Received: 11 December 2018 | Revised: 5 June 2019 | Accepted: 12 June 2019

DOI: 10.1002/nme.6145

RESEARCH ARTICLE

WILEY

A new surrogate modeling method combining polynomial chaos expansion and Gaussian kernel in a sparse Bayesian learning framework

Yicheng Zhou | **Zhenzhou Lu** | **Kai Cheng**

School of Aeronautics, Northwestern Polytechnical University, Xi'an, China

Correspondence
Zhenzhou Lu, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China.
Email: zhenzhoulu@nwpu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 51775439; Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University, Grant/Award Number: CX201935

Summary

Surrogate modeling techniques have been increasingly developed for optimization and uncertainty quantification problems in many engineering fields. The development of surrogates requires modeling high-dimensional and nonsmooth functions with limited information. To this end, the hybrid surrogate modeling method, where different surrogate models are combined, offers an effective solution. In this paper, a new hybrid modeling technique is proposed by combining polynomial chaos expansion and kernel function in a sparse Bayesian learning framework. The proposed hybrid model possesses both the global characteristic advantage of polynomial chaos expansion and the local characteristic advantage of the Gaussian kernel. The parameterized priors are utilized to encourage the sparsity of the model. Moreover, an optimization algorithm aiming at maximizing Bayesian evidence is proposed for parameter optimization. To assess the performance of the proposed method, a detailed comparison is made with the well-established PC-Kriging technique. The results show that the proposed method is superior in terms of accuracy and robustness.

KEYWORDS

Gaussian kernel, hybrid model, polynomial chaos expansion, sparse Bayesian learning

1 | INTRODUCTION

Behavior prediction of engineering systems and mathematical models is often affected by diverse types of uncertainties,[1] which lead to uncertain performance. In this context, uncertainty analysis has been widely used to help decision makers understand the degree of confidence in the decision they made and assess the corresponding risk.[2] However, because computer simulation is time consuming, the uncertainty analysis issue cannot be addressed by a classical method such as direct Monte Carlo simulation. Thus, more advanced statistical methods have to be employed. Among them, surrogate modeling techniques have been increasingly adopted as powerful methods to emulate the output of computationally expensive models.

Polynomial chaos expansion (PCE) offers a number of benefits compared to other surrogate models (also known as meta-models),[3,4] and the use of nonintrusive PCE has received much attention in uncertainty quantification, with applications in computational fluid dynamics problems,[5,6] sensitivity analysis,[7,8] structural reliability analysis,[9] moment estimation,[10] and model dimensionality reduction.[11] The key concept in PCE is to expand the model response onto basis made of multivariate polynomials; they are orthogonal with respect to the joint distribution of the input variables. In this setting, characterizing the probability density function (PDF) is equivalent to evaluating the polynomial chaos coefficients. Although PCE has been proven to be powerful in a wide range of applications, a major drawback restricting the application of PCE is that the computational cost of model response grows exponentially with both the number of input variables and the total expansion degree of PCE. This drastic increase in computational cost with the number of inputs is referred to as the curse of dimensionality.[12] To avoid this issue, Blatman and Sudret[13] employed the least angle regression (LAR) algorithm[14] to approximate the original model by a small subset of the polynomial basis functions. This algorithm is available from UQLab software introduced by Marelli and Sudret.[15] On the sparsity assumption, many other techniques are established in the context of compressed sensing.[16-19] PCE is more suitable for capturing the global trend. However, PCE is often not adequate for capturing local accuracy in the close neighborhood of the sample points. The challenge in accomplishing an exact fit has inspired researchers to explore the kernel-based surrogate modeling techniques, which make use of local information related to each training point and combine this information to define the overall surrogate model. Kernel-based surrogate modeling techniques offer important advantages over PCE, such as the ease of extending the estimated function to higher dimensions and the representation of highly nonlinear functional relationships. Kernel-based surrogate modeling methods typically make use of local information related to each training data point and combine this information to define the overall surrogate model. Support vector regression,[20,21] neural network,[22] and Kriging (also known as Gaussian process modeling)[23-25] are popular kernel-based surrogate modeling techniques. Recently, Schöbi et al[26] and Kersaudy et al[27] have developed a new surrogate modeling technique called PC-Kriging. It combines the sparse PCE obtained by the LAR technique and Gaussian kernels to obtain a hybrid surrogate model. Because of the lack of prior knowledge about the output, leave-one-out cross-validation (LOOCV) is employed as an object function to select optimal regression functions for Kriging from the set of orthonormal multivariate polynomials determined by the LAR technique. In fact, LOOCV is an approximation method used to estimate the actual generalization error, and the overfitting phenomenon may influence its basis selection.[26]

In this paper, we are concerned with the surrogate modeling technique under the sparse Bayesian learning (SBL) framework. The classical SBL is first developed for the relevance vector machine (RVM) by Tipping,[28] which is a probabilistic model assuming that the unknown regression coefficients of basis functions are independent variables with a zero-mean Gaussian distribution. The variance of each coefficient is treated as a hyperparameter that is learned from the observations by maximizing a marginal likelihood function instead of time-consuming cross-validation for model selection. Theoretical analysis in the work of Wipf and Rao[29] demonstrated that the likelihood function of the hyperparameters of the RVM achieves a global maximum at the sparsest solution and that the local maxima are also sparse. Nevertheless, it assumes that the basis function is a predetermined kernel. Consequently, although the RVM[28] has demonstrated a promising performance in many applications compared with other learning methods in the literature,[30-32] its efficiency is highly dependent on the choice of an appropriate kernel. With respect to surrogate-based modeling of complex problems, one of the common practices is to construct several surrogate models and select the one with the best performance. However, this deterministic approach falls short of fully exploiting the resources available for surrogate modeling. To solve the problem, we propose a probabilistic model to simultaneously account for basis functions corresponding to multivariate orthogonal polynomials and kernel functions. Since the new surrogate model involves a hybrid structure, it provides an extension of the classical SBL. The objectives of the proposed surrogate modeling method can be viewed as comprising two parts, ie, to accomplish reasonable local accuracy in the neighborhood of the sample points and to capture the global trend of the functional variation. Therefore, the proposed method gains knowledge on the hybrid structure to simultaneously capture the global and local accuracies. Additionally, the probabilistic model used in this paper also could be adapted to a mixture of more types of basis functions; however, this paper emphasizes on PCE and the Gaussian kernel.

On the other hand, the RVM[28] only considered the general kernel that has unified kernel width. This is equivalent to making a basic assumption that the response is represented below a certain frequency. However, this may not hold in many cases. To overcome the limitation, an adaptive elliptical kernel that generalizes the feature to input space is utilized to improve approximation ability. A stepwise optimization algorithm aiming at maximizing the marginal likelihood function is then employed to optimize the kernel width vector. Three benchmark examples and a wing box structure are used for validating and assessing the performance of the proposed method. A detailed comparison is made with the well-established PC-Kriging surrogate modeling technique.[26,27]

The rest of the work is organized as follows. In Section 2, PCE and conventional kernel functions are first recalled. A hybrid model based on SBL is introduced concisely in Section 3. Section 4 describes the optimization algorithm of marginal likelihood maximization to infer hyperparameters and tune the adaptive kernel widths. The performance of the proposed method is assessed on several examples in Section 5. Section 6 presents conclusions.

2 | REVIEW OF PCE AND KERNEL FUNCTION

2.1 | Polynomial chaos expansion

Consider a scalar response model $Y = g(\boldsymbol{X})$, where $\boldsymbol{X} = \{X_1, \dots, X_n\}$ is a random vector. The joint PDF of $\boldsymbol{X}$ is $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{d=1}^{n} f_{X_d}(x_d)$, where $f_{X_d}(x_d)$ is the marginal PDF of X_i. The model response can be expanded as follows:

$$g(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} \beta_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ (with $\alpha_i \geq 0$) is an n-dimensional index, $\{\beta_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^n\}$ are deterministic polynomial chaos coefficients to be computed, and $\{\psi_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^n\}$ are multivariate orthonormal polynomials. The independence of inputs allows us to construct these multivariate polynomials as a tensor product of univariate orthonormal polynomials with respect to the PDF $f_{X_d}(x_d)$. For instance, for standard Gaussian inputs, the orthogonal polynomials are multivariate Hermite polynomials. In the case of uniformly distributed inputs, multivariate Legendre polynomials are selected. The same formulation is possible for other distribution types of inputs, which is known as generalized PCE.[33] The standard families of univariate orthonormal polynomials used to construct PCE are shown in Table 1.[3,34]

Equation (1) is commonly truncated by prescribing the total expansion degree p for computation. As a result, the number of polynomials retained in the PCE is given by

$$P = \binom{n+p}{p} = \frac{(n+p)!}{n!p!}. \tag{2}$$

Then, the model response Y is approximated as follows:

$$Y = g_{\mathcal{A}^{p,n}}(\boldsymbol{X}) + \varepsilon_{\text{trun}} = \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{p,n}} \beta_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) + \varepsilon_{\text{trun}} \quad \mathcal{A}^{p,n} = \left\{\boldsymbol{\alpha} \in \mathbb{N}^n : |\boldsymbol{\alpha}| \leq p\right\}, \tag{3}$$

where $\varepsilon_{\text{trun}}$ is the truncation error introduced by truncating the expansion to a finite number of terms. In this case, P is the cardinality of $\mathcal{A}^{p,n}$, ie, $P = \text{Card}\left(\mathcal{A}^{p,n}\right)$. Equation (3) is called the full PCE with total expansion degree p. In practice, the best degree p is difficult to select due to the lack of prior knowledge of the features of the model response of interest. Thus, it is fundamental to assess the quality of the built surrogate models in order to select the one with the best accuracy. To extract a global quality, the so-called generalization error is often considered. However, the true model evaluation $g(\boldsymbol{x})$ at an arbitrary realization $\boldsymbol{x}$ of $\boldsymbol{X}$ is undetermined since the computational model is expensive to evaluate. LOOCV[35] is thus used to estimate the generalization error using an available training set and determine the best degree p among several ones to achieve convergence of the global quality. Equation (3) can be rewritten using a vector notation, as follows:

$$g_{\mathcal{A}^{p,n}}(\boldsymbol{X}) = \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\psi}(\boldsymbol{X}), \tag{4}$$

where $\boldsymbol{\beta} = \left\{\beta_{\boldsymbol{\alpha}_1}, \dots, \beta_{\boldsymbol{\alpha}_P}\right\}^{\mathrm{T}}$ and $\boldsymbol{\psi}(\boldsymbol{X}) = \left\{\psi_{\boldsymbol{\alpha}_1}(\boldsymbol{X}), \dots, \psi_{\boldsymbol{\alpha}_P}(\boldsymbol{X})\right\}^{\mathrm{T}}$ gather the polynomial chaos coefficients and basis polynomials in Equation (4), respectively.

2.2 | Kernel function

Kernel functions are actually a projection function. The function projects the original linearly or nonlinearly learning data into high-dimensional feature space, where all of the data can be presented linearly. Consider the n-dimensional

TABLE 1 Type of univariate orthogonal polynomials associated with the random variables with the specified probability density function

Random variable	Polynomial type	Support
Gaussian	Hermite	$(-\infty,\infty)$
Gamma	Laguerre	$[0,\infty)$
Beta	Jacobi	[a,b]
Uniform	Legendre	[a,b]
Poisson	Charlier	{0,1,2,...,}
Binomial	Krawtchouk	{0,1,2,...,n}
Negative binomial	Meixner	{0,1,2,...,}
Hypergeometric	Hahn	{0,1,2,...,n}

TABLE 2 Type of kernel functions

Kernel function	Expression
Gaussian kernel function	$k\left(\boldsymbol{X},\boldsymbol{X}'\right)=\exp\left(-\frac{\lVert X-X'\rVert^2}{2l^2}\right)$
Exponential kernel function	$k\left(\boldsymbol{X},\boldsymbol{X}'\right)=\exp\left(-\frac{\lVert X-X'\rVert}{2\sigma^2}\right)$
Linear kernel function	$k(\boldsymbol{X},\boldsymbol{X}')=\boldsymbol{X}\boldsymbol{X}'$
Multilayer perception kernel function	$k(\boldsymbol{X},\boldsymbol{X}')=\tan\left(\gamma\boldsymbol{X}\boldsymbol{X}'+\theta\right)$
Splines kernel function	$k\left(\boldsymbol{X},\boldsymbol{X}'\right)=1+\left\langle\boldsymbol{X},\boldsymbol{X}'\right\rangle+\frac{1}{2}\left\langle\boldsymbol{X},\boldsymbol{X}'\right\rangle\min\left(\boldsymbol{X},\boldsymbol{X}'\right)-\frac{1}{6}\left(\min\left(\boldsymbol{X},\boldsymbol{X}'\right)\right)^3$

experimental design $\mathcal{X}=\left\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}\right\}^{\mathrm{T}}$ containing N samples $\boldsymbol{x}^{(i)}=\left\{x_1^{(i)},\ldots,x_n^{(i)}\right\}$ $(i=1,\ldots,N)$, a regression model would be the linearity-weighted sum of kernel functions, ie,

$$g_R(\boldsymbol{X})=\sum_{j=1}^{N}\omega_j k\left(\boldsymbol{X},\boldsymbol{x}^{(j)}\right), \tag{5}$$

where $\boldsymbol{\omega}=\{\omega_1,\ldots,\omega_N\}^{\mathrm{T}}$ is the vector of the kernel coefficient of the linear model. The kernel function $k(\boldsymbol{X},\boldsymbol{x}^{(j)})$ conducts the similarity measurement between support vector $\boldsymbol{x}^{(j)}$ and random vector $\boldsymbol{X}$ in input space. A kernel function must satisfy Mercer's conditions.[36] In Table 2, we list several commonly used kernel functions.

Generally, in machine learning, using the Gaussian kernel will yield better prediction performance.[37] It is a local kernel that just responds to the near neighbor of inputs and has strong interpolation ability.[38] However, the kernel width is invariant for all inputs in the standard RVM. Therefore, in the response domain containing both high and low frequencies of response variation, the underfitting phenomenon would occur in the domain of high frequencies, whereas the learning would suffer from overfitting in the subdomain of low frequencies if using a large width l.[39] To tackle this problem, the elliptical Gaussian kernel is employed in this work, ie,

$$k_e\left(\boldsymbol{X},\boldsymbol{x}^{(j)}\right)=\exp\left(-\sum_{d=1}^{n}\frac{\left(x_d^{(j)}-X_d\right)^2}{2l_d^2}\right), \tag{6}$$

where $\boldsymbol{l}=\{l_1,\ldots,l_n\}$ is a vector of kernel width. In this case, the kernel widths $l_d(d=1,\ldots,n)$ represent the range in any spatial direction. Assuming, for instance, that certain values are less significant for the response, then the corresponding kernel width will be very large compared to the other kernel widths.

3 | COMBINATION OF PCEs AND ELLIPTICAL GAUSSIAN KERNEL

3.1 | SBL for the surrogate model

This section introduces the proposed surrogate modeling technique that combines the good characteristic of both PCE and the Gaussian kernel based on SBL (PC-GK-SBL). The hybrid model has the following formulation:

$$g_{\mathrm{PG}}(\boldsymbol{X})=\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\psi}(\boldsymbol{X})+\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{k}_e(\boldsymbol{X}), \tag{7}$$

where $\boldsymbol{k}_e(\boldsymbol{X})=\{k_e(\boldsymbol{X},\boldsymbol{x}^{(1)}),\ldots,k_e(\boldsymbol{X},\boldsymbol{x}^{(N)})\}^{\mathrm{T}}$. After running the model at the experimental design $\mathcal{X}$, the responses of the original computational model are gathered into the vector $\mathcal{Y}=\left\{y^{(1)},\ldots,y^{(N)}\right\}^{\mathrm{T}}$. In this case, the observations can be denoted as a compact form, ie,

$$\mathcal{Y}\approx\boldsymbol{\Psi}\boldsymbol{\beta}+\boldsymbol{K}\boldsymbol{\omega}, \tag{8}$$

where $\boldsymbol{\Psi}=\{\boldsymbol{\Psi}_{ki}\}_{1\le k\le P,1\le i\le N}=\{\psi_{\alpha_k}(\boldsymbol{x}^{(i)})\}_{1\le k\le P,1\le i\le N}$ and $\boldsymbol{K}=\{\boldsymbol{K}_{ij}\}_{1\le i,j\le N}=\{k_e(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})\}_{1\le i,j\le N}$ are respectively the $P\times N$ matrix of polynomials and the $N\times N$ matrix of kernel functions computed at the sample points. In the problem depicted by Equation (8), coefficient vectors $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ and kernel width vector $\boldsymbol{l}$ have to be determined. The number of unknown PC coefficients $\boldsymbol{\beta}$ grows exponentially with both the input dimension n and total expansion degree p, and the number of unknown kernel coefficients $\boldsymbol{\omega}$ is equal to the size of the experimental design. Thus, the evaluation of Equation (8) can be hampered by the overfitting phenomenon. To address this problem, a sparse representation of the surrogate model for capturing the main features of the model response is necessary.

In order to obtain sparse solutions, herein, the SBL is adapted to accommodate the hybrid model. As a Bayesian method, SBL assumes that $g_{PG}(\boldsymbol{x})$ is a random Gaussian process with conditional PDF $f(g_{PC}(\boldsymbol{x}) \mid \boldsymbol{\beta}, \boldsymbol{\omega}, \sigma)$, where σ is the error variance. Meanwhile, it assumes that the error vector $\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{Y}} - \boldsymbol{\Psi\beta} - \boldsymbol{K\omega}$ follows a zero-mean Gaussian distribution with covariance $I_N\sigma$ $\left(\text{ie}, \boldsymbol{\varepsilon} \sim \mathcal{N}_N\left(\mathbf{0}, I_N\sigma\right)\right)$, where I_N is the identity matrix with size N. This leads to a Gaussian likelihood model as follows:

$$f\left(\boldsymbol{\mathcal{Y}}|\boldsymbol{\beta}, \boldsymbol{\omega}, \sigma\right) = (2\pi\sigma)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\Psi\beta} - \boldsymbol{K\omega}\|^2\right). \tag{9}$$

To infer the coefficient vectors $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$, Gaussian prior distributions with zero mean and variance γ_k and λ_j are considered over each $\beta_{\boldsymbol{\alpha}_k}$ and ω_j, respectively. With the prior independence assumption on the coefficients $\beta_{\boldsymbol{\alpha}_k}$ and ω_j, the overall prior distributions are formulated as

$$f\left(\boldsymbol{\beta}|\boldsymbol{\gamma}\right) = \prod_{k=1}^{P} (2\pi\gamma_k)^{-\frac{1}{2}} \exp\left(-\frac{\beta_{\boldsymbol{\alpha}_k}^2}{2\gamma_k}\right) \tag{10}$$

$$f\left(\boldsymbol{\omega}|\lambda\right) = \prod_{j=1}^{N} \left(2\pi\lambda_j\right)^{-\frac{1}{2}} \exp\left(-\frac{\omega_j^2}{2\lambda_j}\right), \tag{11}$$

where $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma})$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda)$. In the above equations, $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_P\}^{\mathrm{T}}$ and $\lambda = \{\lambda_1, \ldots, \lambda_N\}^{\mathrm{T}}$ are the vectors of hyperparameters that control the variance of coefficients $\beta_{\boldsymbol{\alpha}_k}$ and ω_j, respectively.[40] If the set of hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \lambda, \sigma\}$ is known, the posterior PDFs of $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ are given as

$$f\left(\boldsymbol{\beta}|\boldsymbol{\mathcal{Y}}, \boldsymbol{\theta}\right) = (2\pi)^{-\frac{P}{2}}\left|\boldsymbol{\Sigma}_{\boldsymbol{\beta}/\mathcal{Y}}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}\right)^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{\beta}/\mathcal{Y}}^{-1}\left(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}\right)\right\} \tag{12}$$

$$f\left(\boldsymbol{\omega}|\boldsymbol{\mathcal{Y}}, \boldsymbol{\theta}\right) = (2\pi)^{-\frac{N}{2}}\left|\boldsymbol{\Sigma}_{\boldsymbol{\omega}/\mathcal{Y}}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\omega} - \boldsymbol{\mu}_{\boldsymbol{\omega}}\right)^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}/\mathcal{Y}}^{-1}\left(\boldsymbol{\omega} - \boldsymbol{\mu}_{\boldsymbol{\omega}}\right)\right\}, \tag{13}$$

where the posterior covariance and mean of $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ are, respectively,

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}/\mathcal{Y}} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma\Psi}^{\mathrm{T}}\sum\nolimits_{\mathcal{Y}}^{-1}\boldsymbol{\Psi\Gamma} \tag{14}$$

$$\boldsymbol{\mu}_{\boldsymbol{\beta}} = \boldsymbol{\Gamma\Psi}^{\mathrm{T}}\sum\nolimits_{\mathcal{Y}}^{-1}\boldsymbol{\mathcal{Y}} \tag{15}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\omega}/\mathcal{Y}} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda K}^{\mathrm{T}}\sum\nolimits_{\mathcal{Y}}^{-1}\boldsymbol{K\Lambda} \tag{16}$$

$$\boldsymbol{\mu}_{\boldsymbol{\omega}} = \boldsymbol{\Lambda K}^{\mathrm{T}}\sum\nolimits_{\mathcal{Y}}^{-1}\boldsymbol{\mathcal{Y}}, \tag{17}$$

with $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma})$, $\boldsymbol{\Lambda} = \text{diag}(\lambda)$, and $\sum_{\mathcal{Y}} = \sigma I_N + \boldsymbol{\Psi\Gamma\Psi}^{\mathrm{T}} + \boldsymbol{K\Lambda K}^{\mathrm{T}}$. The posterior mean provides a point estimate of the coefficient. Nevertheless, since these parameters are unknown in advance, SBL suggests they should be estimated by maximizing the likelihood function (ie, Bayesian evidence), as follows:

$$\begin{aligned} f\left(\boldsymbol{\mathcal{Y}}|\boldsymbol{\gamma}, \lambda, \sigma\right) &= \int f\left(\boldsymbol{\mathcal{Y}}|\boldsymbol{\beta}, \boldsymbol{\omega}, \sigma\right) f\left(\boldsymbol{\beta}|\boldsymbol{\gamma}\right) f\left(\boldsymbol{\omega}|\lambda\right) d\boldsymbol{\beta} d\boldsymbol{\omega} \\ &= (2\pi)^{-\frac{N}{2}}\left|\sum\nolimits_{\mathcal{Y}}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\mathcal{Y}}^{\mathrm{T}}\sum\nolimits_{\mathcal{Y}}^{-1}\boldsymbol{\mathcal{Y}}\right). \end{aligned} \tag{18}$$

Since there is no closed solution for $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \boldsymbol{\lambda}, \sigma\}$ that maximizes Equation (18), the expectation-maximization (EM) algorithm introduced in Section 3.2 can be employed to obtain the estimate $\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\lambda}}, \hat{\sigma}\}$. Meanwhile, width vector $\boldsymbol{l}$ needs to be optimized. The optimization method that updates $\boldsymbol{l}$ adaptively will be introduced in Section 3.3. This optimization process will be embedded into the EM algorithm in Section 4.

3.2 | EM algorithm for parameter estimation

The EM algorithm is a numerical method to determine $\hat{\boldsymbol{\theta}}$. It proceeds by treating $\{\boldsymbol{\beta}, \boldsymbol{\omega}\}$ as hidden variables and $\mathcal{Y}$ as incomplete data. In this case, complete data $\{\mathcal{Y}, \boldsymbol{\beta}, \boldsymbol{\omega}\}$, which simplify the computation, can be obtained. The algorithm is iterative, with the qth iteration deriving the following expectation and maximization steps:

$$\text{E Step}: P\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y},\boldsymbol{\theta}^{(q)}}\left[\ln f\left(\mathcal{Y}, \boldsymbol{\beta}, \boldsymbol{\omega}|\boldsymbol{\theta}\right)\right] \tag{19}$$

$$\text{M Step}: \boldsymbol{\theta}^{(q+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right), \tag{20}$$

where $P(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ defines a function of the parameter set $\boldsymbol{\theta}$ and its estimate $\boldsymbol{\theta}^{(q)}$ at the qth iteration. $f\left(\mathcal{Y}, \boldsymbol{\beta}, \boldsymbol{\omega}|\boldsymbol{\theta}\right) = f\left(\mathcal{Y}|\boldsymbol{\beta}, \boldsymbol{\omega}, \sigma\right) f\left(\boldsymbol{\beta}|\boldsymbol{\gamma}\right) f\left(\boldsymbol{\omega}|\boldsymbol{\lambda}\right)$ represents the likelihood of the complete data $\{\mathcal{Y}, \boldsymbol{\beta}, \boldsymbol{\omega}\}$. The log-likelihood of the complete data has the following formulation:

$$\begin{aligned}\ln f\left(\mathcal{Y}, \boldsymbol{\beta}, \boldsymbol{\omega}|\boldsymbol{\theta}\right) &= \ln f\left(\boldsymbol{\beta}|\boldsymbol{\gamma}\right) + \ln f\left(\boldsymbol{\omega}|\boldsymbol{\lambda}\right) + \ln f\left(\mathcal{Y}|\boldsymbol{\beta}, \boldsymbol{\omega}, \sigma\right) \\ &= \sum_{k=1}^{P} \ln f\left(\beta_k|\gamma_k\right) + \sum_{j=1}^{N} \ln f\left(\omega_j|\lambda_j\right) + \ln f\left(\mathcal{Y}|\boldsymbol{\beta}, \boldsymbol{\omega}, \sigma\right).\end{aligned} \tag{21}$$

Based on Equation (21), $P(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ can be written as

$$P\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = \sum_{k=1}^{P} P\left(\gamma_k, \boldsymbol{\theta}^{(q)}\right) + \sum_{j=1}^{N} P\left(\lambda_j, \boldsymbol{\theta}^{(q)}\right) + P\left(\sigma, \boldsymbol{\theta}^{(q)}\right). \tag{22}$$

Equation (22) shows that the computation of one parameter in set $\boldsymbol{\theta}$ is dissociated from the others. The expectation terms in Equation (22) with respect to parameters γ_k, λ_j, and σ are given by

$$P\left(\gamma_k, \boldsymbol{\theta}^{(q)}\right) = -\left(\ln\pi + \ln\gamma_k + \gamma_k^{-1} E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\beta_{\boldsymbol{\alpha}_k}^2\right)\right) \tag{23}$$

$$P\left(\lambda_j, \boldsymbol{\theta}^{(q)}\right) = -\left(\ln\pi + \ln\lambda_j + \lambda_j^{-1} E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\omega_j^2\right)\right) \tag{24}$$

$$P\left(\sigma, \boldsymbol{\theta}^{(q)}\right) = -\left(\ln\pi + \ln\sigma + \sigma^{-1} E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\sigma^2\right)\right), \tag{25}$$

where $E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\beta_{\boldsymbol{\alpha}_k}^2\right)$, $E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\omega_j^2\right)$, and $E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\sigma^2\right)$ are the second-order statistical moments of $\beta_{\boldsymbol{\alpha}_k}$, ω_j, and σ, respectively. More specifically, the second-order statistical moment of $\beta_{\boldsymbol{\alpha}_k}$ becomes

$$E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\beta_{\boldsymbol{\alpha}_k}^2\right) = \gamma_k^{(q)} + \left(\gamma_k^{(q)}\right)^2 \boldsymbol{\Psi}_{\cdot k}^{\mathrm{T}} M_{\mathcal{Y}}^{(q)} \boldsymbol{\Psi}_{\cdot k}, \tag{26}$$

where $\boldsymbol{\Psi}_{\cdot k}$ denotes the kth column of data matrix $\boldsymbol{\Psi}$ and $M_{\mathcal{Y}}^{(q)} = \left(\Sigma_{\mathcal{Y}}^{(q)}\right)^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \left(\Sigma_{\mathcal{Y}}^{(q)}\right)^{-1} - \left(\Sigma_{\mathcal{Y}}^{(q)}\right)^{-1}$ is computed by the estimate $\boldsymbol{\theta}^{(q)}$ at the qth iteration. Meanwhile, the second-order statistical moments of ω_j and σ, respectively, have the following similar forms:

$$E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\omega_j^2\right) = \lambda_j^{(q)} + \left(\lambda_j^{(q)}\right)^2 \boldsymbol{K}_{\cdot j}^{\mathrm{T}} M_{\mathcal{Y}}^{(q)} \boldsymbol{K}_{\cdot j} \tag{27}$$

$$E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\sigma^2\right) = \frac{\sigma^{(q)} \operatorname{tr}\left[I_N + \sigma^{(q)} M_{\mathcal{Y}}^{(q)}\right]}{N}, \tag{28}$$

where $\boldsymbol{K}_{\cdot j}$ denotes the jth column of correlation matrix $\boldsymbol{K}$. For the M-step, we maximize Equation (23) with respect to the unknown γ_k, ie,

$$\begin{aligned}\gamma_k &= \underset{\gamma_k \geq 0}{\operatorname{argmax}} P\left(\gamma_k, \boldsymbol{\theta}^{(q)}\right) \\ &= \underset{\gamma_k \geq 0}{\operatorname{argmax}} - \left(\ln\pi + \ln\gamma_k + \gamma_k^{-1} E_{\boldsymbol{\beta},\boldsymbol{\omega}|\mathcal{Y};\boldsymbol{\theta}^{(q)}}\left(\beta^2_{\boldsymbol{\alpha}_k}\right)\right).\end{aligned} \tag{29}$$

Equation (29) can be solved by setting the derivative of Equation (23) to zero, then the following update formula for hyperparameter γ_k is obtained:

$$\gamma_k^{(q+1)} = \gamma_k^{(q)} + \left(\gamma_k^{(q)}\right)^2 \boldsymbol{\Psi}_{\cdot k}^T M_{\mathcal{Y}}^{(q)} \boldsymbol{\Psi}_{\cdot k}. \tag{30}$$

Likewise, the update rules for ω_j and σ can also be simply derived, respectively, as

$$\lambda_j^{(q+1)} = \lambda_j^{(q)} + \left(\lambda_j^{(q)}\right)^2 \boldsymbol{K}_{\cdot j}^{\mathrm{T}} M_{\mathcal{Y}}^{(q)} \boldsymbol{K}_{\cdot j} \tag{31}$$

$$\sigma^{(q+1)} = \frac{\sigma^{(q)} \operatorname{tr}\left[I_N + \sigma^{(q)} M_{\mathcal{Y}}^{(q)}\right]}{N}. \tag{32}$$

3.3 | Optimization of kernel width vector

To find the optimum width vector $\boldsymbol{l}$, the negative log-likelihood $-\ln f\,(\mathcal{Y}|\boldsymbol{\gamma}, \lambda, \sigma)$ is minimized with respect to $\boldsymbol{l}$ using the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno algorithm,[39] which requires the derivatives with respect to $\boldsymbol{l}$. This procedure is equivalent to minimizing the following cost function:

$$\mathcal{L} = \frac{1}{2}\left(\ln\left|\textstyle\sum_{\mathcal{Y}}\right| + \mathcal{Y}^{\mathrm{T}} \textstyle\sum_{\mathcal{Y}}^{-1} \mathcal{Y}\right). \tag{33}$$

To avoid adding positive constraints in the optimization problem, $\ln\boldsymbol{l} = \{\ln l_d\}_{1 \leq d \leq n}$ is used. Hence, the optimization problem is written as

$$\underset{\ln\boldsymbol{l}}{\arg\min}\left[\mathcal{L} = \frac{1}{2}\left(\ln\left|\textstyle\sum_{\mathcal{Y}}\right| + \mathcal{Y}^{\mathrm{T}} \textstyle\sum_{\mathcal{Y}}^{-1} \mathcal{Y}\right)\right]. \tag{34}$$

Based on the chain rule, the gradient of the cost function $\mathcal{L}$ with respect to $\ln l_d$ is given by

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\ln l_d} &= \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial\mathcal{L}}{\partial\boldsymbol{K}_{ij}} \frac{\partial\boldsymbol{K}_{ij}}{\partial\ln l_d} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{D}_{ij} \frac{\partial\boldsymbol{K}_{ij}}{\partial\ln l_d},\end{aligned} \tag{35}$$

where $\boldsymbol{K}_{ij}$ denotes the entry in the ith row and jth column of the kernel matrix $\boldsymbol{K}$. Due to the first term $\boldsymbol{D}_{ij}$ on the right-hand side is independent of the kernel widths, we need to calculate $\boldsymbol{D}_{ij}$ and $\partial\boldsymbol{K}_{ij}/\partial\ln l_k$, respectively. Applying the Lemma in Appendix A, matrix $\boldsymbol{D}$ is derived as

$$\begin{aligned}\boldsymbol{D} &= \frac{1}{2}\frac{\partial}{\partial\boldsymbol{K}}\left(\ln\left|\textstyle\sum_{\mathcal{Y}}\right| + \mathcal{Y}^{\mathrm{T}} \textstyle\sum_{\mathcal{Y}}^{-1} \mathcal{Y}\right) \\ &= \frac{1}{2}\left(\textstyle\sum_{\mathcal{Y}}^{-1} - \sum_{\mathcal{Y}}^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \sum_{\mathcal{Y}}^{-1}\right) \frac{\partial\sum_{\mathcal{Y}}}{\partial\boldsymbol{K}}.\end{aligned} \tag{36}$$

By substituting $\sum_{\mathcal{Y}} = \sigma I_N + \boldsymbol{\Psi}\boldsymbol{\Gamma}\boldsymbol{\Psi}^{\mathrm{T}} + \boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{K}^{\mathrm{T}}$ to Equation (36), it is rewritten as

$$\begin{aligned}\boldsymbol{D} &= \frac{1}{2}\left(\textstyle\sum_{\mathcal{Y}}^{-1} - \sum_{\mathcal{Y}}^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \sum_{\mathcal{Y}}^{-1}\right) \frac{\partial}{\partial\boldsymbol{K}}\left(\sigma I_N + \boldsymbol{\Psi}\boldsymbol{\Gamma}\boldsymbol{\Psi}^{\mathrm{T}} + \boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{K}^{\mathrm{T}}\right) \\ &= \frac{1}{2}\left(\textstyle\sum_{\mathcal{Y}}^{-1} - \sum_{\mathcal{Y}}^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \sum_{\mathcal{Y}}^{-1}\right) 2\boldsymbol{K}\boldsymbol{\Lambda} \\ &= \left(\textstyle\sum_{\mathcal{Y}}^{-1} - \sum_{\mathcal{Y}}^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \sum_{\mathcal{Y}}^{-1}\right) \boldsymbol{K}\boldsymbol{\Lambda}.\end{aligned} \tag{37}$$

For the second term on the right-hand side in Equation (35), the derivative of $\boldsymbol{K}_{ij}$ with respect to $\ln l_d$ is given by

$$\frac{\partial \boldsymbol{K}_{ij}}{\partial \ln l_d} = 2\boldsymbol{K}_{ij}\left(x_d^{(i)} - x_d^{(j)}\right)^2. \tag{38}$$

By combining Equations (37) and (38), we can compute the derivative Equation (35) of the cost function $\mathcal{L}$, ie,

$$\frac{\partial \mathcal{L}}{\partial \ln l_d} = \sum_{i=1}^{N}\sum_{j=1}^{N} 2\boldsymbol{D}_{ij}\boldsymbol{K}_{ij}\left(x_d^{(i)} - x_d^{(j)}\right)^2 = 0. \tag{39}$$

4 | THE OPTIMIZATION ALGORITHM

In this section, an optimization algorithm is introduced for the proposed surrogate modeling method. Suppose that we have generated an experimental design $\mathcal{X} = \left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right)$ and calculated the corresponding vector of response $\mathcal{Y} = \left(y^{(1)}, \ldots, y^{(N)}\right)$, we obtain the algorithm provided as Algorithm 1.

Algorithm 1 The optimization algorithm for PC-GK-SBL

Input: Training sample set $\{\mathcal{X}, \mathcal{Y}\}$.

Initialization: Set $q = 0$.

Initialize $\boldsymbol{\Gamma}^{(0)} = \mathrm{diag}(\boldsymbol{\gamma}^{(0)}) = I_P,\ \boldsymbol{\Lambda}^{(0)} = \mathrm{diag}(\lambda^{(0)}) = I_N,\ \sigma^{(0)} = \mathrm{var}(\mathcal{Y})/N$, and $\boldsymbol{l}^{(0)}$

Iteration: Increment q by 1 and perform these steps

- EM step:

Compute parameters $\gamma_k^{(q+1)} = \gamma_k^{(q)} + \left(\gamma_k^{(q)}\right)\boldsymbol{\Psi}_{\bullet k}^{\mathrm{T}} M_{\mathcal{Y}}^{(q)} \boldsymbol{\Psi}_{\bullet k}$, $\lambda_j^{(q+1)} = \lambda_j^{(q)} + \left(\lambda_j^{(q)}\right)\boldsymbol{K}_{\bullet j}^{\mathrm{T}} M_{\mathcal{Y}}^{(q)} \boldsymbol{K}_{\bullet j}$

and $\sigma^{(q+1)} = \frac{\sigma^{(q)}\mathrm{tr}\left[I_N + \sigma^{(q)} M_{\mathcal{Y}}^{(q)}\right]}{N}$

- Update step:

Update $\boldsymbol{\Gamma}^{(q+1)} = \mathrm{diag}(\boldsymbol{\gamma}^{(q+1)}),\ \boldsymbol{\Lambda}^{(q+1)} = \mathrm{diag}(\lambda^{(q+1)}),\ \sum_{\mathcal{Y}}^{(q+1)} = \sigma I_N + \boldsymbol{\Psi}\boldsymbol{\Gamma}^{(q+1)}\boldsymbol{\Psi}^{\mathrm{T}} + \boldsymbol{K}\boldsymbol{\Lambda}^{(q+1)}\boldsymbol{K}^{\mathrm{T}}$

and $M_{\mathcal{Y}}^{(q+1)} = \left(\sum_{\mathcal{Y}}^{(q+1)}\right)^{-1} \mathcal{Y}\mathcal{Y}^{\mathrm{T}} \left(\sum_{\mathcal{Y}}^{(q+1)}\right)^{-1} - \left(\sum_{\mathcal{Y}}^{(q+1)}\right)^{-1}.$

- Width optimization step:

Use quasi-Newton BFGS algorithm to find a better $\boldsymbol{l}^{(q+1)}$

Update $\boldsymbol{K}, \sum_{\mathcal{Y}}^{(q+1)}$ and $M_{\mathcal{Y}}^{(q+1)}$

- Stop rule: Iterate if $\left|\mathcal{L}^{(q+1)} - \mathcal{L}^{(q)}\right| \geq \mathbf{Tol}$

Output: $\boldsymbol{\mu}_{\beta} = \boldsymbol{\Gamma}^{(q+1)}\boldsymbol{\Psi}^{\mathrm{T}}\left(\sum_{\mathcal{Y}}^{(q+1)}\right)^{-1}\mathcal{Y}$, $\boldsymbol{\mu}_{\omega} = \boldsymbol{\Lambda}^{(q+1)}\boldsymbol{K}^{\mathrm{T}}\left(\sum_{\mathcal{Y}}^{(q+1)}\right)^{-1}\mathcal{Y}, \hat{\sigma} = \sigma^{q+1}$ and $\boldsymbol{l} = \boldsymbol{l}^{(q+1)}$.

In the process of carrying out the EM step to update $\boldsymbol{\gamma}$ and λ, some γ_k's and λ_j's are driven to zero (or are numerically indistinguishable from infinitesimal given the machine precision[29]), effectively forcing the associated coefficients to zero. In other words, if $\gamma_k = 0$ or $\lambda_j = 0$, then $f\left(\beta_{\boldsymbol{\alpha}_k} | \gamma_k = 0\right) = \delta\left(\beta_{\boldsymbol{\alpha}_k}\right)$ or $f(\omega_j | \lambda_j = 0) = \delta(\omega_j)$, which will force the posterior probability to satisfy[29]

$$\begin{aligned} \mathrm{Prob}\left(\beta_{\boldsymbol{\alpha}_k} = 0 | \mathcal{Y}, \gamma_k = 0\right) = 1 \\ \mathrm{Prob}\left(\omega_j = 0 | \mathcal{Y}, \lambda_j = 0\right) = 1. \end{aligned} \tag{40}$$

Equation (40) guarantees the sparsity of $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$. Only a relatively small set of $\beta_{\boldsymbol{\alpha}_k}$ and ω_j, for which the corresponding γ_i and λ_j remain relatively large compared to zero, is remained to contribute for the representation of $g_{\mathrm{PG}}(\boldsymbol{X})$, and the level of sparsity is determined automatically.[40] The proposed algorithm has converged when the change in lost function $\mathcal{L}$ in two consecutive iterations is less than the tolerance level $\mathbf{Tol} = 10^{-6}$. At convergence of the algorithm, we can use Equations (12) and (13) to compute the predictive distribution for an unknown point $\boldsymbol{x}$ conditioned on $\hat{\boldsymbol{\theta}} = \left\{\hat{\boldsymbol{\gamma}}, \hat{\lambda}, \hat{\sigma}\right\}$, ie,

$$f\left(g_{\mathrm{PG}}(\boldsymbol{x}) | \mathcal{Y}, \hat{\boldsymbol{\theta}}\right) = \int f\left(g_{\mathrm{PG}}(\boldsymbol{x}) | \boldsymbol{\beta}, \boldsymbol{\omega}, \hat{\sigma}\right) f\left(\boldsymbol{\beta} | \mathcal{Y}, \hat{\boldsymbol{\theta}}\right) f\left(\boldsymbol{\omega} | \mathcal{Y}, \hat{\boldsymbol{\theta}}\right) d\boldsymbol{\beta} d\boldsymbol{\omega}. \tag{41}$$

Since the three terms on the right-hand side are all Gaussian, it is rewritten as

$$f\left(g_{\mathrm{PG}}(\boldsymbol{x})\,|\,\mathcal{Y},\hat{\boldsymbol{\theta}}\right)=\frac{1}{\sqrt{2\pi v(\boldsymbol{x})}}\exp\left(-\frac{\left(g_{\mathrm{PG}}(\boldsymbol{x})-\hat{g}(\boldsymbol{x})\right)^2}{2v(\boldsymbol{x})}\right), \tag{42}$$

with predictive mean $\hat{g}(\boldsymbol{x}) = \boldsymbol{\mu_\beta}^{\mathrm{T}}\boldsymbol{\psi}(\boldsymbol{x}) + \boldsymbol{\mu_\omega}^{\mathrm{T}}\boldsymbol{k}_e(\boldsymbol{x})$ and associated variance $v(\boldsymbol{x}) = \hat{\sigma} + \boldsymbol{\psi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\Sigma}_{\boldsymbol{\beta}/\mathcal{Y}}\boldsymbol{\psi}(\boldsymbol{x}) + \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\Sigma}_{\boldsymbol{\omega}/\mathcal{Y}}\boldsymbol{k}_e(\boldsymbol{x})$. The prediction mean $\hat{g}(\boldsymbol{x})$ is then used as the surrogate to the original model. The predictive variance $v(\boldsymbol{x})$ comprises the estimated error variance $\hat{\sigma}$ on data and that due to the uncertainty in the prediction of these coefficients (ie, the latter two terms). It is notable that the predictive variance plays an important role in the context of design of experiment. It can be defined as an active learning function for sequentially selecting new points, in order to reduce the generalization error of the surrogate model. This sequential design (also known as the adaptive sampling method) has been studied in the Kriging framework.[41,42]

In addition, the optimal degree p of PCE also needs to be determined. This ability can be assessed in practice by LOOCV over the training set, as introduced in Section 2.1. In this paper, the LOOCV-based degree adaptivity method is used for exploiting the full potential of the proposed method. The detailed discussion of the degree adaptivity method is omitted, and we refer the reader to the work of Blatman and Sudret[13] for more details.

5 | NUMERICAL EXAMPLES

This section is dedicated to the validation and assessment of the proposed surrogate modeling method. Three benchmark functions are first considered. In these cases, the same settings are used in these analyses for the sake of consistency, unless explicitly stated otherwise. The quality of the various surrogate models is measured by building an independent testing set. The testing set is made of a large number of randomly selected samples that are exactly evaluated. The exact model responses y_{nt} are compared with the predictions $\hat{y}_{nt}$, and the relative mean-square error (RMSE) is computed to gauge the overall accuracy of the surrogate, ie,

$$\mathrm{RMSE}=\frac{\sum_{nt=1}^{N_{\mathrm{test}}}\left(y_{nt}-\hat{y}_{nt}\right)^2}{\sum_{nt=1}^{N_{\mathrm{test}}}\left(y_{nt}-\overline{y}\right)^2}\quad N_{\mathrm{test}}=10000, \tag{43}$$

where $\overline{y}$ is the mean value of $(y_1,\ldots,y_{10\,000})$.

Due to PC-Kriging performing well compared to the original sparse PCE and original Kriging,[26,27] the performance of PC-GK-SBL is then compared with that of two formulations of PC-Kriging, namely, sequential PC-Kriging (SPC-Kriging) and optimal PC-Kriging (OPC-Kriging). The results of SPC-Kriging and OPC-Kriging are obtained using UQLab, which is a MATLAB-based uncertainty quantification software.[15,43] SPC-Kriging employs first the LAR algorithm to determine the optimal sparse set of multivariate orthonormal polynomials in the input space. Then, the set of polynomials is used as the trend of a universal Kriging model. OPC-Kriging employs the same LAR algorithm as SPC-Kriging, yet iteratively adds polynomials to the trend part of the universal Kriging model one by one, and fits the hyperparameters of the autocorrelation function in each iteration. In this case, polynomials are added to the trend in the order they are selected by the LAR algorithm. Based on the leave-one-out error, the best surrogate is found to be OPC-Kriging. The surrogate models are built using a purely space-filling design strategy called Latin hypercube sampling (LHS).[44] To enrich the designs automatically, the nested LHS is used.[45] The main idea of the nested LHS is to enrich an existing experimental design generated by LHS such that the resulting combined design meets the LHS requirements as closely as possible. For more details, see the works of Wang[45] and Blatman and Sudret.[46]

5.1 | Example 1: Ishigami function

We first consider the Ishigami function, which is a highly nonlinear function with three inputs proposed in UQLab,[15] ie,

$$g(\boldsymbol{X})=\sin X_1 + a\sin^2 X_2 + bX_3^4 a\sin X_1,$$

where $X_i(i=1,2,3)$ are uniformly distributed on the interval $[-\pi,\pi]$, $a=7$, and $b=0.1$.

Since random variables are uniformly distributed in the input space, multivariate Legendre polynomials are chosen in the test case. The maximum PC expansion order is fixed at 10, which means that the initial pool of candidate basis functions $\boldsymbol{\psi}(\boldsymbol{X})$ contains $\binom{10+3}{3} = 286$ multivariate Legendre polynomials. The analysis is replicated 50 times to assess the statistical uncertainty. For each replication, an initial experimental design generated by LHS with size $N = 40$ is used, then the design is augmented by the nested LHS up to 160 sample points.

The RMSE on a logarithmic scale computed using 10 000 random samples is reported in Figure 1 under the form of boxplots. Each box is characterized by the first quartile (bottom line), the median (red line), and the third quartile (upper line). The whiskers indicate the variability of the data outside the first and third quartiles. The ends of the whiskers lie at a distance of 1.5 interquartile range from the first/third quartile. Outliers are represented by red crosses. It turns out that both the median and interquartile range are significantly smaller compared to that of OPC-Kriging and SPC-Kriging. Moreover, the mean and standard deviation of the RMSE corresponding to the 50 experimental designs with different sizes are given in Table 3. It is worth mentioning that the proposed method performs best with the least mean and standard deviation. In particular, a highly accurate surrogate model is obtained with only 40 sample points. Table 3 also shows that PC-GK-SBL has less variability, which confirms its superior robustness against the two PC-Kriging techniques.

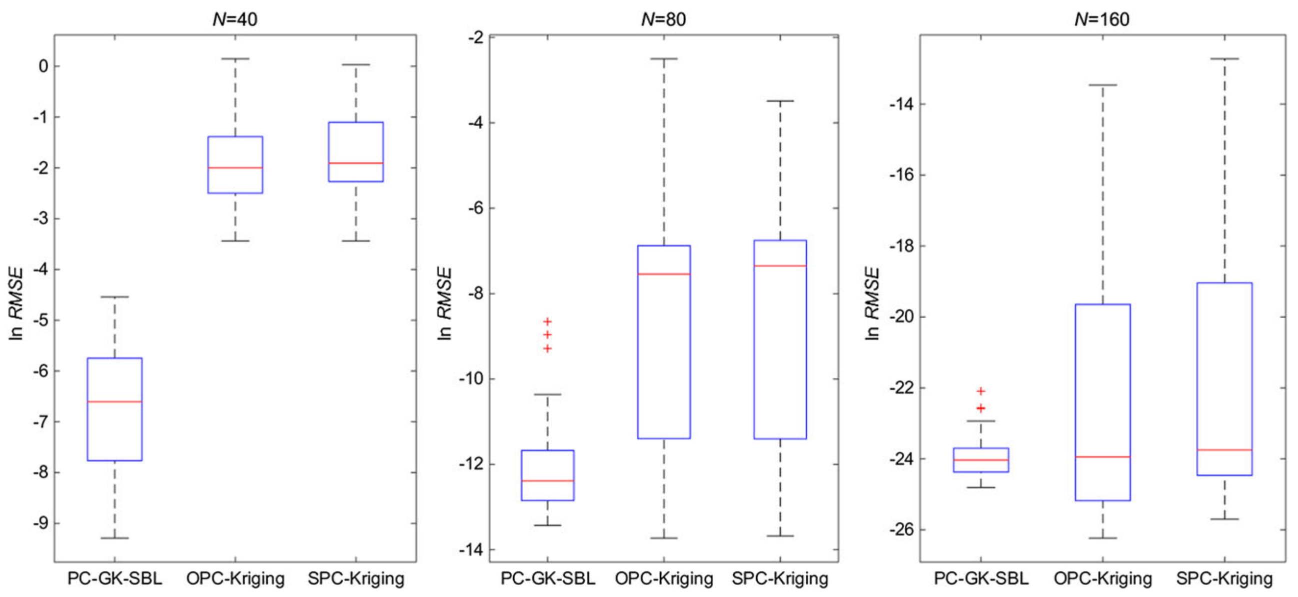

FIGURE 1 Example 1: Comparison of PC-GK-SBL, OPC-Kriging, and SPC-Kriging for 50 replications of 40 sample points augmented until 160 sample points by the nested Latin hypercube sampling technique. RMSE, relative mean-square error

TABLE 3 Mean and standard deviation of the relative mean-square error (RMSE) on a logarithmic scale for different methods

	N	RMSE	
		Mean	**Standard deviation**
PC-GK-SBL	40	−6.6475	1.2467
	80	−12.0639	1.1373
	160	−23.9663	0.6129
OPC-Kriging	40	−1.7485	0.9857
	80	−8.8445	2.6315
	160	−22.8105	2.9837
SPC-Kriging	40	−1.6844	0.9336
	80	−8.6362	2.5988
	160	−22.3060	3.0382

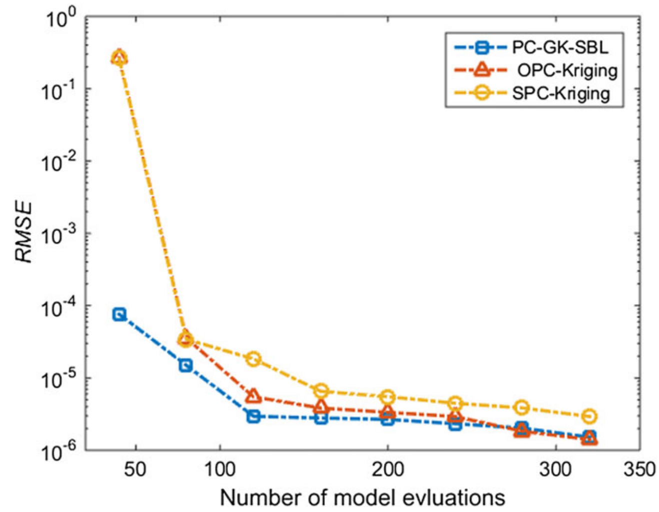

FIGURE 2 Example 1: Convergence curves of PC-GK-SBL, OPC-Kriging, and SPC-Kriging using a quasi-random Sobol sequence, $p = 10$. RMSE, relative mean-square error

To further assess the effect of the size of experimental design, we examine the convergence of the RMSE computed by PC-GK-SBL, OPC-Kriging, and SPC-Kriging, whereas N varies from 40 to 320. The considered experimental designs are generated by a quasi-random Sobol experimental sequence. The samples are directly imported from UQLab in our framework so that exactly the same information is provided to these methods. It is shown that PC-GK-SBL yields faster convergence, as shown in Figure 2. In particular, it requires 40 sample points to achieve an accuracy level of 10^{-4}, whereas OPC-Kriging and SPC-Kriging need 80 sample points to achieve a similar level of accuracy. For a sufficiently large size of experimental design, PC-GK-SBL and OPC-Kriging converge to the same accuracy, where, in both cases, the major features of the original model will be properly captured.

5.2 | Example 2: Borehole function

The Borehole function is a benchmark function used for emulation and prediction tests.[47] The function has eight input parameters and initially models the water flow through a borehole by the following function:

$$g(\boldsymbol{x}) = \frac{2\pi x_3 (x_4 - x_6)}{\ln(x_2/x_1)\left(1 + 2x_7x_3/\ln(x_2/x_1)\, x_1^2 x_8 + x_3/x_5\right)}.$$

The feasible ranges of variation of the input variables are given in Table 4. Since input vector $\boldsymbol{X}$ obeys a jointly uniform distribution, multivariate Legendre polynomials are used in the example.

We consider $p = 3$ for this problem for investigating the performance of the proposed surrogate modeling method. The pair $(d,p) = (8,3)$ leads to 165 multivariate Legendre polynomials. The analysis is replicated 50 times in order to assess

TABLE 4 Inputs of the Borehole function and their ranges

Parameters	Range
X_1	[0.05, 0.15]
X_2	[100, 50000]
X_3	[63070, 115600]
X_4	[990, 1110]
X_5	[63.1, 116]
X_6	[700, 820]
X_7	[1120, 1680]
X_8	[9855, 12045]

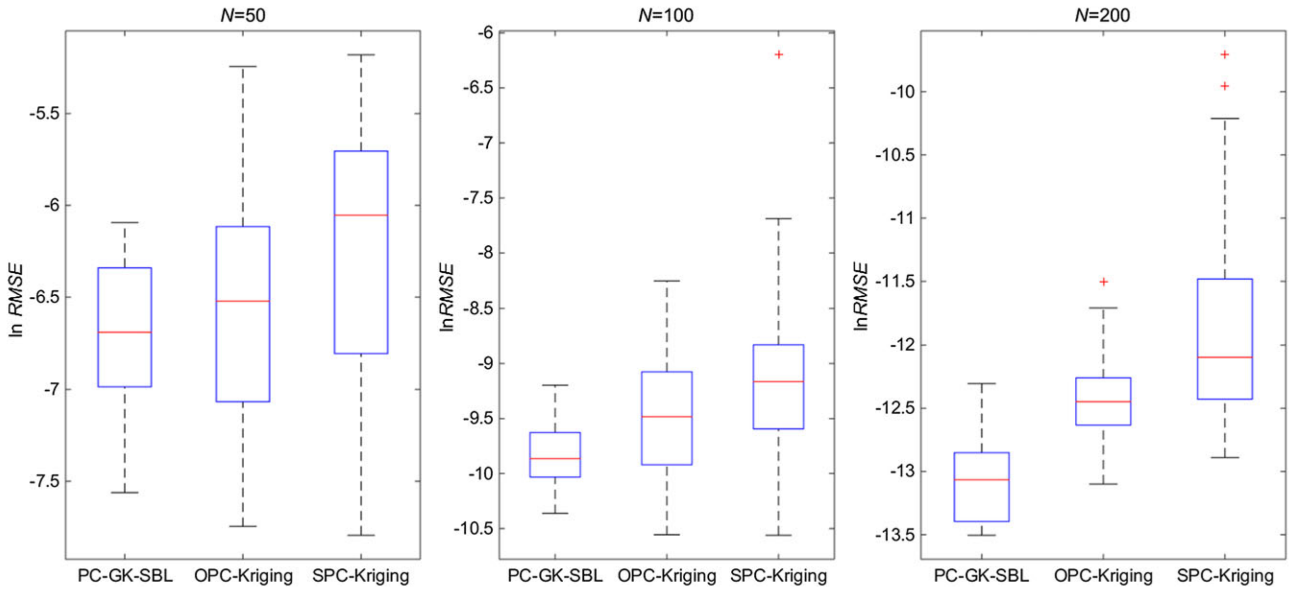

FIGURE 3 Example 2: Comparison of PC-GK-SBL, OPC-Kriging, and SPC-Kriging for 50 replications of 50 points augmented until 200 points by the nested Latin hypercube sampling technique. RMSE, relative mean-square error [Colour figure can be viewed at wileyonlinelibrary.com]

the statistical uncertainty. The RMSE on a logarithmic scale for each of the 50 experiment designs is computed according to Equation (43). For each replication, the initial experimental design consists of 50 design points that are augmented by the nested LHS technique until 200 design points.

The results are provided in Figure 3 under the form of boxplots with $N = 50$, $N = 100$, and $N = 200$. As in the previous example, the mean and standard deviation of the RMSE corresponding to the 50 experimental designs with different sizes are given in Table 5. PC-GK-SBL yields a more accurate surrogate model than OPC-Kriging and SPC-Kriging with respect to the smaller medians of the boxes and means respectively shown in Figure 3 and Table 5. Furthermore, it is also observed that PC-GK-SBL results in slightly more robust approximations with the significantly smaller interquartile range of boxes and less variability.

It is found that SPC-Kriging has the worst performance especially for a large size, as it uses a straightforward approach that directly embeds the set of polynomials found by the LAR algorithm into a universal Kriging. In other words, SPC-Kriging does not use a model selection strategy to decrease the complexity of the surrogate model. As a result, it brings some unrelevant information to the final Kriging model. On the other hand, the performance of OPC-Kriging is slightly better than the performance of SPC-Kriging as it reduces the number of polynomials in the trend part and, thus, reduces the complexity of the surrogate model.

For the sake of completeness, convergence curves are provided in Figure 4, whereas N varies between 50 and 400. Over the entire displayed range of experimental designs in Figure 4, the performance of PC-GK-SBL is better than that of OPC-Kriging and SPC-Kriging, as faster convergence is achieved.

TABLE 5 Mean and standard deviation of the relative mean-square error (RMSE) on a logarithmic scale for different methods

	N	RMSE	
		Mean	**Standard deviation**
PC-GK-SBL	50	−6.7120	0.3996
	100	−9.8202	0.3006
	200	−13.0327	0.3601
OPC-Kriging	50	−6.5358	0.6261
	100	−9.4656	0.5557
	200	−12.4266	0.3454
SPC-Kriging	50	−6.2428	0.7340
	100	−9.1372	0.7393
	200	−11.8235	0.8276

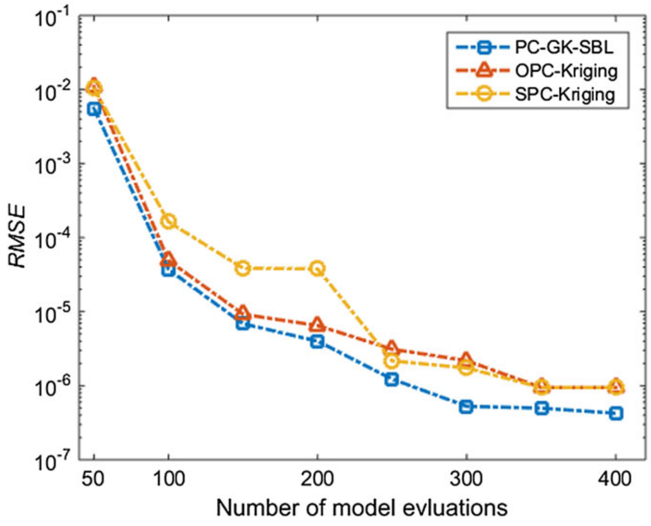

FIGURE 4 Example 2: Convergence curves of PC-GK-SBL, OPC-Kriging, and SPC-Kriging using a quasi-random Sobol sequence, $p = 3$. RMSE, relative mean-square error

5.3 | Example 3: Wing Weight function

Consider a Wing Weight model available in the Virtual Library of Simulation Experiments (http://www.sfu.ca/~ssurjano). It models a light aircraft wing, and the model output is the wing's weight.[48,49]

$$g(\boldsymbol{X}) = 0.036 \mathrm{S}_{\mathrm{w}}^{0.758} \mathrm{W}_{\mathrm{fw}}^{0.0035} \left(\frac{\mathrm{A}_{\mathrm{r}}}{\cos^2(Q_s)} \right)^{0.6} \mathrm{D}_{\mathrm{pc}}^{0.006} \mathrm{T}_{\mathrm{r}}^{0.04} \left(\frac{100\mathrm{T}_{\mathrm{c}}}{\cos(\mathrm{Q}_s)} \right)^{-0.3} \left(\mathrm{N}_{\mathrm{z}} \mathrm{W}_{\mathrm{dg}} \right)^{0.49} + \mathrm{S}_{\mathrm{w}} \mathrm{W}_{\mathrm{p}}$$

The feasible ranges of variation of the inputs are given in Table 6. Different values of the parameters can affect the model performance. The same study as in the previous examples is conducted for the function. The total degree is fixed at 5, which corresponds to $\binom{10+5}{5} = 3003$ multivariate Legendre polynomials.

As in the previous test case, the effectiveness and robustness of the method are assessed by replicating the analyses using 50 random experimental designs. The results are provided in Figure 5 under the form of boxplots and in Table 7 under the form of mean and standard deviation with $N = 75$, $N = 100$, and $N = 150$. The accuracy of PC-GK-SBL and OPC-Kriging is comparable, as shown by the similar median and mean of the results for both methods. With respect to the robustness, the proposed method exhibits a significantly smaller spread than OPC-Kriging and SPC-Kriging. As the size of experimental design increases, the advantage becomes much more obvious. The robustness of the method is

TABLE 6 Parameters of the Wing Weight model and their meaning and ranges

Parameters	Meaning	UOM	Range
$\mathrm{S_w}$	Wing area	ft^2	[150, 200]
$\mathrm{W_{fw}}$	Weight of fuel in the wing	lb	[220, 300]
$\mathrm{A_r}$	Aspect ratio	–	[6, 10]
$\mathrm{Q_s}$	Quarter-chord sweep	degrees	[−10, 10]
$\mathrm{D_{pc}}$	Dynamic pressure at cruise	$\mathrm{lb/ft}^2$	[16, 45]
$\mathrm{T_r}$	Taper ratio	–	[0.5, 1]
$\mathrm{T_c}$	Aerofoil thickness-to-chord ratio	–	[0.08, 0.18]
$\mathrm{N_z}$	Ultimate load factor	–	[2.5, 6]
$\mathrm{W_{dg}}$	Flight design gross weight	lb	[1700, 2500]
$\mathrm{W_p}$	Paint weight	$\mathrm{lb/ft}^2$	[0.025, 0.08]

Abbreviation: UOM, unit of measure.

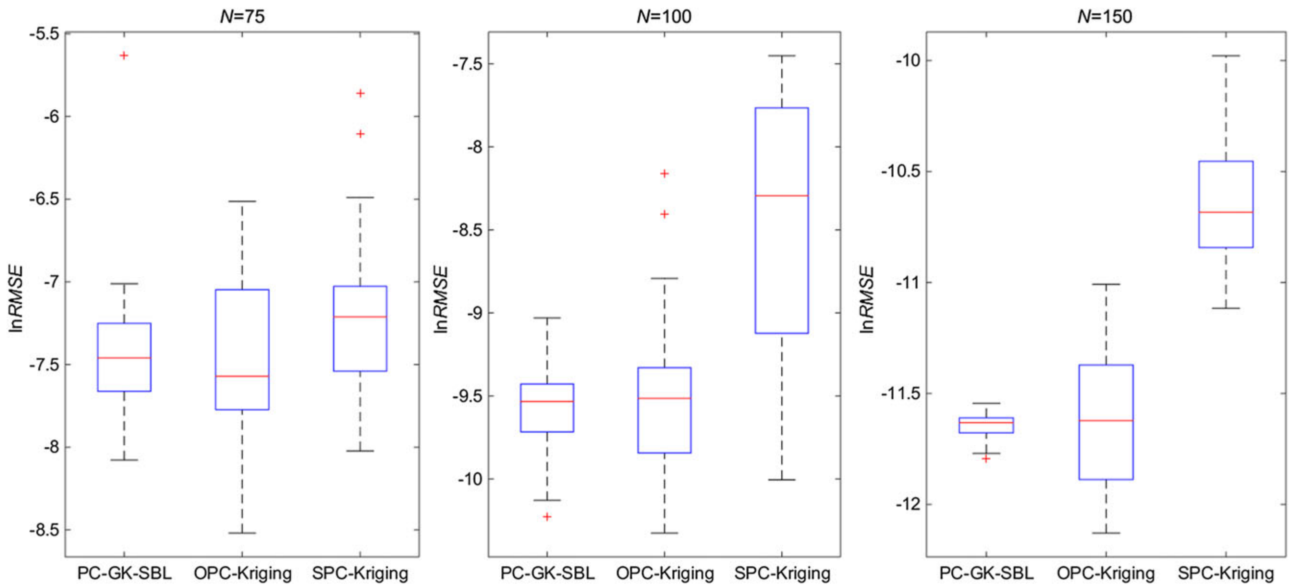

FIGURE 5 Example 3: Comparison of PC-GK-SBL, OPC-Kriging, and SPC-Kriging for 50 replications of 75 points augmented until 150 points by the nested Latin hypercube sampling technique. RMSE, relative mean-square error [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Mean and standard deviation of the relative mean-square error (RMSE) on a logarithmic scale for different methods

	N	**RMSE**	
		Mean	**Standard deviation**
	75	−7.2166	0.3801
PC-GK-SBL	100	−9.5971	0.2707
	150	−11.6429	0.0642
	75	−7.4477	0.5220
OPC-Kriging	100	−9.5199	0.4601
	150	−11.5977	0.3069
	75	−7.2174	0.4601
SPC-Kriging	100	−8.4705	0.7638
	150	−10.6238	0.3114

actually better illustrated by the consistency in the generated surrogate models when changing the experimental design. Furthermore, it can be noticed that several log values are considered as outliers by the boxplot of OPC-Kriging. This can be attributed to an overoptimistic estimation of the generalization capacity by the cross-validation model selection. The performance difference is apparently more significant between PC-GK-SBL and SPC-Kriging than between PC-GK-SBL and OPC-Kriging. The phenomenon is in line with the findings in the previous example.

We finally examine the convergence of the RMSE computed by PC-GK-SBL, OPC-Kriging, and SPC-Kriging, considering experimental designs of varying sizes obtained with the quasi-random Sobol experimental sequence. The results are provided in Figure 6. As expected, SPC-Kriging has the slowest convergence. In contrast, PC-GK-SBL is revealed to be particularly efficient, outperforming both OPC-Kriging and SPC-Kriging. In particular, it yields a relatively small RMSE close to 10^{-6} using $N = 250$ model evaluations.

5.4 | Example 4: simplified wing box structure

In the example, PC-GK-SBL is now applied to a simplified wing box structure.[50,51] Figure 7 shows the schematic diagram of the simplified wing box model. This structure consists of 64 bars and 42 plates. The 64 bars are divided into three groups, ie, 24 bars in the $\boldsymbol{x}$-direction, 16 bars in the $\boldsymbol{y}$-direction, and 24 bars in the $\boldsymbol{z}$-direction. The length of the bars in the $\boldsymbol{x}$-, $\boldsymbol{y}$-, and $\boldsymbol{z}$-directions is denoted as L_x, L_y, and L_z, respectively. The sectional area of all the bars is represented as $\boldsymbol{A}$, the

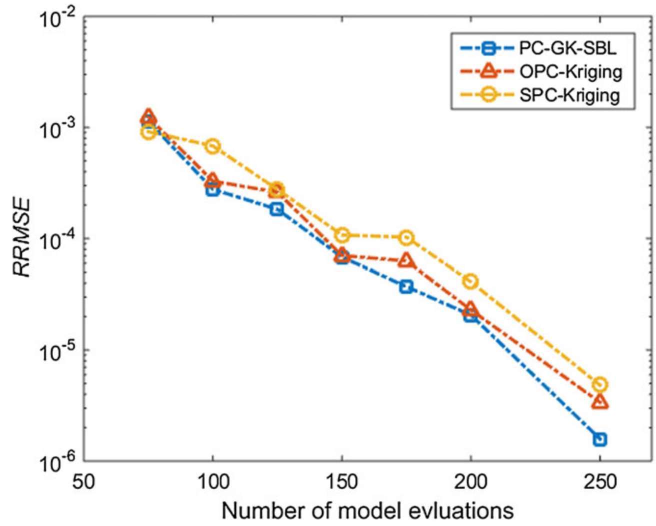

FIGURE 6 Example 3: Convergence curves of PC-GK-SBL, OPC-Kriging, and SPC-Kriging using a quasi-random Sobol sequence, $p = 5$. RMSE, relative mean-square error

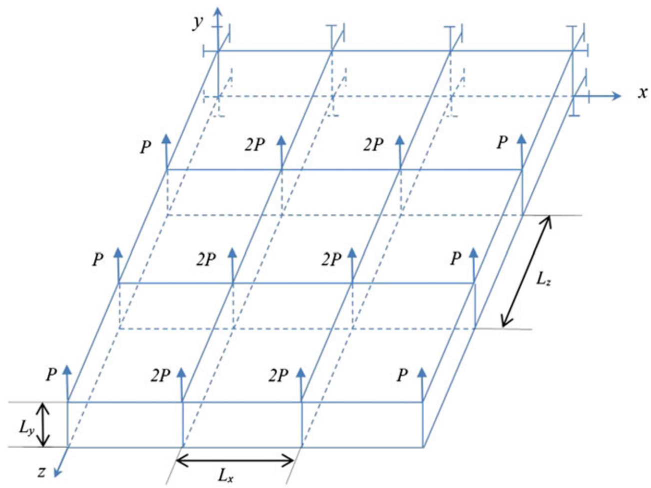

FIGURE 7 The simplified wing box structure model [Colour figure can be viewed at wileyonlinelibrary.com]

thickness of all the plates is denoted as TH, E is the elastic modulus of all bars and plates, and P is the external load. The Poisson's ratio is 0.3. It is assumed that L_x, L_y, L_z, $\boldsymbol{A}$, E, P, and TH are random parameters following a Gaussian distribution. The distribution parameters of these inputs are shown in Table 5. The maximum node displacement of the structure in direction $\boldsymbol{y}$, denoted by $D_{\max}$, represents the response quantity of interest and is computed with a finite-element analysis code developed in ANSYS 18 software in this work. Figure 8 shows the deformation distribution of the wing box, in which

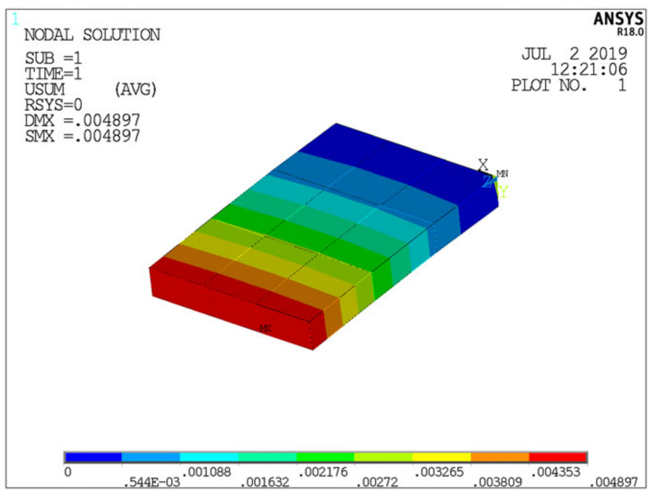

FIGURE 8 Deformation distribution of the wing box structure

TABLE 8 Parameters of the wing box structural model

Parameters	Meaning	UOM	Mean	Coefficient of variation
L_x	Length of the bar in the $\boldsymbol{x}$-direction	m	0.4	0.1
L_y	Length of the bar in the $\boldsymbol{y}$-direction	m	0.2	0.1
L_z	Length of the bar in the $\boldsymbol{z}$-direction	m	0.6	0.1
$\boldsymbol{A}$	Sectional area	m^2	1×10^{-4}	0.1
E	Elastic modulus	Gpa	7.1	0.1
P	External load	N	1.5×10^3	0.1
TH	Thickness	m	3×10^{-3}	0.1

Abbreviation: UOM, unit of measure.

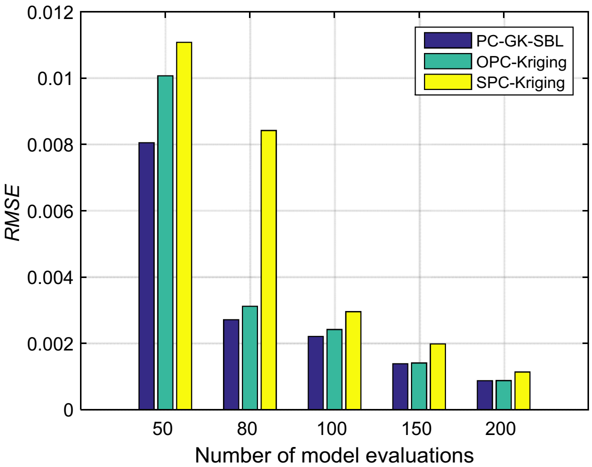

FIGURE 9 Comparisons of PC-GK-SBL, OPC-Kriging, and SPC-Kriging using the relative mean-square error (RMSE) with different sample sizes

the realization of input parameters is selected as the mean in Table 8. In this case, the maximum node displacement appears on the top side of the free end, where the value is 0.004897 m. It can be concluded that the more the distance between the node and the fixed end is, the larger the node displacement will be.

Since parameters are independent random variables that follow a Gaussian distribution, the multivariate Hermite polynomials are used in the analysis. In this respect, the random vector $\boldsymbol{Z}$ that obeys a jointly uniform distribution (with independent components) is transformed as a standard Gaussian random vector $\boldsymbol{X}$ by the following formula:

$$X_i = \Phi^{-1}(Z_i) \quad (i = 1, \dots, 7),$$

where Φ^{-1} denotes the inverse standard Gaussian cumulative distribution function. The initial experimental design of $\boldsymbol{Z}$ is generated by LHS with size 50; it is increased with the nested LHS technique. PC-GK-SBL, OPC-Kriging, and SPC-Kriging are applied on these experimental designs, which are directly obtained from UQLab. The RMSE in Equation (43) is computed, and the results are shown in Figure 9. It is found that, for a small sample size, the proposed method performs much better than OPC-Kriging and SPC-Kriging. Moreover, the proposed method and OPC-Kriging converge to almost the same accuracy with the increment of the sample size, where, in both cases, the major contributions will be properly captured.

6 | CONCLUSION

A major contribution of this paper is developing a novel surrogate modeling technique called PC-GK-SBL by combining PCE and the Gaussian kernel. The hybrid model here has the advantage of the global prediction ability of PCE and the local learning ability of the Gaussian kernel. It relies on a Bayesian method called SBL, which incorporates a parameterized prior to find sparse representatives for the output. Due to the Bayesian framework, not only the prediction response but also the prediction variance can be obtained. In particular, an optimization algorithm that measures the goodness of fit is introduced to identify the optimal surrogate models from a training set.

For validation purposes, several applications are considered. Three benchmark examples are first studied. A comparison with two PC-Kriging modeling methods is carried out. It is shown that the proposed method yields a more accurate and more robust surrogate model. As a second step, a simplified wing box structure model is investigated. It appears that the two methods converge to the same solution for a sufficiently large sample size, but faster convergence is achieved by the proposed method with a relatively small sample size.

Overall, it can be concluded that the proposed method achieves superior performance in terms of accuracy and robustness. The proposed method can still be improved by using an adaptive design of experiment based on its prediction variance. Note also that a local optimization method is used to update kernel widths of the Gaussian kernel; a future work will be devoted to incorporating this efficient surrogate modeling technique with a global optimization framework.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under grant 51775439 and sponsored by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under grant CX201935.

ORCID

Yicheng Zhou https://orcid.org/0000-0002-7170-3542
Kai Cheng https://orcid.org/0000-0003-1243-7093

REFERENCES

1. Fajraoui N, Marelli S, Sudret B. On optimal experimental designs for sparse polynomial chaos expansions. *J Uncertain Quantification*. 2017;5(1):1061-1085.
2. Feng K, Lu Z, Yun W. Aircraft icing severity analysis considering three uncertainty types. *AIAA Journal*. 2019. In press. https://doi.org/10.2514/1.J057529
3. Xiu D, Karniadakis GE. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput*. 2002;24(2):619-644.
4. Yaghoubi V, Marelli S, Sudret B, Abrahamsson T. Sparse polynomial chaos expansions of frequency response functions using stochastic frequency transformation. *Probabilistic Eng Mech*. 2017;48:39-58.
5. Abraham S, Raisee M, Ghorbaniasl G, Contino F, Lacor C. A robust and efficient stepwise regression method for building sparse polynomial chaos expansions. *J Comput Phys*. 2017;332:461-474.
6. Bryson DE, Rumpfkeil MP. All-at-once approach to multifidelity polynomial chaos expansion surrogate modeling. *Aerosp Sci Technol*. 2017;70:121-136.
7. Blatman G, Sudret B. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliab Eng Syst Saf*. 2010;95(11):1216-1229.
8. Cheng K, Lu Z. Adaptive sparse polynomial chaos expansions for global sensitivity analysis based on support vector regression. *Comput Struct*. 2018;194:86-96.
9. Marelli S, Sudret B. An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*. 2018;75:67-74.
10. Savin É, Faverjon B. Computation of higher-order moments of generalized polynomial chaos expansions. *Int J Numer Methods Eng*. 2017;111:1192-1200.
11. Raisee M, Kumar D, Lacor C. A non-intrusive model reduction approach for polynomial chaos expansion using proper orthogonal decomposition. *Int J Numer Methods Eng*. 2015;103(4):293-312.
12. Blatman G, Sudret B. Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *Comptes Rendus Mécanique*. 2008;336(6):518-523.
13. Blatman G, Sudret B. Adaptive sparse polynomial chaos expansion based on least angle regression. *J Comput Phys*. 2011;230(6):2345-2367.
14. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *Ann Stat*. 2004;32:407-499.
15. Marelli S, Sudret B. UQLab: a framework for uncertainty quantification in Matlab. In: Proceedings of the 2nd International Conference on Vulnerability, Risk Analysis and Management; 2014; Liverpool, UK.
16. Eldar YC, Kutyniok G, eds. *Compressed Sensing: Theory and Applications*. Cambridge, UK: Cambridge University Press; 2012.
17. Huan X, Safta C, Sargsyan K, et al. Compressive sensing with cross-validation and stop-sampling for sparse polynomial chaos expansions. *SIAM/ASA J Uncertain Quantification*. 2018;6:907-936.
18. Cheng K, Lu Z. Sparse polynomial chaos expansion based on D-MORPH regression. *Appl Math Comput*. 2018;323:17-30.
19. Diaz P, Doostan A, Hampton J. Sparse polynomial chaos expansions via compressed sensing and D-optimal design. *Comput Methods Appl Mech Eng*. 2018;336:640-666.

20. Gunn SR. *Support Vector Machines for Classification and Regression* [technical report]. Southampton, UK: Image Speech and Intelligent Systems, University of Southampton; 1997.
21. Li H-S, Zhao A-L, Tee KF. Structural reliability analysis of multiple limit state functions using multi-input multi-output support vector machine. *Adv Mech Eng*. 2016;8(10):1-11.
22. Smith M. *Neural Networks for Statistical Modeling*. New York, NY: Von Nostrand Reinhold; 1993.
23. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. *Statistical Science*. 1989;4:409-423.
24. Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press; 2006.
25. Couckuyt I, Koziel S, Dhaene T. Surrogate modeling of microwave structures using Kriging, co-Kriging, and space mapping. *Int J Numer Model Electron Netw Devices Fields*. 2012;26:64-73.
26. Schöbi R, Sudret B, Wiart J. Polynomial-chaos-based Kriging. *Int J Uncertain Quantification*. 2015;5(2):171-193.
27. Kersaudy P, Sudret B, Varsier N, Picon O, Wiart J. A new surrogate modeling technique combining Kriging and polynomial chaos expansions - application to uncertainty analysis in computational dosimetry. *J Comput Phys*. 2015;286:103-117.
28. Tipping ME. Sparse Bayesian learning and the relevance vector machine. *J Mach Learn Res*. 2001;1:211-244.
29. Wipf D, Rao BD. Sparse Bayesian learning for basis selection. *IEEE Trans Signal Process*. 2004;52(8):2153-2164.
30. Zhang Z, Rao BD. Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. *IEEE J Sel Top Signal Process*. 2011;5(5):912-926.
31. Nicolaou MA, Gunes H, Pantic M. Output-associative RVM regression for dimensional and continuous emotion prediction. *Image Vis Comput*. 2012;30(3):186-196.
32. Mohsenzadeh Y, Sheikhzadeg H, Reza AM, Bathaee N, Kalayeh MM. The relevance sample-feature machine: a sparse Bayesian learning approach to joint feature-sample selection. *IEEE Tans Cybern*. 2013;43(6):2241-2254.
33. Schoutens W. *Stochastic Processes and Orthogonal Polynomials*. New York, NY: Springer-Verlag; 2000.
34. Cherkassky V, Ma Y. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*. 2004;17(1):113-126.
35. Stone M. Cross-validatory choice and assessment of statistical predictions. *J R Stat Soc B Methodol*. 1974;111-147.
36. Wirtz D, Karajan N, Haasdonk B. Surrogate modeling of multiscale models using kernel methods. *Int J Numer Methods Eng*. 2014;101:1-28.
37. Smola AJ, Schökopf B. A tutorial on support vector regression. *Stat Comput*. 2004;14(3):199-222.
38. Suresh S, Narasimhan S, Sundararajan N. Adaptive control of nonlinear smart base-isolated buildings using Gaussian kernel functions. *Int J Numer Methods Eng*. 2008;15:585-603.
39. Tzikas DG, Likas AC, Galatsanos NP. Sparse Bayesian modeling with adaptive kernel learning. *IEEE Trans Neural Netw*. 2009;20(6):926-937.
40. Ji S, Xue Y, Carin L. Bayesian compressive sensing. *IEEE Trans Signal Process*. 2008;56(6):2346-2356.
41. Liu H, Cai J, Ong Y-S. An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error. *Comput Chem Eng*. 2017;106:171-182.
42. Lam CQ. *Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit* [doctoral dissertation]. Columbus, OH: The Ohio State University; 2008.
43. Konakli K, Sudret B. Polynomial meta-models with canonical low-rank approximations: numerical insights and comparison to sparse polynomial chaos expansions. *J Comput Phys*. 2016;321:1144-1169.
44. Stein M. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*. 1987;29(2):143-151.
45. Wang GG. Adaptive response surface method using inherited Latin hypercube design points. *J Mech Des*. 2003;125:210-220.
46. Blatman G, Sudret B. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Eng Mech*. 2010;25:183-197.
47. Xiong S, Qian PZG, Wu CFJ. Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*. 2013;55(1):37-46.
48. Forrester A, Sóbester A, Keane A. *Engineering Design via Surrogate Modeling: A Practical Guide*. Chichester, UK: John Wiley & Sons; 2008.
49. Moon H. *Design and Analysis of Computer Experiments for Screening Input Variables* [doctoral dissertation]. Columbus, OH: The Ohio State University; 2010.
50. Wang P, Lu Z, Xiao S. A generalized separation for the variance contributions of input variables and their distribution parameters. *Appl Math Model*. 2017;47:381-399.
51. Xiao S, Lu Z. Structural reliability sensitivity analysis based on classification of model output. *Aerosp Sci Technol*. 2017;71:52-61.
52. Roweis S. Matrix identities. 1999. http://www.cs.toronto.edu/~roweis/notes/matrixid.pdf

How to cite this article: Zhou Y, Lu Z, Cheng K. A new surrogate modeling method combining polynomial chaos expansion and Gaussian kernel in a sparse Bayesian learning framework. *Int J Numer Methods Eng*. 2019;120:498–516. https://doi.org/10.1002/nme.6145

APPENDIX

Lemma. *In the computation of the matrix,*[52] *for the derivative of the determinant of a positive definite symmetric matrix* $\boldsymbol{C}$ *on a logarithmic scale, we have*

$$\frac{\partial \boldsymbol{C}}{\partial \phi} ln \left|\det(\boldsymbol{C})\right| = \boldsymbol{C}^{-1}. \tag{A.1}$$

For the derivative of symmetric inverse matrix $\boldsymbol{C}^{-1}$*, we have*

$$\frac{\partial}{\partial \phi} \boldsymbol{t}^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{r} = -\boldsymbol{C}^{-1} \boldsymbol{t} \boldsymbol{r}^{\mathrm{T}} \boldsymbol{C}^{-1}, \tag{A.2}$$

where $\boldsymbol{t}$ *and* $\boldsymbol{r}$ *are two matrices that do not depend on* $\boldsymbol{C}$ *and* ϕ*.*