scalable GP modeling that integrates B-spline surfaces within the probabilistic framework of GP, resulting in the GPBSS model. GPBSS is particularly noteworthy for its linear time complexity, making it an efficient and scalable regression tool for large-scale data sets in low-dimensional spaces. In our comparison studies in low-dimensional applications ($p \leq 4$), GPBSS achieved a favorable balance between computational efficiency and predictive accuracy compared with benchmark GP approximations such as SPGP, GPRR, and KISSGP. Numerical examples also show that the GPBSS-based BO method not only significantly outperforms the Kriging-based approach in computational speed but also achieves comparable levels of optimization accuracy, making it an efficient and reliable choice for complex optimization tasks.

Because of the limited expressiveness of B-spline surfaces in high-dimensional spaces, the GPBSS model may not scale well to regression problems with more than four input dimensions (i.e., $p > 4$). Despite this limitation, as demonstrated in numerical examples, GPBSS excels in data-intensive regression and optimization tasks in low-dimensional spaces. We believe that the proposed approach will open up new avenues for applying GPs to

Figure 10. (Color online) Comparison of RMSE for Kriging and GPBSS Across Different Sample Sizes



highly data-intensive scenarios involving low-dimensional data sampled at high rates or over long intervals, such as geostatistics (Atkinson and Lloyd 1998, Boer et al. 2001, Son et al. 2019), image processing (Salimi-Khorshidi et al. 2011, Yasarla et al. 2021), and additive manufacturing (Xu et al. 2024). Future research will focus on addressing these limitations by extending the proposed method to higher-dimensional scenarios.

Acknowledgments

The authors thank the editor, associate editor, and anonymous reviewers for constructive comments, which greatly improved the article.

Endnote

[1] See https://ww2.mathworks.cn/help/matlab/ref/peaks.html.

References

Akaike H (1998) Information theory and an extension of the maximum likelihood principle. *Selected Papers of Hirotugu Akaike* (Springer, New York), 199–213.

Akritas AG, Akritas EK, Malaschonok GI (1996) Various proofs of Sylvester's (determinant) identity. *Math. Comput. Simulations* 42(4):585–593.

Atkinson PM, Lloyd CD (1998) Mapping precipitation in Switzerland with ordinary and indicator kriging. Special issue: Spatial interpolation comparison 97. *J. Geographic Inform. Decision Anal. (Oxford)* 2(1–2):72–86.

Austin R (2011) *Unmanned Aircraft Systems: UAVS Design, Development and Deployment* (John Wiley & Sons, Hoboken, NJ).

Besag J (1975) Statistical analysis of non-lattice data. *Statistician* 24(3):179–195.

Bilionis I, Zabaras N (2012) Multi-output local Gaussian process regression: Applications to uncertainty quantification. *J. Comput. Phys.* 231(17):5718–5746.

Blevins TL (2012) PID advances in industrial control. *IFAC Proc. Volumes* 45(3):23–28.

Boer EP, de Beurs KM, Hartkamp AD (2001) Kriging and thin plate splines for mapping climate variables. *Internat. J. Appl. Earth Observation Geoinformation* 3(2):146–154.

Cao J, Genton MG, Keyes DE, Turkiyyah GM (2021) Sum of Kronecker products representation and its Cholesky factorization for spatial covariance matrices from large grids. *Comput. Statist. Data Anal.* 157(1):107165.

Caragea PC, Smith RL (2007) Asymptotic properties of computationally efficient alternative estimators for a class of multivariate normal models. *J. Multivariance Anal.* 98(7):1417–1440.

Cetin M, Iplikci S (2015) A novel auto-tuning pid control mechanism for nonlinear systems. *ISA Trans.* 58(September):292–308.

Che Y, Ma Y, Li Y, Ouyang L (2024) A novel active-learning kriging reliability analysis method based on parallelized sampling considering budget allocation. *IEEE Trans. Reliability* 73(1):589–601.

Chen Z, Fan J, Wang K (2023) Multivariate Gaussian processes: Definitions, examples and applications. *Metron* 81(2):145–180.

Cressie N (1993) *Statistics for Spatial Data* (John Wiley & Sons, Hoboken, NJ).

Cressie N, Johannesson G (2008) Fixed rank kriging for very large spatial data sets. *J. Roy. Statist. Soc. Ser. B* 70(1):209–226.

Eidsvik J, Shaby BA, Reich BJ, Wheeler M, Niemi J (2014) Estimation and prediction in spatial models with block composite likelihoods. *J. Comput. Graphical Statist.* 23(2):295–315.

Finley AO, Datta A, Banerjee S (2022) spNNGP R package for nearest neighbor Gaussian process models. *J. Statist. Software* 103(5):1–40.

Finney MA (1998) *FARSITE: Fire Area Simulator-model development and evaluation*, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT.

Fricker TE, Oakley JE, Urban NM (2013) Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics* 55(1):47–56.

Friedman JH (1991) Multivariate adaptive regression splines. *Ann. Statist.* 19(1):1–67.

Fuhg JN, Fau A, Nackenhorst U (2021) State-of-the-art and comparative review of adaptive sampling methods for kriging. *Arch. Computational Methods Engrg.* 28:2689–2747.

Furrer R, Genton MG, Nychka D (2006) Covariance tapering for interpolation of large spatial data sets. *J. Comput. Graph. Statist.* 15(3):502–523.

Ghosh M, Li Y, Zeng L, Zhang Z, Zhou Q (2021) Modeling multivariate profiles using Gaussian process-controlled B-splines. *IISE Trans.* 53(7):787–798.

Gu C (2013) *Smoothing Spline ANOVA Models*, vol. 297 (Springer, Berlin).

Guinness J (2018) Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics* 60(4):415–429.

He X, Shen L, Shen Z (2001) A data-adaptive knot selection scheme for fitting splines. *IEEE Signal Processing Lett.* 8(5):137–139.

Heagerty PJ, Lele SR (1998) A composite likelihood approach to binary spatial data. *J. Amer. Statist. Assoc.* 93(443):1099–1111.

Hensman J, Fusi N, Lawrence ND (2013) Gaussian processes for big data. Nicholson AE, Smyth P, ed. *Proc. Twenty-Ninth Conf. Uncertainty Artificial Intelligence (UAI'13)* (AUAI Press, Arlington, TX), 282–290.

Hristopulos DT (2024) Non-separable covariance kernels for spatiotemporal Gaussian processes based on a hybrid spectral method and the harmonic oscillator. *IEEE Trans. Inform. Theory* 70(2):1268–1283.

Hurtt GC, Dubayah R, Drake J, Moorcroft PR, Pacala SW, Blair JB, Fearon MG (2004) Beyond potential vegetation: Combining lidar data and a height-structured model for carbon studies. *Ecological Appl.* 14(3):873–883.

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13:455–492.

Katzfuss M (2017) A multi-resolution approximation for massive spatial data sets. *J. Amer. Statist. Assoc.* 112(517):201–214.

Katzfuss M, Guinness J (2021) A general framework for Vecchia approximations of Gaussian processes. *Statist. Sci.* 36(1):124–141.

Katzfuss M, Guinness J, Gong W, Zilber D (2020) Vecchia approximations of Gaussian-process predictions. *J. Agricultural Biological Environment. Statist.* 25(3):383–414.

Kaufman CG, Schervish MJ, Nychka DW (2008) Covariance tapering for likelihood-based estimation in large spatial data sets. *J. Amer. Statist. Assoc.* 103(484):1545–1555.

Kleijnen JPC (2009) Kriging metamodeling in simulation: A review. *Eur. J. Oper. Res.* 192(3):707–716.

Klein T, Randin C, Körner C (2015) Water availability predicts forest canopy height at the global scale. *Ecology Lett.* 18(12):1311–1320.

Lee JH, Song JY, Kim DW, Kim JW, Kim YJ, Jung SY (2018) Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines. *IEEE Trans. Industry Electronics* 65(2):1791–1798.

Li Y, Li Y, Wang D (2025) Periodic Gaussian process controlled B-spline for scalable modeling of irregularly spaced signals. *IEEE Trans. Inform. Theory* 71(10):7842–7855.

Li Y, Zhou Q, Jiang W, Tsui KL (2024a) Optimal composite likelihood estimation and prediction for distributed Gaussian process modeling. *IEEE Trans. Pattern Anal. Machine Intelligence* 46(2):1134–1147.

Li Y, Zhang Y, Wu J, Xie M (2024b) Regularized periodic Gaussian process for nonparametric sparse feature extraction from noisy periodic signals. *IEEE Trans. Automation Sci. Engrg.* 22:3011–3020.

Lindsay BG (1988) Composite likelihood methods. *Contemporary Math.* 80(1):221–239.

Lindsay BG, Yi GY, Sun J (2011) Issues and strategies in the selection of composite likelihoods. *Statist. Sinica* 21(1):71–105.

Liu H, Ong YS, Shen X, Cai J (2020) When Gaussian process meets big data: A review of scalable GPS. *IEEE Trans. Neural Networks Learn. Systems* 31(11):4405–4423.

Ma P, Huang JZ, Zhang N (2015) Efficient computation of smoothing splines via adaptive basis sampling. *Biometrika* 102(3):631–645.

Mallasto A, Feragen A (2017) Learning from uncertain curves: The 2-Wasserstein metric for Gaussian processes. Guyon I, von Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Red Hook, NY), 5660–5670.

Matheron G (1963) Principles of geostatistics. *Econom. Geology* 58(8):1246–1266.

Morris MD, Mitchell TJ, Ylvisaker D (1993) Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction. *Technometrics* 35(3):243–255.

Mühlenbein H, Schomisch M, Born J (1991) The parallel genetic algorithm as function optimizer. *Parallel Comput.* 17(6–7):619–632.

Park JS (1994) Optimal latin-hypercube designs for computer experiments. *J. Statist. Planning Inference* 39(1):95–111.

Powell JL (1994) Estimation of semiparametric models. *Handbook Econom.* 4:2443–2521.

Prautzsch H, Boehm W, Paluszny M (2002) *Bézier and B-Spline Techniques* (Springer Science & Business Media, Boston).

Rasmussen CE, Williams CKI (2006) *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA).

Redivo-Zaglia M, Rodriguez G (2012) SMT: A matlab toolbox for structured matrices. *Numerical Algorithms* 59(4):639–659.

Ruppert D, Wand MP, Carroll RJ (2003) *Semiparametric Regression* (Cambridge University Press, New York).

Sacks J, Schiller SB, Welch WJ (1989) Designs for computer experiments. *Technometrics* 31(1):41–47.

Salimi-Khorshidi G, Nichols T, Smith S, Woolrich M (2011) Using Gaussian-process regression for meta-analytic neuroimaging inference based on sparse observations. *IEEE Trans. Medical Imaging* 30(7):1401–1416.

Sang H, Huang JZ (2012) A full scale approximation of covariance functions for large spatial data sets. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 74(1):111–132.
Santner TJ, Williams BJ, Notz WI (2003) The design and analysis of computer experiments. *Series in Statistcs* (Springer, New York):1–13.
Schulz E, Speekenbrink M, Krause A (2018) A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psych.* 85(August):1–16.
Sherman J, Morrison WJ (1950) Adjustment of an Inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist.* 21(1):124–127.
Snelson E, Ghahramani Z (2006) Sparse Gausian processes using pseudo-inputs. *Adv. Neural Inform. Processing Systems* 18:1257–1264.
Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. Pereira F, Burges CJ, Bottou L, Weinberger KQ, ed. *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY), 2951–2959.
Son PW, Rhee JH, Hwang J, Seo J (2019) Universal kriging for Loran ASF map generation. *IEEE Trans. Aerospace Electronic Systems* 55(4):1828–1842.
Srinivas N, Krause A, Kakade SM, Seeger M (2010) Gaussian process optimization in the bandit setting: No regret and experimental design. Fürnkranz J, Joachims T, ed. *Proc. 27th Internat. Conf. Machine Learn.* (Omnipress, Madison, WI), 1015–1022.
Stein ML (2008) A modeling approach for large spatial data sets. *J. Korean Statist. Soc.* 37(1):3–10.
Stein ML, Chi Z, Welty LJ (2004) Approximating likelihoods for large spatial data sets. *J. Roy. Statist. Soc. Ser. B* 66(2):275–296.
Su G, Peng L, Hu L (2017) A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Structural Safety* 68(September):97–109.
Sun L, Hong L, Hu Z (2014) Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Oper. Res.* 62(6):1416–1438.
Sun J, Zhou S, Veeramani D, Liu K (2024) Prediction of condition monitoring signals using scalable pairwise Gaussian processes and Bayesian model averaging. *IEEE Trans. Automation Sci. Engrg.* 22:2746–2757.
Titsias M (2009) Variational learning of inducing variables in sparse Gaussian processes. *Proc. 12th Internat. Conf. Artificial Intelligence Statist.*, vol. 5 (PMLR, New York), 567–574.
Valavanis KP, Vachtsevanos GJ (2014) *Handbook of Unmanned Aerial Vehicles* (Springer, Berlin).
Varin C, Reid N, Firth D (2011) An overview of composite likelihood methods. *Statist. Sinica* 21(1):5–42.
Vaserstein LN (1969) Markov processes over denumerable products of spaces describing large systems of automata. *Problemy Peredachi Informatsii* 5(3):47–52.
Vecchia AV (1988) Estimation and model identification for continuous spatial processes. *J. Roy. Statist. Soc. Ser. B (Methodological)* 50(2):297–312.
Villani C (2009) *Optimal Transport: Old and New* (Springer, Berlin).
Wahba G (1990) *Spline Models for Observational Data* (SIAM, Philadelphia).
Wang X, Hong LJ, Jiang Z, Shen H (2023) Gaussian process-based random search for continuous optimization via simulation. *Oper. Res.* 73(1):1–23.
Wilson A, Nickisch H (2015) Kernel interpolation for scalable structured Gaussian processes (KISS-GP). Bach F, Blei DM, ed. *Proc. 32nd Internat. Conf. Machine Learn.* (PMLR, Bethesda, MD), 1775–1784.
Xiong S (2021) The reconstruction approach: From interpolation to regression. *Technometrics* 63(2):225–235.
Xu R, Huang S, Song Z, Gao Y, Wu J (2024) A deep mixed-effects modeling approach for real-time monitoring of metal additive manufacturing process. *IISE Trans.* 56(9):945–959.
Yasarla R, Sindagi VA, Patel VM (2021) Semi-supervised image deraining using Gaussian processes. *IEEE Trans. Image Processing* 30:6570–6582.
Yuan Y, Chen N, Zhou S (2013) Adaptive B-spline knot selection using multi-resolution basis set. *IIE Trans.* 45(12):1263–1277.
Zhang Y, Apley DW, Chen W (2020) Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Sci. Rep.* 10(1):4924.