entropy

MDPI

Article

Gaussian Processes and Polynomial Chaos Expansion for Regression Problem: Linkage via the RKHS and Comparison via the KL Divergence

Liang Yan *, Xiaojun Duan, Bowen Liu and Jin Xu

College of Liberal Arts and Sciences, National University of Defense Technology, Changsha 410073, China; xjduan@nudt.edu.cn (X.D.); bowen_liu12@163.com (B.L.); xujin_nudt@163.com (J.X.)

* Correspondence: yanliang@nudt.edu.cn; Tel.: +86-0731-8700-1605

Received: 21 January 2018; Accepted: 12 March 2018; Published: 12 March 2018

Abstract: In this paper, we examine two widely-used approaches, the polynomial chaos expansion (PCE) and Gaussian process (GP) regression, for the development of surrogate models. The theoretical differences between the PCE and GP approximations are discussed. A state-of-the-art PCE approach is constructed based on high precision quadrature points; however, the need for truncation may result in potential precision loss; the GP approach performs well on small datasets and allows a fine and precise trade-off between fitting the data and smoothing, but its overall performance depends largely on the training dataset. The reproducing kernel Hilbert space (RKHS) and Mercer's theorem are introduced to form a linkage between the two methods. The theorem has proven that the two surrogates can be embedded in two isomorphic RKHS, by which we propose a novel method named Gaussian process on polynomial chaos basis (GPCB) that incorporates the PCE and GP. A theoretical comparison is made between the PCE and GPCB with the help of the Kullback–Leibler divergence. We present that the GPCB is as stable and accurate as the PCE method. Furthermore, the GPCB is a one-step Bayesian method that chooses the best subset of RKHS in which the true function should lie, while the PCE method requires an adaptive procedure. Simulations of 1D and 2D benchmark functions show that GPCB outperforms both the PCE and classical GP methods. In order to solve high dimensional problems, a random sample scheme with a constructive design (i.e., tensor product of quadrature points) is proposed to generate a valid training dataset for the GPCB method. This approach utilizes the nature of the high numerical accuracy underlying the quadrature points while ensuring the computational feasibility. Finally, the experimental results show that our sample strategy has a higher accuracy than classical experimental designs; meanwhile, it is suitable for solving high dimensional problems.

Keywords: Gaussian process; polynomial chaos expansion; reproducing kernel Hilbert space; Kullback–Leibler divergence; experimental design

1. Introduction

Computer simulations are widely used in learning tasks, where a single simulation is an instance of the system [1,2]. A simple approach to the learning task is to randomly sample input variables and run the simulations for each input to obtain the features of the systems. Similar approaches are utilized in Monte Carlo techniques [3]. However, even a single simulation can be computationally costly due to its high complexity, and so, obtaining a trustworthy result via sufficient simulations becomes intractable. Mathematical methods and statistical theorems are introduced to generate surrogate models to replace the simulations, especially when dealing with complex systems with many parameters [4,5]. Although the main drawback of surrogate models is that only approximations can be obtained, they are computationally efficient whilst maintaining the essential information of the systems, hence analyzing the properties of the system. Attempting to construct surrogate models with an acceptable number of simulations necessitates the development of robust techniques to determine their reliability and validity [6–8]. Plenty of researchers are working on improving sampling strategies to decrease the number of simulations, which makes the task more significant [9]. With an increasing number of surrogate models being developed, there needs to be a comprehensive understanding of the uncertainties introduced by those models. The main purpose of uncertainty quantification (UQ) is to establish a relationship between input and output, i.e., the propagation of input uncertainties, and then to quantify the difference between surrogate models and original simulations. UQ can provide a measure of the surrogate model's accuracy and an indication of how to update the model at the same time [10–12].

Denote f as a function (or simulator) of the original system, then given experimental design $\boldsymbol{X}$, the output $Y = f(\boldsymbol{X})$ is produced, where caption notation is used because the input and output are usually vectors (or matrices) in simulations. From a statistical perspective, the input uncertainties are introduced by their randomness, so we represent the input with a random variable $\boldsymbol{x}$, whose prior probability density function (PDF) is $p(\boldsymbol{x})$, such as the multivariate Gaussian distribution; as for the output uncertainties, a common technique is to integrate the system uncertainty and the approximation error as a noise term $\boldsymbol{\epsilon}$. In fact, the output $\boldsymbol{y}$ is also a random variable $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$ determined by $f, \boldsymbol{x}$ and $\boldsymbol{\epsilon}$. Now, suppose a surrogate $\bar{f}(\boldsymbol{x})$ is constructed to approximate $f(\boldsymbol{x})$, then UQ is used to identify the distribution and statistical features (for example, Kullback–Leibler divergence) of $\boldsymbol{y}$, which are essential to the validation and verification of surrogates. Basically, there are two preconditions that need to be satisfied: firstly, the surrogate models are well defined, i.e., any $\bar{f}$ is a measurable function with respect to (w.r.t) corresponding probability space $p(\boldsymbol{x})$; secondly, techniques are needed that learn from the prior information to obtain the best guess of the true function.

There is a number of studies proposing different surrogates for specific applications in the literature, such as multivariate adaptive regression splines (MARS) [13], support vector regression (SVR) [14], artificial neural network (ANN) [15] for reliability and sensitivity analyses and kriging [16] for structural reliability analysis. We mainly focus on two popular methods that have been extensively studied recently. One popular method that is extensively studied in the literature is the polynomial chaos expansion (PCE), also known as a spectral approach [17]. PCE aims to represent an arbitrary random variable of interest as a spectral expansion function of other random variables with prior PDF. Xiu et al. [18–20] have generalized the PCE in terms of the Askey scheme of polynomials, so the surrogates can be expressed by a series of orthogonal polynomials w.r.t the distributions of the input variables. These polynomials can be extended as a basis of a polynomial space. In general, methods used to solve PCE problems are categorized as two types: intrusive and non-intrusive. The main idea behind the intrusive methods is the substitution of the input $\boldsymbol{x}$ and output $f(\boldsymbol{x})$ with the truncated PCE and calculating the coefficients with the help of Galerkin projection [21]. However, the explicit formation of f is required to compose the Galerkin system, and a specific algorithm or program is needed to solve a particular problem. It is for these reasons that the intrusive models are not widely used; non-intrusive methods have been developed to avoid these limitations [21,22]. There are two main aspects of the non-intrusive methods: one is the choice of sampling strategies, for example Monte Carlo techniques; the other one is computational approaches. These two aspects are not independent of each other: for example, if $\boldsymbol{x} \sim N(0,1)$, then the Gaussian quadrature method is introduced to solve the numerical integration and $\boldsymbol{X}$ is the set of corresponding quadrature points. Another one of the more common methods in constructing surrogate models is the Gaussian process (GP), which is actually a Bayesian approach. Instead of attempting to identify a specific real model of the system, the GP method provides a posterior distribution over the model in order to make robust predictions about the system. As described in the highly influential works [23–26], the GP can be treated as a distribution over functions with properties controlled by a kernel. For the two prerequisites discussed in the previous paragraph, the GP generates a surrogate model that lies in a space spanned by kernels; meanwhile, Bayesian linear regression or classification methods are introduced to utilize the prior information.

Both the PCE and GP methods build surrogates, but there are some differences between them. The PCE method builds surrogates of a random variable $\boldsymbol{y}$ as a function of another prior random variable $\boldsymbol{x}$ rather than the distribution density function itself. The PCE surrogates are based on the orthogonal polynomial basis corresponding to the $p(\boldsymbol{x})$, so it is simple to obtain the mean and standard deviation of $\boldsymbol{y}$. In contrast, the GP utilizes the covariance information so that it performs better in capturing the local features. Although both the PCE and GP approaches are feasible methods to compute the mean and standard deviation of $\boldsymbol{y}$, the PCE performs more efficiently than the GP method.

As mentioned above, both the PCE and GP methods have their own trade-offs to consider when building surrogates, and there exists a connection to be explored. According to Paul Constantine's work [27], ordinary kriging (i.e., GP in geostatistics) interpolation can be viewed as a transformed version of the least squares (LS) problem, and the PCE can be viewed as the least squares with selected basis and weights. However, the GP reverts to interpolation when the noise term is zero. When taking the noise term into consideration, the Gaussian process with the kernel (i.e., covariance matrix) $X^T X$ can be viewed as a ridge regression problem [28] with a regularization term. Furthermore, different numerical methods can affect the precision of the PCE method, as well. For example, Xiu [20] analyzed the aliasing error w.r.t the projection method and interpolation method. Thus, the inherent connection of the two models cannot be simply summarized as an LS solution, and how to output a model with high precision remains an interesting question.

There are connections between the PCE and GP methods that have been explored by R. Schobi, etc. They introduced a new meta-modeling method naming PC-kriging [29] (polynomial-chaos-based kriging) to solve the problems like rare event estimation [30], structural reliability analysis [31], quantile estimation [32], etc. In their papers, the PCE models can be viewed as a special form of GP where a Dirac function is introduced as the kernel. They also proposed the idea that the PCE models have better performance in capturing the global features and that the GP models approximate the local characteristics. We would like to describe the PC-kriging method as a GP model with a PCE-form trend function along with a noise term. The global features are dominated by the PCE trend, and local structures (residuals) are approximated by the ordinary GP process. The PC-kriging model thus introduces the coefficients as parameters to be optimized, and the solution can be derived by Bayesian linear regression with the basis consisting of the PCE polynomials. They also use the LARSalgorithms to calibrate the model and to select a sparse design. They construct a rigid framework to optimize the parameters, validate and calibrate the model and evaluate the model accuracy.

Unlike the PC-kriging, which takes the PCE as a trend, this paper focuses on the construction of the kernel in the GP to solve the regression problems, through which we can combine the two methods into a unified framework, unifying positive aspects from both and in so doing refining the surrogates. In other words, we wish to find the connection between the GP and the PCE by analyzing the attribution of their solutions, and we want to propose a new approach to achieve high-precision predictions. The main idea of this paper is described as follows. Firstly, the PCE surrogate is embedded in a Hilbert space whose bases are the orthonormal polynomials themselves, then a suitable inner product and a Mercer kernel [33] are defined to build a reproducing kernel Hilbert space (RKHS) [33]. Secondly, on the other hand, the kernel of the GP can be de-composited as the product of eigenfunctions, and we can define an inner product to generate a RKHS, as well. We have explicitly elaborated the two procedures respectively and proven that the two RKHS are isometrically isomorphic. Hence, a connection between these two approaches has been established via RKHS. Furthermore, we can obtain a solution of the PCE model by solving a GP model with the Mercer kernel w.r.t the PCE polynomial basis. We name this approach Gaussian processes on polynomial chaos basis (GPCB). In order to illustrate the capability of the GPCB method, we use the Kullback–Leibler divergence [34] to explicitly compare the PDFs of the posterior prediction of the GPCB and PCE method. Provided that the true function can be approximated by a finite number of PCE bases, it can be concluded that the GPCB can converge to the optimal subset of the RKHS wherein the true function lies.

The experimental design from the PCE model, i.e., the full tensor product of quadratures in each dimension, is used in the GPCB. We have overcome two concerns about the PCE and GP, respectively. Firstly, the PCE is based on a truncated polynomial basis, while the GPCB keeps all polynomials, which can be regarded as maintaining information in every feature. Secondly, the GP's behavior depends on the experimental design; however, it often achieves the optimal result in local small datasets. The quadrature points derived from the PCE model are distributed evenly in the input space, and those points have high numerical precision w.r.t the polynomial basis; hence, they can work well with the GPCB. However, we must admit that the GPCB is still a GP approach, so when the dimension of input variables grows, the computational burden is on the table. In order to cope with the high dimensional problems, sampling strategies to lower the number of experimental designs are put on the table. The AK-MCSmethod [35] is a useful tool that adaptively selects new experimental designs; however, the experimental design tends to validate the selected surrogate model. We propose a new method that is model-free and that makes full use of the quadrature points. We randomly choose a sparse subset from the quadrature points to form a new experimental design while maintaining the accuracy. Several classical sampling strategies like MC, Halton and LHS are introduced to compare their capabilities. Our sample scheme has superior performance under the conditions in this paper. The GPCB is a novel method to build surrogate models, and it can be used for various physical problems such as reliability analysis and risk assessment.

This paper is divided into two parts. In Part 1, we discuss the mathematical rigor of the method: a brief summary of PCE and GP is presented in Section 2; the reproducing kernel Hilbert space (RKHS) is introduced to connect these two methods in Section 3; the GPCB method is proposed based on the discussion in Section 4; meanwhile, a theoretical Kullback–Leibler divergence between the GPCB and PCE method is demonstrated. In Part 2, an explicit Mehler kernel is presented with the Hermite polynomial basis in the last part of Section 4; several tests of the GPCB with some benchmark functions are presented in Section 5, along with the random constructive sampling method for high dimensional problems.

2. Brief Review of PCE and GP

Firstly, we want to have a clear idea of how the PCE and GP work under the circumstances that, e.g., only samples of input $\boldsymbol{X}$ and output Y are obtained. Different assumptions are made to cope with PCE and GP, respectively, and the processing procedures are presented in the following subsections.

2.1. Polynomial Chaos Expansion

Just as discussed in Section 2.2, the output is assumed to be represented by a model $y = f(\boldsymbol{x}) + \epsilon$, where $f(\boldsymbol{x}) : \boldsymbol{x} \in \Omega \to \mathbb{R}$ is the real function underneath and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Here, we define $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$ as a d-dimensional vector of independent random variables in a bounded domain $\Omega \subset \mathbb{R}^d$. Suppose $\{x_i,\ i = 1, \ldots, d\}$ are independent and identically distributed; the joint PDF has the form $p(\boldsymbol{x}) = \prod_{i=1}^d p(x_i)$. In the context of PCE, we aim to seek a surrogate of the model $f(\boldsymbol{x})$ as an expansion of a series of orthonormal polynomials $\boldsymbol{\phi}_\alpha(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = \sum_{\alpha \in \mathbb{N}^d} \beta_\alpha \boldsymbol{\phi}_\alpha(\boldsymbol{x}), \alpha = \{\alpha_1, \ldots, \alpha_d\} \tag{1}$$

where α is the multi-index, $\boldsymbol{\phi}_\alpha(\boldsymbol{x}) = \prod_{i=1}^d \phi_{\alpha_i}^{(i)}(x_i)$, $\int_{\Omega_i} \phi_m^{(i)}(x_i)\phi_n^{(i)}(x_i)p_i(x_i)dx_i = \delta_{mn}$ with Ω_i the marginal domain of Ω and δ_{mn} the Kronecker delta. Xiu et al. [20] have summarized various correspondences between the distribution and polynomial basis to form generalized polynomial chaos.

It is proven that the original model $f(\boldsymbol{x})$ can be approximated to any degree of accuracy in a strong sense [20], e.g., mean-square norm $\|f(\boldsymbol{x}) - \sum_{\alpha \in \mathbb{N}^D} \beta_\alpha \boldsymbol{\phi}_\alpha(\boldsymbol{x})\|$ in an L^2 norm defined on Ω, although f is not necessarily the span of orthonormal polynomial bases. Since we are unable to calculate an infinite series, the truncation scheme corresponding to multi-index α is introduced such that we can rearrange the polynomials. For simplicity, we can rewrite Equation (1) in the following form:

$$f(\boldsymbol{x}) \approxeq \sum_{l=0}^{M} \beta_l \boldsymbol{\phi}_l(\boldsymbol{x}) \triangleq f_P, \tag{2}$$

We can simply solve the above system via the ordinary least squares method or the non-intrusive method. Specifically, we focus on the non-intrusive projection method, whereby we can directly obtain the coefficients by taking the expectation value of Equation (2) multiplied by $\boldsymbol{\phi}_l(\boldsymbol{x})$:

$$\beta_l = \int f(\boldsymbol{x})\boldsymbol{\phi}_l(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \approx \sum_{i=1}^{N} \omega_i f(X_i)\boldsymbol{\phi}_l(X_i), l = 0, \ldots, M \tag{3}$$

where the second equation is derived by the numerical integration techniques, such as the Gaussian quadrature rule, and $\{X_i, i = 1, \ldots, N\}$ and $\{\omega_i, i = 1, \ldots, N\}$ are the corresponding nodes and weights. The integration is exact when $f(\boldsymbol{x})$ is of polynomial complexity. Together with Equations (2) and (3), $f_P(\boldsymbol{x})$ has the form:

$$f_P(\boldsymbol{x}) \approx \sum_{l=0}^{M} \left[\sum_{i=1}^{N} \omega_i f(X_i)\boldsymbol{\phi}_l(X_i) \right] \boldsymbol{\phi}_l(\boldsymbol{x}) \triangleq \sum_{l=0}^{M} \beta_l \boldsymbol{\phi}_l(\boldsymbol{x}). \tag{4}$$

$\{f(X_i), i = 1, \ldots, N\}$ remain unknown to us, and usually, they are substituted by $\{Y_i, i = 1, \ldots, N\}$. Note $Y_i = f(X_i) + \epsilon_i$, so such a substitution will introduce noise into the surrogate; hence, the approximation error is neglected as a source of uncertainty.

2.2. Gaussian Process Regression

The analysis of the Gaussian process regression model [26] is reviewed in this section. A Gaussian prior is placed over function $f(\boldsymbol{x})$, i.e., $f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}))$, where m is the mean function and k is the kernel function, which is positive semi-definite bounded. More specifically, let $\boldsymbol{X} = \{X_i,\ i = 1, \ldots, N\} \in \Omega^N$ be the input data, and let $Y = \{Y_i, i = 1, \ldots, N\} \in \mathbb{R}^N$ be the output data, then we have $Y = f(\boldsymbol{X}) + \boldsymbol{\epsilon}$ with $f(\boldsymbol{X}) \sim \mathcal{N}(m(\boldsymbol{X}), k(\boldsymbol{X}, \boldsymbol{X}))$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$. Bear in mind that the mathematical expression of $f(\boldsymbol{x})$ is implicit, so $f(\boldsymbol{x})$ is approximated to achieve the best guess prediction f_G in the statistical sense. With the help of Bayes' theorem, prediction and corresponding variance at a new point $\boldsymbol{x}$ can be obtained by the following equations [36]:

$$\begin{aligned} p_G(f(\boldsymbol{x})|Y, \boldsymbol{X}, \boldsymbol{x}, \boldsymbol{\theta}) &= \mathcal{N}(f_G(\boldsymbol{x}), cov(f_G(\boldsymbol{x}))), \\ f_G(\boldsymbol{x}) \triangleq \mathbb{E}[f(\boldsymbol{x})|Y, \boldsymbol{X}, \boldsymbol{x}, \boldsymbol{\theta}] &= K_{\boldsymbol{x}}^T [K + \sigma_\epsilon^2 I]^{-1} Y, \\ cov(f_G(\boldsymbol{x})) &= K_{\boldsymbol{xx}} - K_{\boldsymbol{x}}^T [K + \sigma_\epsilon^2 I]^{-1} K_{\boldsymbol{x}}. \end{aligned} \tag{5}$$

where $K = k(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$ as the covariance matrix with $K_{ij} = k(X_i, X_j)$ and $K_{\boldsymbol{x}} = k(\boldsymbol{X}, \boldsymbol{x}) \in \mathbb{R}^{N \times 1}$, $K_{\boldsymbol{xx}} = k(\boldsymbol{x}, \boldsymbol{x}) \in \mathbb{R}$ are defined similarly. Note that Equation (5) shows that the mean value of the posterior distribution can be expressed as a linear combination of N kernel functions as follows:

$$f_G(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i k(X_i, \boldsymbol{x}), \quad \alpha = (K + \sigma_\epsilon^2 I)^{-1} Y \tag{6}$$

3. Links between the PCE and GP

The basic concepts of PCE and GP are discussed in Section 2. GP generates a surrogate based on Bayes' theorem and the Gaussian hypothesis; however, it is controlled by the kernel function and the experimental design and usually does not utilize prior distribution information; PCE substitutes the model with orthonormal polynomials, which is more computational efficient, but performs badly when facing noisy or big data. This section aims to build a connection between PCE and GP; hence, they can be studied in the same structure and be combined to improve the performance of the surrogates. The reproducing kernel Hilbert space will be of great help to build such a bridge, and we are going to present it as follows.

3.1. Generate an RKHS from a Mercer Kernel Constructed by the PCE Basis

We have obtained a complete orthonormal basis $\{\boldsymbol{\phi}_l\}$ of Hilbert space $\mathcal{H} := span\{\boldsymbol{\phi}_l(\boldsymbol{x})\}$ with inner product $< f(\boldsymbol{x}), g(\boldsymbol{x}) >= \int f(\boldsymbol{x})g(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$ in Section 2.1. We can see that the PCE method generates surrogates, which are actually a linear combination of $\{\boldsymbol{\phi}_l\}$, so there exists a unique expansion $f = \sum_l f_l \boldsymbol{\phi}_l \in \mathcal{H}$. According to Mercer's theorem [33], we aim to define a kernel having the following form:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_l \lambda_l \boldsymbol{\phi}_l(\boldsymbol{x})\boldsymbol{\phi}_l(\boldsymbol{x}') \quad s.t. \quad k(\boldsymbol{x}, \boldsymbol{x}) < \infty \quad for \quad \forall \boldsymbol{x} \in \Omega. \tag{7}$$

If we have positive weights λ_l that satisfy $\sum_l \lambda_l \boldsymbol{\phi}_l^2(\boldsymbol{x}) < \infty$, then for any $\boldsymbol{x} \in \Omega$, together with the Cauchy–Schwarz inequality, we have:

$$|f(\boldsymbol{x})| \leq \sqrt{\sum_l \frac{< f(\boldsymbol{x}), \boldsymbol{\phi}_l(\boldsymbol{x}) >^2}{\lambda_l}} \sqrt{\sum_l \lambda_l \boldsymbol{\phi}_l^2(\boldsymbol{x})}. \tag{8}$$

$|f(\boldsymbol{x})|$ is point-wise bounded because $f(\boldsymbol{x}) \in \mathcal{H}$ for any $\boldsymbol{x} \in \Omega$. By checking the right side of the above inequality, the second term is ensured in advance, then $f(\boldsymbol{x})$ lies in a subspace of $\mathcal{H}$ such that:

$$\mathcal{H}_P = \left\{ f \in \mathcal{H} \,\middle|\, < f, f >_{\mathcal{H}_P} = \sum_l \frac{< f(\boldsymbol{x}), \boldsymbol{\phi}_l(\boldsymbol{x}) >^2}{\lambda_l} < \infty, \sum_l \lambda_l \boldsymbol{\phi}_l^2(\boldsymbol{x}) < \infty \right\}. \tag{9}$$

Proposition 1. *$\mathcal{H}_P$ defined in Equation (9) is an RKHS with Mercer kernel defined in Equation (7).*

3.2. Generate an RKHS from the Reproducing Kernel Map Construction

We aim to compose a space of functions in which all the GP surrogates are embedded. Given Equation (6) and an arbitrary experimental design $\boldsymbol{X}$, define a space of functions as follows:

$$\mathcal{H}'_G = \left\{ f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i k(\boldsymbol{x}, X_i) \,\middle|\, N \in \mathbb{N}, \boldsymbol{X} \in \Omega^N, \boldsymbol{x} \in \Omega, f_i \in \mathbb{R}, \sum_{i=1}^{N}\sum_{j=1}^{N} f_i f_j k(X_i, X_j) < +\infty \right\}. \tag{10}$$

Proposition 2. *$\mathcal{H}'_G$ is a pre-Hilbert space with the inner product $< \cdot, \cdot >_{\mathcal{H}'_G}$*

Now that $\mathcal{H}'_G$ is a pre-Hilbert space and given the norm $\|f(\boldsymbol{x})\|_{\mathcal{H}'_G} = \sqrt{< f(\boldsymbol{x}), f(\boldsymbol{x}) >_{\mathcal{H}'_G}}$, we can define a closure of $\mathcal{H}'_G$ as $\mathcal{H}_G$ derived by the classical Hilbert space theory. This is an abstract space where the norm of $\mathcal{H}'_G$ extends to the closure $\mathcal{H}_G$. Thus, we have a Hilbert space $\mathcal{H}_G$.

Proposition 3. *$\mathcal{H}_G$ defined above is the unique RKHS of the kernel $k(\cdot, \cdot)$.*

3.3. Reproducing Kernel Hilbert Spaces as a Linkage

$\mathcal{H}_P$ and $\mathcal{H}_G$ are RKHS with the Mercer kernel and GP kernel, respectively. We are going to investigate the relationship between the two RKHS, by which we can discuss the two approaches in a unified structure. Let $\boldsymbol{X}$ be a sample set and GP kernel $k(\cdot,\cdot)$ be a real positive semi-definite kernel, then according to Mercer's theorem, $k(X_i, X_j)$ has an eigenfunction expansion:

$$k(X_i, X_j) = \sum_l \lambda_l \boldsymbol{\phi}_l(X_i)\boldsymbol{\phi}_l(X_j), \tag{11}$$

where the eigenfunctions $\{\boldsymbol{\phi}_l\}$ are orthonormal, i.e., $<\boldsymbol{\phi}_l, \boldsymbol{\phi}_{l'}>= \int \boldsymbol{\phi}_l(\boldsymbol{x})\boldsymbol{\phi}_{l'}(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \delta_{ll'}$ and $\{\lambda_l, \boldsymbol{\phi}_l\}$ satisfies $\sum_l \lambda_l \boldsymbol{\phi}_l^2(\boldsymbol{x}) < \infty$. Let $f_{\boldsymbol{X}}(\boldsymbol{x}) \in \mathcal{H}_G$ with experimental design $\boldsymbol{X}$, then we can rewrite it according to Equations (10) and (11):

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{i=1}^{N} f_i \sum_l \lambda_l \boldsymbol{\phi}_l(X_i)\boldsymbol{\phi}_l(\boldsymbol{x}) = \sum_l \left[\lambda_l \sum_{i=1}^{N} f_i \boldsymbol{\phi}_l(X_i)\right] \boldsymbol{\phi}_l(\boldsymbol{x}) \triangleq \sum_l c_l(\boldsymbol{X})\boldsymbol{\phi}_l(\boldsymbol{x}). \tag{12}$$

where $c_l(\boldsymbol{X})$ is identically determined by l and $\boldsymbol{X}$, and it has a similar form as a function lies in $\mathcal{H}_P$. Actually, given $f_{\boldsymbol{X}}(\boldsymbol{x}), g_{\boldsymbol{X}'}(\boldsymbol{x}) \in \mathcal{H}_G$, we have:

$$\begin{aligned}
< f_{\boldsymbol{X}}(\boldsymbol{x}), g_{\boldsymbol{X}'}(\boldsymbol{x}) >_{\mathcal{H}_G} &= \sum_{i=1}^{N}\sum_{j=1}^{N'} f_i g_j \left[\sum_l \lambda_l \boldsymbol{\phi}_l(X_i)\lambda_l \boldsymbol{\phi}_l(X'_j)/\lambda_l\right] \\
&= \sum_i \left\{\left[\lambda_l \sum_{i=1}^{N} f_i \boldsymbol{\phi}_l(X_i)\right]\left[\lambda_l \sum_{j=1}^{N'} g_j \boldsymbol{\phi}_l(X'_j)\right]\right\}/\lambda_l \\
&= \sum_l c_l(\boldsymbol{X})c_l(\boldsymbol{X'})/\lambda_l \\
&= \sum_l < f_{\boldsymbol{X}}(\boldsymbol{x}), \boldsymbol{\phi}_l >< g_{\boldsymbol{X}'}(\boldsymbol{x}), \boldsymbol{\phi}_l > /\lambda_l \\
&=< f_{\boldsymbol{X}}(\boldsymbol{x}), g_{\boldsymbol{X}'}(\boldsymbol{x}) >_{\mathcal{H}_P} .
\end{aligned} \tag{13}$$

The above equation gives us the information that their inner product stands in $\mathcal{H}_P$, as well, so we can conclude that $f_{\boldsymbol{X}}$ lies in $\mathcal{H}_P$. It also shows us that the two inner products are equivalent. Next, we are going to propose a rigid theorem to prove that the two spaces are isometrically isomorphic.

Theorem 1. *The reproducing kernel Hilbert space $\mathcal{H}_G$ of a given kernel k is isometrically isomorphic to the space $\mathcal{H}_P$.*

According to the proof of Theorem 1 in Appendix A.4, it is reasonable to introduce a weighted l^2 space $l^2_{1/\lambda}$ because it is difficult to find a direct linear map between $\mathcal{H}_G$ (where c_l varies according to $\boldsymbol{X}$ and k) and $\mathcal{H}_P$ (where c_l varies according to the distribution of $\boldsymbol{x}$). Figure 1 shows two flowcharts used to describe different processes in generating the RKHS.

Furthermore, the GP prediction is a combination of the kernel functions, which consist of infinite eigenfunctions, while the PCE prediction is always a combination of finite polynomial bases. The Kullback–Leibler divergence (KL divergence) is a useful criterion to indicate the performance of different surrogate models. We are going to present the comparison of the GPCB and PCE methods with the help of KL divergence in the next section.

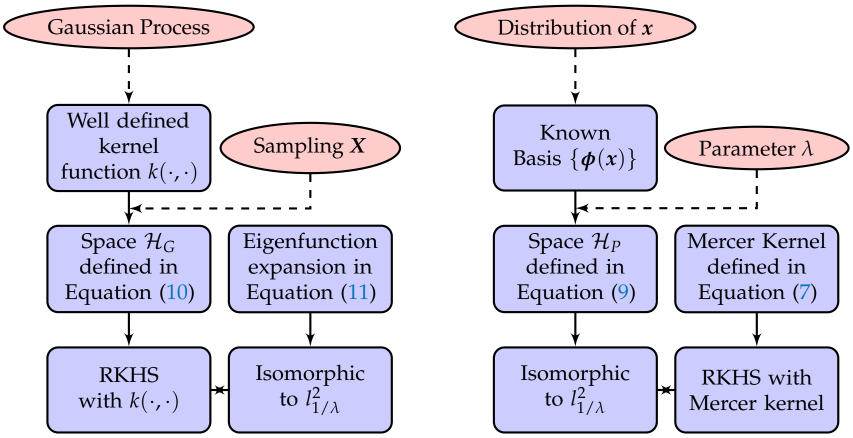

Gaussian Process
Well defined kernel function $k(\cdot,\cdot)$
Sampling $\boldsymbol{X}$
Space $\mathcal{H}_G$ defined in Equation (10)
Eigenfunction expansion in Equation (11)
RKHS with $k(\cdot,\cdot)$
Isomorphic to $l^2_{1/\lambda}$
Distribution of $\boldsymbol{x}$
Known Basis $\{\boldsymbol{\phi}(\boldsymbol{x})\}$
Parameter λ
Space $\mathcal{H}_P$ defined in Equation (9)
Mercer Kernel defined in Equation (7)
Isomorphic to $l^2_{1/\lambda}$
RKHS with Mercer kernel

Figure 1. Left: Generate the reproducing kernel Hilbert space (RKHS) with the reproducing kernel map; right: generate the reproducing Mercer kernels with the polynomial chaos expansion (PCE) basis.

4. Gaussian Process on Polynomial Chaos Basis

$\mathcal{H}_G$ and $\mathcal{H}_P$ are isomorphic as discussed in previous Section 3, so it is natural to come up with the idea that GP can be conducted with $k(\cdot,\cdot)$ as the Mercer Kernel generated by polynomial basis in the PCE, and the new model is called Gaussian process on polynomial chaos basis (GPCB). In fact, the GPCB generates a PCE-like model, but with a different philosophy. Note that the posterior distribution of the predictions regarding experimental design $\{\boldsymbol{X}, Y\}$ can be calculated analytically, so we are able to compute the KL divergence as well, which are presented as follows.

4.1. Comparison of the PCE and GPCB with the Kullback–Leibler Divergence

The true distribution of the system is always implicit in practice. Without loss of generality, the underlying true system is assumed to be $f_{\bar{P}}(\boldsymbol{x}) = \sum_{l=0}^{\bar{M}} \bar{\beta}_l \boldsymbol{\phi}_l(\boldsymbol{x})$ if $f_{\bar{P}}(\boldsymbol{x}) \in C^0(\bar{\Omega})$ such that it can be approximated by the polynomials to any degree of accuracy [20].

Firstly, we presume that $\bar{M} \leq M$, i.e., $\boldsymbol{\beta}$ in Equation (4) is an unbiased estimator of $\bar{\boldsymbol{\beta}}$. Hence, the PCE approximation can be considered as a precise approximate of the true function. We compare the performance of the GPCB and the PCE method by comparing their difference in the posterior distribution of the prediction. It is known that given experimental design $\{\boldsymbol{X}, Y\}$ and kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{l=0}^{\infty} \lambda_l \boldsymbol{\phi}_l(\boldsymbol{x}) \boldsymbol{\phi}_l(\boldsymbol{x}')$, the distribution of the prediction of the GPCB reads:

$$p_G(f_G(\boldsymbol{x})) = \mathcal{N}(K_{\boldsymbol{x}}^T K_Y^{-1} Y, K_{\boldsymbol{xx}} - K_{\boldsymbol{x}}^T K_Y^{-1} K_{\boldsymbol{x}}) \triangleq \mathcal{N}(\mu_1, \Sigma_1), \tag{14}$$

where conditions $\boldsymbol{X}, Y, \boldsymbol{x}$ are dropped in $p_G(f_G(\boldsymbol{x})|\boldsymbol{X}, Y, \boldsymbol{x})$ for simplicity and $K_Y = K + \sigma^2 I$. Similarly, the prediction of the PCE with the projection method is $f_P(\boldsymbol{x}) = \boldsymbol{\phi}\Phi^T W Y$, which is derived from the estimation $\boldsymbol{\beta} = \Phi^T W Y$. Here, $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{1\times(P+1)}$, $\Phi = \boldsymbol{\phi}(\boldsymbol{X}) \in \mathbb{R}^{N\times(P+1)}$, $W = diag\{\omega_1, \ldots, \omega_N\}$ is a diagonal matrix. The corresponding prediction variance is $cov(f_P(\boldsymbol{x})) = \boldsymbol{\phi} cov(\boldsymbol{\beta}) \boldsymbol{\phi}^T = \sigma^2 \boldsymbol{\phi}\Phi^T W^2 \Phi \boldsymbol{\phi}^T$. Dropping the conditions in $p_P(f_P(\boldsymbol{x})|\boldsymbol{X}, Y, \boldsymbol{x})$ as well, the previous results indicate:

$$p_P(f_P(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{\phi}\Phi^T W Y, \sigma^2 \boldsymbol{\phi}\Phi^T W^2 \Phi \boldsymbol{\phi}^T) \triangleq \mathcal{N}(\mu_2, \Sigma_2). \tag{15}$$

We can evaluate the discrepancy between $p_G(f_G(\boldsymbol{x}))$ and $p_P(f_P(\boldsymbol{x}))$, hence comparing their performance. The KL divergence can be calculated analytically:

$$D_{KL}(p_P, p_G) = \frac{1}{2}\left(-1 + \frac{\Sigma_2}{\Sigma_1} + \log\frac{\Sigma_1}{\Sigma_2} + \frac{(\mu_1 - \mu_2)^2}{\Sigma_1}\right) \triangleq \frac{1}{2}\left(-1 + b - \log b + \frac{a^2}{\Sigma_2} b\right), \tag{16}$$

where a, b are simplified notations for the corresponding parts in Equation (16). In fact, b is the point-wise ratio between the posterior variances of the predictions, and a represents the difference between the posterior mean of the prediction. We discuss the properties of D_{KL} starting from a special case to the general conditions hereafter.

Let k be a truncated kernel with the M basis of PCE, i.e., $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{l=0}^{M} \lambda_l \boldsymbol{\phi}_l(\boldsymbol{x}) \boldsymbol{\phi}_l(\boldsymbol{x}')$. Actually, it can be seen as the assignment of $\{\lambda_l, l > M\}$ with the value of zero, which can be achieved by optimizing the value λ_l with a specific procedure. b can be simplified as:

$$
\begin{aligned}
b &= \frac{\sigma^2 \boldsymbol{\phi} \Phi^T W^2 \Phi \boldsymbol{\phi}^T}{\boldsymbol{\phi}\left(\Lambda - \Lambda \Phi^T K_Y^{-1} \Phi \Lambda\right) \boldsymbol{\phi}^T} = \frac{\boldsymbol{\phi} \Phi^T W^2 \Phi \boldsymbol{\phi}^T}{\boldsymbol{\phi} \Phi^T W K_Y^{-1} K W \Phi \boldsymbol{\phi}^T} \\
&= \frac{\boldsymbol{\phi} \Phi^T W^2 \Phi \boldsymbol{\phi}^T}{\boldsymbol{\phi} \Phi^T W U (S + \sigma^2 I)^{-1} U^T U S U^T W \Phi \boldsymbol{\phi}^T} \in \left[\frac{s_{\max} + \sigma_\epsilon^2}{s_{\max}}, \frac{s_{\min} + \sigma_\epsilon^2}{s_{\min}}\right],
\end{aligned}
\tag{17}
$$

where $\Lambda = diag\{\lambda_l\}$ is a diagonal matrix and $K = USU^T$ is the eigenvalue decomposition. Let $s_{\max}$ be the maximum eigenvalue and $s_{\min}$ be the minimal one; the above interval holds because K is a positive definite matrix. Note that b is an invariant with fixed $\boldsymbol{x}$. On the other hand, the distribution of a is as follows:

$$
\begin{aligned}
a &= \boldsymbol{\phi}\left(\Phi^T W Y - \Lambda \Phi^T K_Y^{-1} Y\right) = \sigma_\epsilon^2 \boldsymbol{\phi} \Phi^T W K_Y^{-1} Y \\
&\sim \mathcal{N}\left(\sigma_\epsilon^2 \boldsymbol{\phi} \Phi^T W K_Y^{-1} \hat{Y}, \sigma_\epsilon^6 \boldsymbol{\phi} \Phi^T W K_Y^{-1} K_Y^{-1} W \Phi \boldsymbol{\phi}^T\right).
\end{aligned}
\tag{18}
$$

Here, $\hat{Y}$ denotes the mean value of the observations, i.e., the true response. It is necessary to state that the randomness of a is brought by the random variable $\boldsymbol{\epsilon}$ in observation Y. In fact, we have the expectation of $D_{KL}(p_P, p_G)$ as follows:

$$
\begin{aligned}
\mathbb{E}_\epsilon\left[D_{KL}(p_P, p_G)\right] &= \frac{1}{2}\left(-1 + b - \log b + \frac{\mathbb{E}_\epsilon a^2}{\Sigma_2} b\right) \\
&= \frac{1}{2}\left(-1 + b - \log b + (var(c) + (\mathbb{E}_\epsilon a)^2) b / \Sigma_2\right) \\
&\leq \frac{1}{2}\left(-1 + b - \log b + \frac{\sigma_\epsilon^2 + \|\hat{Y}\|^2}{\sigma_\epsilon^2}\left(\frac{\sigma_\epsilon^2}{s_{\min} + \sigma_\epsilon^2}\right)^2 b\right).
\end{aligned}
\tag{19}
$$

Presume that the observation Y is normalized, as well as $\hat{Y}$; hence, $\|\hat{Y}\|^2$ can be estimated as $\mathcal{O}(1)$. The main difference is affected by the kernel $k(\cdot, \cdot)$ (or Λ) and the term σ_ϵ^2. More specifically, The GPCB can achieve a smaller variance than the PCE method in a point-wise manner because $b > 1$, and the differences between the predictions of the two methods is of the order of σ_ϵ^2. Furthermore, if σ_ϵ^2 is sufficient small, i.e., $\sigma_\epsilon^2 \ll s_{\min}$, we have $b \to 1$, thus $\mathbb{E}_\epsilon\left[D_{KL}(p_P, p_G)\right] \to 0$. If $\sigma_\epsilon^2 = 0$, i.e., we investigate the noise-free models, then $b = 1$ and $a = 0$, which enforces $\Sigma_1 = \Sigma_2$ and $\mu_1 = \mu_2$, respectively. This means that p_G and p_P are identical distributions, i.e., $D_{KL}\left(p_P, p_G\right) = 0$.

We can conclude that the expected value of $D_{KL}(p_P, p_G)$ is bounded by a certain constant, which mainly depends on the σ_ϵ^2 and Λ. In other words, since σ_ϵ^2 is given in the prior, and the Λ are optimized; hence, the $D_{KL}(p_P, p_G)$ is constrained, which means that the GPCB is as stable as the PCE method. Nonetheless, if f_P has reached a desired prediction precision, then f_G with the kernel constructed with the same basis can have a desirable precision, as well as smaller variance.

Secondly, we consider that $\bar{M} > M$, where the $\boldsymbol{\beta}$ of the PCE method is not an unbiased estimate of $\bar{\boldsymbol{\beta}}$ any longer. Let $p_{\bar{P}}$ denote the PCE approximation with the $\bar{M}$ basis; under the circumstance, $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{k=0}^{\bar{M}} \lambda_l \boldsymbol{\phi}_l(\boldsymbol{x}) \boldsymbol{\phi}_l(\boldsymbol{x}')$ is achieved by tuning the value of Λ via a certain learning method; hence,

$D_{KL}(p_{\bar{P}}, p_G)$ is also bounded by a constant according to Equation (19), i.e., the GPCB can converge to the precise PCE prediction $p_{\bar{P}}$, as well. However, the KL divergence $D_{KL}(p_{\bar{P}}, p_P)$ is given as:

$$\begin{aligned} D_{KL}\left(p_{\bar{P}}, p_P\right) &= \frac{1}{2}\left(-1 + \frac{\bar{\Sigma}}{\Sigma_2} - \log\frac{\bar{\Sigma}}{\Sigma_2} + \frac{(\bar{\mu}-\mu_2)^2}{\Sigma_1}\right) \triangleq \frac{1}{2}\left(-1 + \bar{b} - \log\bar{b} + \frac{\bar{a}^2}{\Sigma_1}\right), \\ \bar{b} &= \frac{\boldsymbol{\phi}\Phi^T W^2 \Phi \boldsymbol{\phi}^T + \boldsymbol{\phi}_r \Phi_r^T W^2 \Phi_r \boldsymbol{\phi}_r^T + \boldsymbol{\phi} A \boldsymbol{\phi}_r^T}{\boldsymbol{\phi}\Phi^T W^2 \Phi \boldsymbol{\phi}^T} \geq 1, \\ \bar{a} &= \boldsymbol{\phi}_r \Phi_r^T W Y \sim \mathcal{N}\left(\boldsymbol{\phi}_r \Phi_r^T W \hat{Y}, \sigma_\epsilon^2 \boldsymbol{\phi}_r \Phi_r^T W^2 \Phi_r \boldsymbol{\phi}_r^T\right). \end{aligned} \tag{20}$$

We denote $\boldsymbol{\phi}_r$ as the basis that belongs to the model $f_{\bar{P}}$, but f_P. It is shown that the biased PCE f_P has smaller variance, however with a bias whose mean value depends on $\boldsymbol{\phi}_r$. We notice that $\boldsymbol{\phi}_r$ represents the high-order polynomials; hence, the bias can be considerably large for general cases, and so is the $D_{KL}\left(p_{\bar{P}}, p_P\right)$. We can conclude that even though the biased PCE has smaller variance, the relatively large bias can lead to a false prediction.

In fact, if the underlying system is smooth enough to be modeled by a polynomial approximation, then we can adaptively increase the number of polynomial bases (and the experimental designs if necessary) to reach a precise approximation. However, on the other hand, we can directly use the GPCB method, which is a one-step Bayesian approximation, that converges to the hypothetical true system $f_{\bar{P}}$. Roughly speaking, the GPCB finds $\bar{M}$ automatically by tuning the parameters Λ instead of adaptively changing the value of P in the PCE method. It indeed provides more convenience for computation. The key problem is the evaluation of Λ. Specifically, we introduce the Mehler kernel [37], which is an analytic expression of the Mercer kernel constructed by Hermite polynomials, and discuss the learning procedure of Λ_l.

4.2. Construction of the Kernel with Hermite Polynomials

Recall that we have $\{\boldsymbol{\phi}_\alpha\}$ in PCE as an orthonormal basis, then we regard them as eigenfunctions of a kernel $k(\cdot,\cdot)$. Since $p(\boldsymbol{x})$ follows the standard Gaussian distribution, then $\phi_l^{(i)}(x_i) = He_l(x_i)/\sqrt{l!}$, where x_i is the i-th variable of $\boldsymbol{x}$ and $He_l(x_i)$ is the Hermite polynomial of degree l:

$$He_l(x_i) = (-1)^l e^{\frac{x_i^2}{2}} \frac{d^l}{dx_i^l} e^{-\frac{x_i^2}{2}}. \tag{21}$$

Here, we denote the real l-th multi-index of the l-th polynomial in Equation (2) as $\boldsymbol{\alpha}^{(l)} = (\alpha_1^{(l)}, \ldots, \alpha_d^{(l)})$, $|\boldsymbol{\alpha}^{(l)}| = \alpha_1^{(l)} + \ldots + \alpha_d^{(l)}$ and $\boldsymbol{\alpha}^{(l)}! = \alpha_1^{(l)}! \ldots \alpha_d^{(l)}!$, then according to the previous analysis, we can get $\boldsymbol{\phi}_{\boldsymbol{\alpha}^{(l)}}(\boldsymbol{x}) = \boldsymbol{He}_{\boldsymbol{\alpha}^{(l)}}(\boldsymbol{x})/\sqrt{\boldsymbol{\alpha}^{(l)}!}$. We have Mehler kernel $Me(\boldsymbol{X}, \boldsymbol{X}')$ [37] with orthonormal Hermite polynomials as eigenfunctions:

$$Me(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\frac{D_{\boldsymbol{x}} D_{\boldsymbol{x}}^T + D_{\boldsymbol{x}'} D_{\boldsymbol{x}'}^T}{2}\right\}\left[\exp\sum_{i=1}^{d} \rho_i x_i x_i'\right] = \sum_{l=0}^{\infty} \boldsymbol{\rho}^{\boldsymbol{\alpha}^{(l)}} \boldsymbol{\phi}_{\boldsymbol{\alpha}^{(l)}}(\boldsymbol{x}) \boldsymbol{\phi}_{\boldsymbol{\alpha}^{(l)}}(\boldsymbol{x}'), \tag{22}$$

where the eigenvalue is $\lambda_{\boldsymbol{\alpha}^{(l)}} = \boldsymbol{\rho}^{\boldsymbol{\alpha}^{(l)}} = \prod_{i=1}^{d} \rho_i^{\alpha_i^{(l)}}$ for parameter $\boldsymbol{\rho}$, D_x is the symbol representing the row gradient operator, i.e., $D_{\boldsymbol{x}} = (\partial/\partial x_1, \ldots, \partial/\partial x_d)$, and $D_{\boldsymbol{x}'}$ is defined similarly. Specifically, in the one-dimensional case:

$$Me(x, x') = \frac{1}{\sqrt{1-\rho^2}} \exp\left(-\frac{\rho^2(x^2 + x'^2) - 2\rho x x'}{2(1-\rho^2)}\right) = \sum_{l=0}^{\infty} \rho^l \phi_l(x) \phi_l(x'), \tag{23}$$

where the eigenvalue $\lambda_l = \rho^l > 0$. The truncated kernel $Me_M(x, x') = \sum_{l=0}^{M} \rho^l \phi_l(x) \phi_l(x')$, and its attribution can be investigated by varying ρ and M. Figure 2a illustrates the truncated kernel $Me_M(x, x')$, which shows that $Me_M(x, x')$ tends to converge to $Me(x, x')$ as M grows. Figure 2b shows the values of $Me(x, -0.8)$ with different ρ. It presents to us that the influence of eigenvalue λ_l is greater on the Mehler kernel.

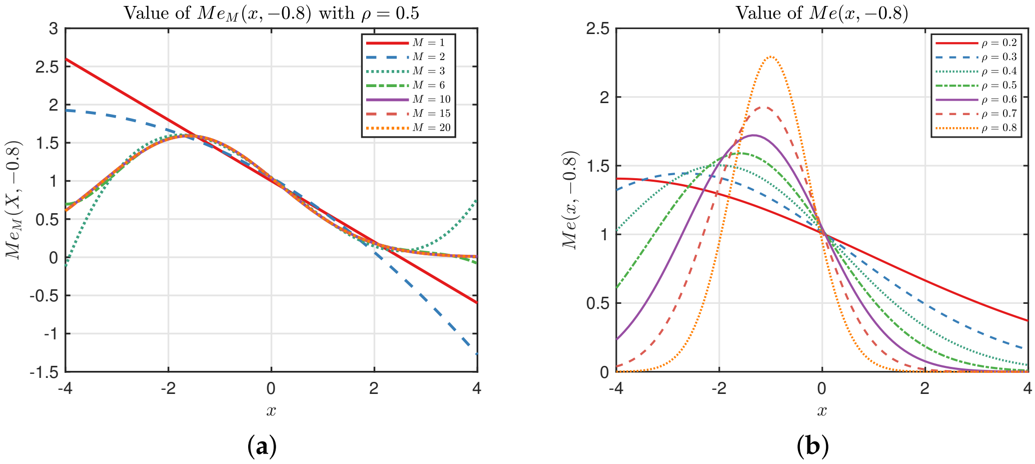


Figure 2. Comparison of the effect of M and ρ on the 1D Mehler kernel with one fixed point $\{-0.8\}$. (**a**) Kernel value of $Me_M(x, -0.8)$ with $\rho = 0.5$; (**b**) kernel value of $Me(x, -0.8)$.

4.3. Learning the Hyper-Parameter ρ of $Me(x, x')$

It is clear that $\lambda_l = \rho^l$ has a great impact on the kernel values, hence affecting the convergence of the $f_G(\boldsymbol{x})$. We start with a simple example, where $f(x) = 5 + x + \exp(x), x \sim \mathcal{N}(0, 2^2)$ is the true underlying function, and the noise term is ignored in the observation. Considering the Taylor expansion of $\exp(x)$, $f(x)$ can be approximated by the PCE with sufficiently large M. In fact, we can calculate the projection $< f(x), \phi_l(x) >$ to seek the value of M. When $l > M$, $< f(x), \phi_l(x) > \approx 0$. On the other hand, as discussed in Section 4.1, the GPCB can find M automatically by tuning the hyper-parameter ρ. Let the experimental design $\boldsymbol{X}$ be the zeros of $He_{10}(x)$ of Equation (21), i.e., quadrature points corresponding to degree 10; we compare the performance of the GPCB with ρ equal to $0.1, 0.45, 0.7$, respectively. The results are displayed in Figure 3. Note that the projection value is the absolute value of the true value in the figure for better illustration. It shows that we are able to approximate $f(x)$ with polynomials up to degree 40. If $\rho = 0.45$ for the Mehler kernel, the GPCB almost converges to exact $f(x)$, whereas $\rho = 0.1$ leads to a fast convergence rate and $\rho = 0.7$ results in a slow convergence rate.

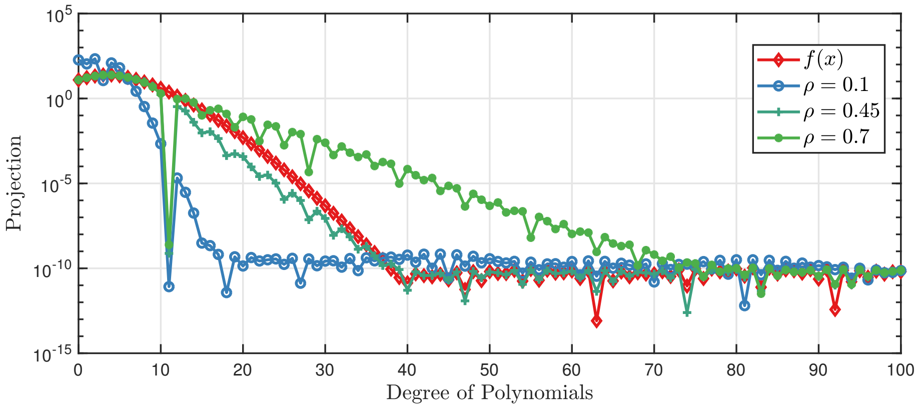


Figure 3. Projections on the first 100 polynomials of $f(x)$ and the Gaussian process on polynomial chaos basis (GPCB) with $\rho = 0.1, 0.45, 0.7$.

Figure 3 shows that the ρ has a crucial impact on the performance of the GPCB method, so a tractable method to optimize ρ is needed. A natural criterion is the KL divergence, which can be minimized by finding the optimal hyper-parameters ρ. We discussed the KL divergence of the GPCB and the PCE surrogates under the assumption that the PCE surrogate model can approximate the true system to any degree of accuracy. However, the distribution of the real system is usually unknown, which makes the calculation of KL divergence intractable. In fact, it can be easily deduced that the minimization of KL divergence is equivalent to minimizing the negative log marginal likelihood Δ, which (actually is 2Δ) reads:

$$\Delta = Y^T K_Y^{-1} Y + \log \left[(2\pi)^N |K_Y| \right] \tag{24}$$

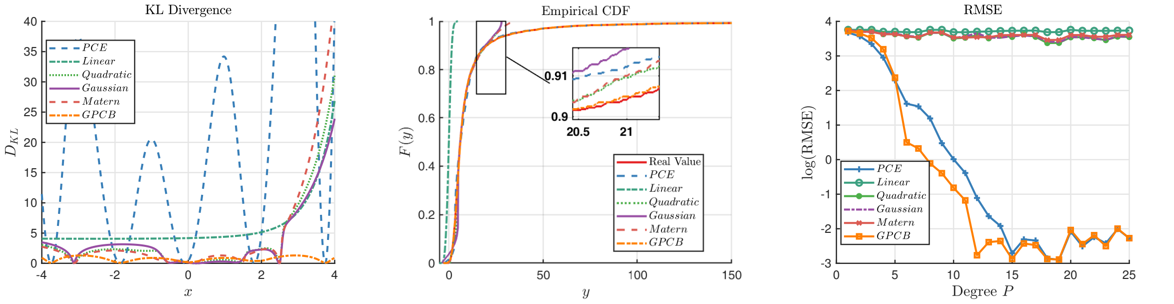
It is important to optimize ρ and σ_ϵ^2 to obtain a suitable kernel to get an accurate approximation. Classical methods like gradient-based techniques can be used to search for the optimal ρ; however, it may perform poorly because it is locally optimized. As we can see in Equation (23), it is indicated that ρ should take a value between zero and one, so we can propose a global method to solve our optimization problem. The algorithm for generating a GPCB approximation is given in Algorithm 1:

Algorithm 1: General procedure of the GPCB method with the Mehler kernel.

```
Input: Simulator $y(x)$, prior distribution $p(x)$
Output: A GPCB approximation $f_G(x)$
Data: $\Delta_0 = \infty, \Delta_1 = 0, \rho_0 = 0.001, \rho_1 = 0.999, eps = 1e-6$
Initialize polynomial basis $\boldsymbol{\phi}_l(\boldsymbol{x})$ w.r.t $p(\boldsymbol{x})$;
Construct corresponding Mehler kernel according to Equation (22);
Sample an experimental design $\{\boldsymbol{X}, Y\}$ with $Y = y(\boldsymbol{X})$;
while $|\Delta_0 - \Delta_1| \geq eps$ do
    Divide $[\rho_0, \rho_1]$ into 10 intervals to get $\rho^{(0)} = \min\{\rho_0, \rho_1\}, \rho^{(1)}, \ldots, \rho^{(10)} = \max\{\rho_0, \rho_1\}$;
    Calculate $\Delta$ according to Equation (24) w.r.t $\rho^{(i)}$ to get $\Delta^{(i)}$;
    Select the first two minimal $\Delta^{(i)}$ to get $\Delta^{(i_1)}, \Delta^{(i_2)}$;
    Assign $\Delta_0 = \Delta^{(i_1)}, \Delta_1 = \Delta^{(i_2)}$;
    Assign $\rho_0 = \rho^{(i_1)}, \rho_1 = \rho^{(i_2)}$;
end
Select $\rho$ with minimal $\Delta$, together with Equation (22) and Equation (5), to obtain $f_G(x)$.
```

5. Numerical Investigation

In this section, we investigate the GPCB method for various benchmark functions. Firstly, we investigate the same example in Figure 3; however, the noise term is considered, i.e., $y = f(x) + \epsilon = 5 + x + \exp(x) + \epsilon$ is the observation, where $x \sim \mathcal{N}(0, 2^2), \epsilon \sim \mathcal{N}(0, 0.1^2)$. Three methods, i.e., GP with the RBF kernel, PCE and GPCB, are implemented. It is necessary to note that the Monte Carlo (MC) sampling strategy is used in the normal GP approaches, and Gaussian quadrature points are introduced in the GPCB approach. The main reason is that the quadrature points are too sparse for the widely-used kernels to capture the local features. For example, in this case, $P = 10$ in PCE, then the maximum quadrature point is 10.376, which is beyond $3\sigma_x$. In the first set of experiments, let $P = 10$ in PCE, which means 11 sample points are used in the experiments; furthermore, 10,000 samples are introduced as test dataset to output the ECDF (empirical cumulative distribution function) and RMSE (root-mean-squared error). The GP algorithm is implemented by the gpml toolbox [38] written in MATLAB with four different kernels, i.e., linear, quadratic, Gaussian and Matérn-3/2 kernels. The comparisons of the results are displayed in Figure 4.

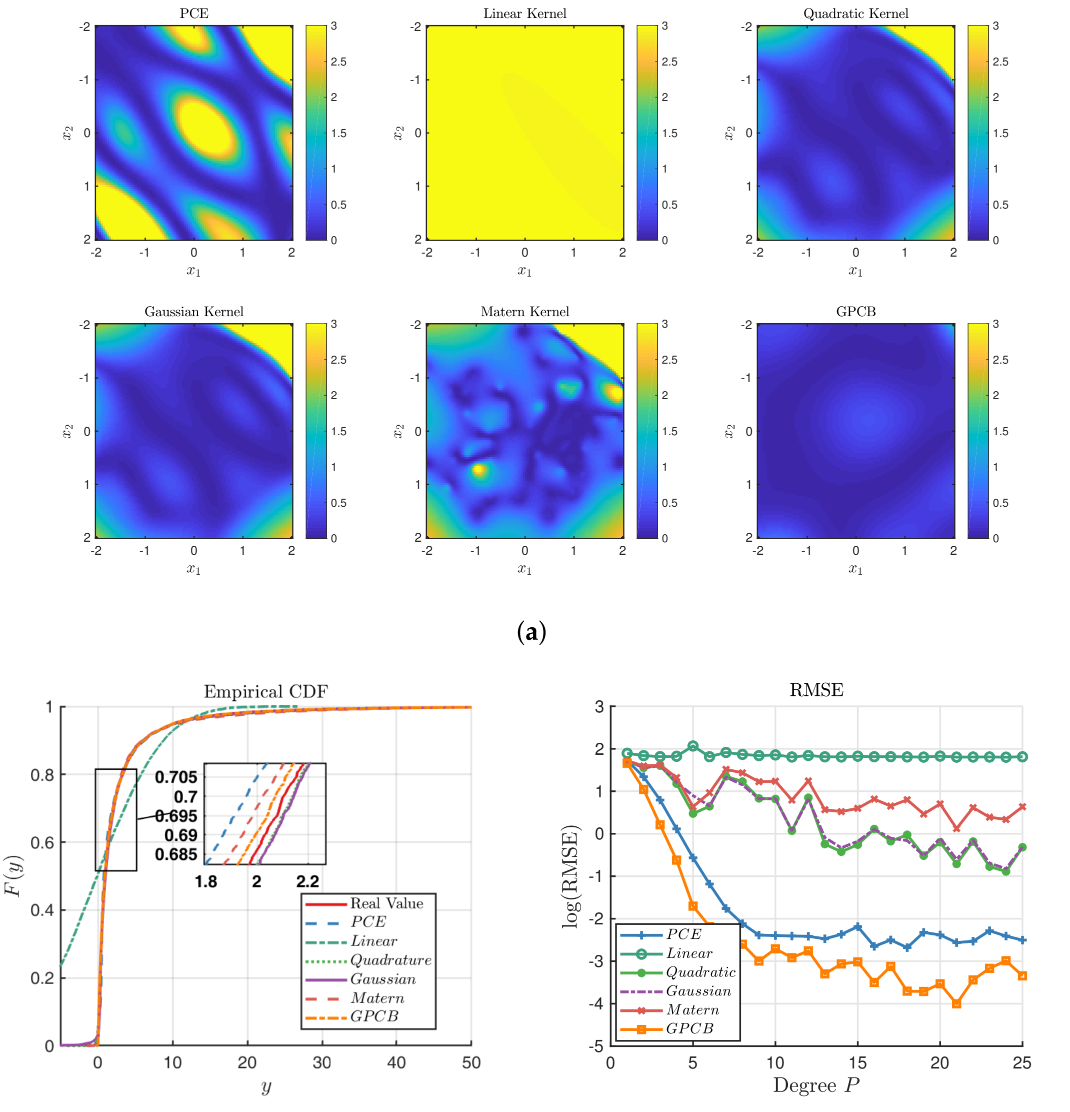


(**a**) (**b**) (**c**)

Figure 4. Comparisons among the GP, polynomial chaos expansion (PCE) and GPCB surrogates for the 1D example. (**a**) Comparison of the KL divergence, $P = 10$; (**b**) comparison of the ECDF of prediction, $P = 10$; (**c**) comparison of the RMSE with different degrees P in the 1D example.

Figure 4a illustrates the point-wise KL divergence between the true value of $f(\boldsymbol{x})$ and the predictions on the interval $[-4, 4]$ based on Equation (16). It is clear that the distribution of GPCB prediction is statistically closest to the true response, although the GP methods with quadratic, Gaussian and Matérn-3/2 kernels outperform the GPCB at some points. Figure 4b compares the ECDF of y based on the test dataset. It shows that both the PCE and GPCB have a similar ECDF with the true value. Upon closer inspection, which is shown in the magnified subregion, it is obvious that the ECDF of the GPCB is almost exactly the same as the real ECDF, which shows that GPCB has actually captured the feature of $f(x)$ with high precision. At the same time, the RMSEs of the PCE, linear kernel, quadratic kernel, Gaussian kernel, Matérn-3/2 kernel and GPCB are $1.0570, 37.3526, 23.6383, 25.6044, 27.4498, 0.4108$, respectively. We have implemented another experiment, which uses the degree P (i.e., the number of experimental design) as the second set of experiments, which are illustrated in Figure 4c. Figure 4c shows that GPCB generally outperforms the ordinary GP with the RBF kernel, which indicates that GPCB performs better with a few (or sparse) training points. It is notable that the PCE and GPCB perform with almost the same precision when the degree is greater than 16. It echoes the idea that PCE and GPCB are statistical equivalent, as we present in Section 4.1.

Similar experiments are conducted with a two-dimensional function, which is expressed as $f(\boldsymbol{x}) = \exp(x_1)/\exp(x_2)$. Let $x_1, x_2 \sim \mathcal{N}(0, 1)$, $y = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$ be the real model where $\boldsymbol{\epsilon}$ is an independent noise term with a normal distribution $N(0, 0.1^2 I_2)$. Unlike the first test function, this test example is a limit state function. Let the maximum degree for each dimension p_t be seven for the PCE method, which makes 64 training points in total. Another dataset of 10,000 independent samples is introduced as the test set to calculate the ECDF and RMSE, as well. Similarly, we have the point-wise KL divergence in the region $[-2, 2; -2, 2]$ as shown in Figure 5a. It is clear that the GPCB is globally closer to the true distribution than other methods. The GP with quadratic, Gaussian and Matérn-3/2 kernels can approximate the center part well, while the PCE does not seem to perform as well. Figure 5b shows that the six methods except the linear kernel are able to reconstruct the distribution of the prediction, and upon closer observation, we find that the ECDF of the GPCB and Matérn-3/2 kernel are the best approximations among the six methods. We also consider another set of experiments focusing on the number of experimental designs, which equals $(p_t + 1)^2$ for the 2D function. The RMSEs of the three methods with respect to different p_t are displayed in Figure 5c. It also shows that the PCE and GPCB generally outperform the normal GP approaches, and the GPCB has the best performance.



(**a**)

(**b**)

(**c**)

Figure 5. Comparisons among the GP, PCE and GPCB surrogates for the 2D example. (**a**) Comparison of the KL divergence with $p_t = 7$; (**b**) comparison of the ECDF of prediction, $p_t = 7$; (**c**) comparison of the RMSE with different degrees p_t in the 2D example.

To summarize, the GPCB generates a surrogate of infinite series, while the PCE can only generate a surrogate with up to $P + 1$ polynomials, and they tend to behave with similar precision when P is large enough. A set of sparse quadrature points sampled in the PCE, which are derived from the Gaussian quadrature rule, is a good design for the GPCB. The GPCB with those training points generally performs better than the normal GP methods and PCE. However, the size of such a training set grows dramatically with the dimension ($N = (p_t + 1)^d$ in total), so it is not practical in real-life applications. We aim to present a strategy of sampling from those quadrature points, namely candidate points in the next section, and analyze the performance of our algorithm on the selected points.

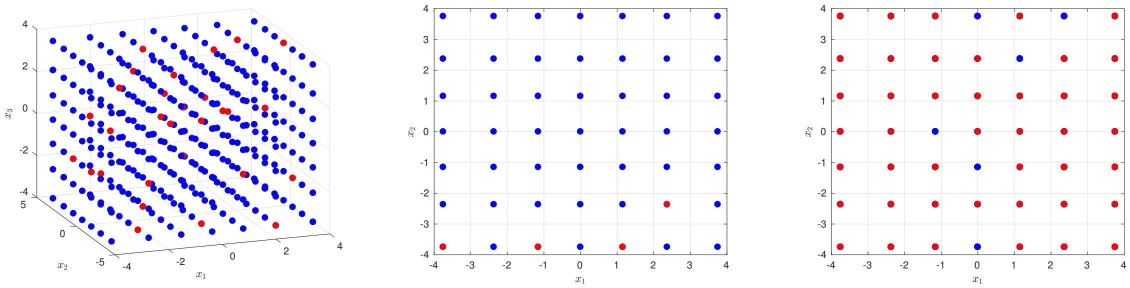
5.1. A Random Constructive Design in High Dimensional Problems

As the dimension of a system grows, so do the number of design points of PCE due to a tensor product of quadrature points in each dimension. It is possible that PCE could deal with thousands of points of training data with acceptable computational time; however, it becomes expensive for

GP approaches, including our GPCB approach. Monte Carlo sampling techniques can substitute quadrature design; however, these are not always stable. Other sampling strategies like Halton sampling and Latin hypercube sampling [39] are widely used.

In this work, we want to utilize the high accuracy of quadrature points and also want to reduce the massive number of points. Let $\boldsymbol{x} \in \mathbb{R}^d$, and p_t is the maximum degree in each dimension, so $p_t + 1$ quadrature points are needed in each dimension, which makes the total number of tensor products of quadrature points be $\#\{\boldsymbol{X}_c\} = (p_t + 1)^d$. We seek to find a subset of the candidate design $\boldsymbol{X}_c$. Furthermore, we wish to obtain a subset having a good coverage rate in the space. Therefore, we proposed the random definite design in our paper. Note that the LHS design can be extended to a larger interval $(1, (p_t + 1)^D)$ and can produce points at midpoints (endpoints), so we use the LHS design to sample N indices from the interval. More specifically, we presume that the points in $\boldsymbol{X}_c$ are equally important, so we arrange those points with a certain order to get their indices. Then, we sample from the indices with the LHS design, and each index is related to a certain quadrature point. It can be easily implemented by the MATLAB built-in function lhsdesign. The corresponding N points are what we need.

Take a three-dimensional input space as an example, where $x_i \sim \mathcal{N}(0, 1), i = 1, 2, 3$. Set $p_t = 6$, then $\#\{\boldsymbol{X}_c\} = 343$. The candidate design and its subset of 50 points $\boldsymbol{X}$ are illustrated in Figure 6. We can see from the figure that our sampling is sparse in the whole set of candidate points, and it behaves uniformly in dimension one as illustrated in Figure 6b, with similar conclusions in the other two dimensions. When projecting our sampling from dimension three to get Figure 6c, we can see that the selected points almost cover every point of $\boldsymbol{X}_c$, which means it has all features in dimensions one and two, i.e., quadrature point values of the two dimensions. It shows that such a method can generate a sparse subset meanwhile guaranteeing the coverage rate in the whole candidate design. We name it the random constructive design.



(**a**) (**b**) (**c**)

Figure 6. Left: $\boldsymbol{X}_c$ and $\boldsymbol{X}$ in 3D view; the blue dots represent the quadrature points, while the red points represent our samplings; middle: this shows the sparsity of our sampling in $\boldsymbol{X}_c$; right: this shows that our sampling actually covered almost every feature of $\boldsymbol{X}_c$. (**a**) $\boldsymbol{X}_c$ and $\boldsymbol{X}$ in 3D view; (**b**) one slice of $\boldsymbol{X}_c$; (**c**) projection of $\boldsymbol{X}$ on $\boldsymbol{X}_c$ in dimensions one and two.

Now, we want to find out whether these samples retain their capability of accuracy. Firstly, we use the PCE method to test those samples. We will look into the benchmark Ishigami function [40]: $f(\boldsymbol{x}) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4\sin(x_1)$, where four different sampling strategies are compared here. Set $p_t = 15$, and the candidate design $\boldsymbol{X}_c$ has a size of 4096. The error term ϵ is eliminated in this simulation for the accuracy test. The RMSE are computed on 10,000 independently-sampled data, and the results are presented below in Figure 7. It can be seen that our sample always performs better than other samples. When the number of samples surpasses 900, the RMSE becomes 1.0605×10^{-5}, which equals the RMSE with the whole candidate points. Therefore, we only select 20% of $\boldsymbol{X}_c$ and get the same precision. Furthermore, if we set our precision to be 10^{-2}, only 400 points are needed.

This shows that the quadrature points have high precision in numerical calculation. In other words, the points in the candidate set are good points.

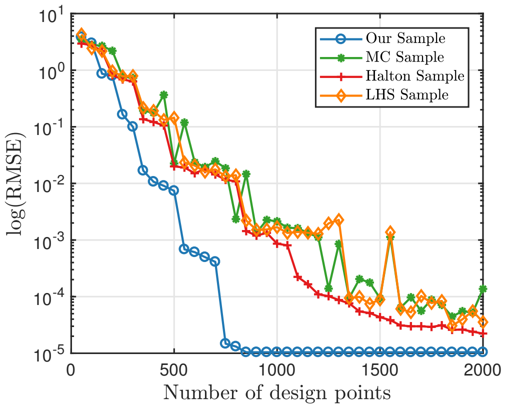


Figure 7. Comparison of the RMSE between four sampling strategies with the PCE method.

Then, the random constructive design is used with the PCE, GP and GPCB methods for the Ishigami function, with the noise term added in the observations. We take the RMSE as a criterion to compare their performance, and the results are illustrated in Figure 8. Figure 8a shows that the GPCB is always better than the PCE method, and they tend to behave the same. However, as the number of sampling points grows, the GP with the quadratic kernel, Gaussian kernel and Matérn-3/2 kernel generally outperform the other methods. We can see that the Ishigami function is a bounded function; therefore, it is likely to fill the whole observation space as the number of samples increases, hence improving the accuracy of the GP method. We plot the ECDF with respect to the three methods when $N = 1000$ in Figure 8b. It is clear that the GP with the quadratic kernel, Gaussian kernel and Matérn-3/2 kernel can almost recover the true distribution of the response, which is beyond the capability of the PCE and GPCB.

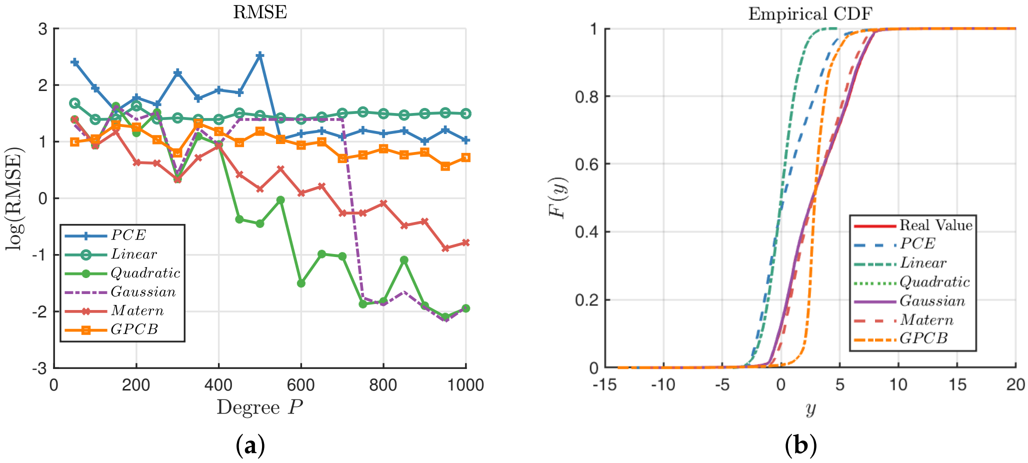


Figure 8. Comparisons among the GP, PCE and GPCB surrogates for the Ishigami function. (**a**) Comparison of the RMSE; (**b**) comparison of the ECDF; $N = 1000$.

A six-dimensional problem is being tested with the G-function [41], which is not like the Ishigami function and is unbounded in the domain $[-\infty, \infty]^6$:

$$f(\boldsymbol{x}) = \prod_{i=1}^{6} \frac{|4x_i - 2| + a_i}{1 + a_i}, \quad \textit{where} \quad a_i = \frac{i-2}{2}, \quad \forall i = 1, \ldots, 6 \tag{25}$$

The experiment is performed with the same approaches, and the results are shown below in Figure 9. Figure 9a shows that the GPCB outperforms the PCE and GP with the Gaussian Kernel, and it has similar precision with the quadratic and Matérn-3/2 kernels. We notice that the GPCB is more stable than the PCE method, which behaves badly especially when $N = 150, 300$. Figure 9b shows that none of these three methods can reconstruct the probability of y very well; however, we can note that the GPCB is still comparatively the closest.

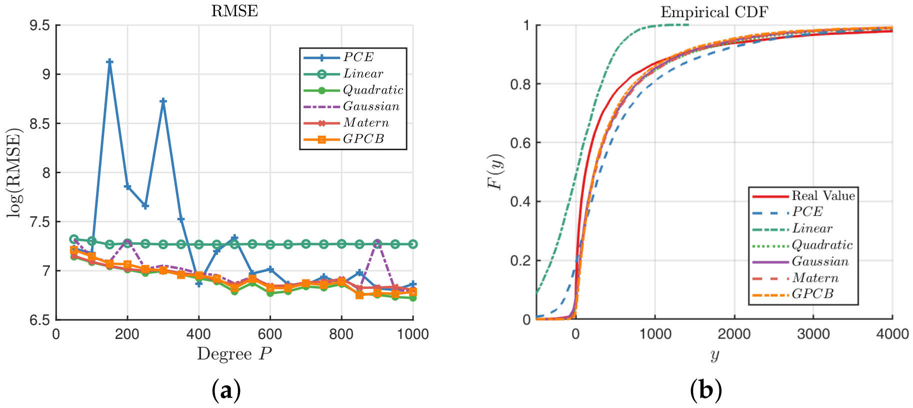


Figure 9. Comparisons among the GP, PCE and GPCB surrogates for the G-function. (**a**) Comparison of the RMSE; (**b**) comparison of the ECDF; $N = 1000$.

Finally, we are going to present a more complicated model with 15 dimensional functions with the following form:

$$f(\boldsymbol{x}) = \boldsymbol{a}_1^T \boldsymbol{x} + \boldsymbol{a}_2^T \sin(\boldsymbol{x}) + \boldsymbol{a}_3^T \cos(\boldsymbol{x}) + \boldsymbol{x}^T M \boldsymbol{x}. \tag{26}$$

The distribution of $\boldsymbol{x}$ is the product of 15 independent distributions, i.e., $x_i \sim \mathcal{N}(0,1), i = 1, \ldots, 15$. This function is introduced by the work of O'Hagan, where $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, M$ are defined in [42]. We can see that this function is dominated by the linear and quadratic term, so it may be well approximated by the low-order PCE model. Let $p_t = 3$ in the PCE model; we can see from Figure 10a that the GP with the quadratic kernel performs best among the six methods, while the PCE performs better than the GPCB and other GP methods. On the other hand, the GPCB is always generally better than the GP method except with the quadratic kernel for this function. When $N = 1000$, the PCE can generate y, which follows the real distribution according to the ECDF in Figure 10b.

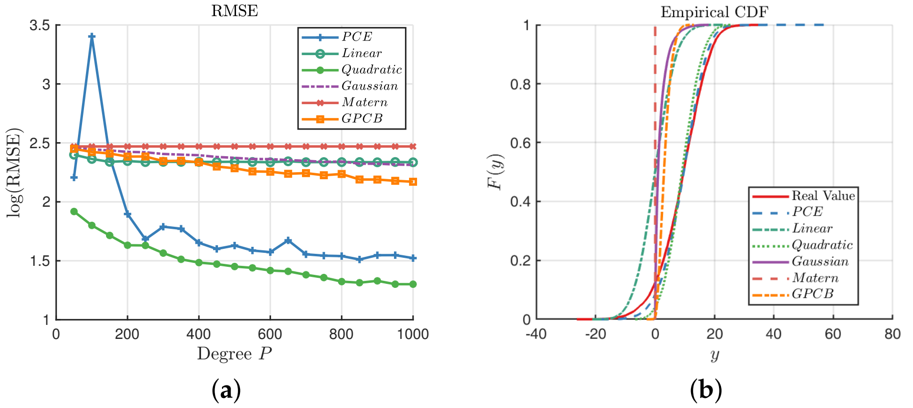


Figure 10. Comparisons among the GP, PCE and GPCB surrogates for Equation (26). (**a**) Comparison of the RMSE; (**b**) comparison of the ECDF; $N = 1000$.

6. Conclusions

This paper has examined two different surrogates of computational models, i.e., polynomial chaos expansion and Gaussian process regression. First, we present a brief review of these two approaches. Next, we discuss the relationship between PCE and GP and find that PCE and GP surrogates are embedded in two isomorphic RKHS. Mercer's theorem is introduced to generate a kernel based on a PCE basis, by which a new approach is proposed, which we name GPCB. An example shows that with the same experimental design, GPCB tends to retain useful information in a suitable subspace of the RKHS by changing the hyper-parameters, whereas PCE simply sets the information of the residual to zero. We further investigate the approximation performance on two test functions in 1D and 3D, respectively, and their approximation properties are illustrated. In order to deal with the high dimensional scenario, a random constructive design from the quadrature points is used to generate an experimental design. The results give us several directions for choosing models: basically, the GPCB outperforms the PCE, but when the original model can be well approximated by low-order PCE (Figure 10), it seems cumbersome to introduce the GPCB and GP; when the response function is bounded (Figure 8), if we have enough training resources, the GP can be a better choice; when the objective function is unbounded (Figures 4 and 9) or cannot be approximated by finite polynomials (Figure 5), we should probably choose the GPCB.

Future work can extend the family of the Mercer kernel or equivalent kernel (other than the Mehler kernel presented in this paper) beyond a classical approximation method. We can also analyze the experimental design for GP regression in many ways. Although we can see that our sampling method behaves fair enough in the experiments, there is also the opportunity to discover further suitable experimental design schemes to fit different computational purposes, which would be of great interest. The stability of our method will be investigated in future work, i.e., how many points are needed to train a good surrogate and whether our method always produces a suitable design. Furthermore, we can establish closer connections between numerical analysis and statistics via such combinations.

Acknowledgments: This work is supported by the program for New Century Excellent Talents in University, State Education Ministry in China (No. NCET 10-0893) and the National Science Foundation of China (No. 11771450, No. 61573367).

Author Contributions: Liang Yan proposed the original idea, implemented the experiments in the work and wrote the paper. Xiaojun Duan contributed to the theoretical analysis and simulation designs. Bowen Liu partially undertook the writing and simulation work. All authors read and approved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proofs of Proposition 1, Proposition 3, Proposition 2 and Theorem 1

Here, we use the same notations as in Section 3.

Appendix A.1. Proof of Proposition 1

Proof. Define the inner product in the above subspace $\mathcal{H}_P$ as follows:

$$< f(\boldsymbol{x}), g(\boldsymbol{x}) >_{\mathcal{H}_P} = \sum_l f_l g_l / \lambda_l. \tag{A1}$$

Firstly, it is obvious that $\mathcal{H}_P$ is a Hilbert space in $\mathcal{H}$. Secondly, for any $\boldsymbol{x} \in \Omega$, $k(\boldsymbol{x}, \cdot)$ belongs to $\mathcal{H}_P$ because:

$$< k(\boldsymbol{x}, \cdot), k(\boldsymbol{x}, \cdot) >_{\mathcal{H}_P} = \sum_l \frac{< k(\boldsymbol{x}, \cdot), \boldsymbol{\phi}_l(\cdot) >^2}{\lambda_l} = \sum_i \lambda_l \boldsymbol{\phi}_l^2(\boldsymbol{x}) < \infty. \tag{A2}$$

It also has the reproducing property for:

$$< f(\cdot), k(\boldsymbol{x},\cdot) >_{\mathcal{H}_P} = \sum_l f_l \lambda_l \boldsymbol{\phi}_l(\boldsymbol{x})/\lambda_l = f(\boldsymbol{x}) \qquad for \quad \forall\, \boldsymbol{x} \in \Omega, \tag{A3}$$

We have the conclusion that $\mathcal{H}_P$ is an RKHS derived from the Mercer kernel. □

Appendix A.2. Proof of Proposition 2

Proof. In fact, $\mathcal{H}'_G$ is a space of all finite linear combinations of functions $k(\boldsymbol{x},\cdot) : \Omega \to \mathbb{R}$, so we can denote $\mathcal{H}'_G := span\{k(\boldsymbol{x},\cdot)|\boldsymbol{x} \in \Omega\}$, and the elements in $\mathcal{H}'_G$ have the general form of $f(\boldsymbol{x}) = \sum_{i=1}^N f_i k(\boldsymbol{x}, X_i)$. Therefore, different N and all experimental designs $\boldsymbol{X}$ are allowed, which enables that $f(\boldsymbol{x}) = \sum_{i=1}^N f_i k(\boldsymbol{x}, X_i), g(\boldsymbol{x}) = \sum_{i=1}^{N'} f_i k(\boldsymbol{x}, X'_i) \in \mathcal{H}'_G$. The linearity of $\mathcal{H}'_G$ is given by the following explanation.

Let $t_1, t_2 \in \mathbb{R}$ be scalars, then we can rewrite $t_1 f^{(1)}(\boldsymbol{x}) + t_2 f^{(2)}(\boldsymbol{x})$ as a function $f(\boldsymbol{x})$ such that:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_1} t_1 f_i^{(1)} k(\boldsymbol{x}, X_i^{(1)}) + \sum_{j=1}^{N_2} t_2 f_j^{(2)} k(\boldsymbol{x}, X_j^{(2)}) \triangleq \sum_{l=1}^{N} f_l k(\boldsymbol{x}, X_l), \tag{A4}$$

where $N = N_1 + N_2$, $\{f_l, l = 1, \ldots, N\} = \{t_1 f_i^{(1)}, i = 1, \ldots, N_1\} \cup \{t_2 f_j^{(2)}, j = 1, \ldots, N_2\}$, $\boldsymbol{X} = \boldsymbol{X}^{(1)} \cup \boldsymbol{X}^{(2)}$. Additionally, we have:

$$\begin{aligned}\sum_{l=1}^{N}\sum_{m=1}^{N} f_l f_m k(X_l, X_m) =& \sum_{i=1}^{N_1}\sum_{i'=1}^{N_1} t_1^2 f_i^{(1)} f_{i'}^{(1)} k(X_i^{(1)}, X_{i'}^{(1)}) \\ &+ \sum_{j=1}^{N_2}\sum_{j'=1}^{N_2} t_2^2 f_j^{(2)} f_{j'}^{(2)} k(X_j^{(2)}, X_{j'}^{(2)}) + 2\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} t_1 t_2 f_i^{(1)} f_j^{(2)} k(X_i^{(1)}, X_j^{(2)}) \\ &< +\infty,\end{aligned} \tag{A5}$$

which means that $t_1 f^{(1)}(\boldsymbol{x}) + t_2 f^{(2)}(\boldsymbol{x})$ also belongs to $\mathcal{H}'_G$. Then, we would like to show that $\mathcal{H}'_G$ is an inner product space. Let the kernel function be positive semi-definite, and we define the inner product of $\mathcal{H}'_G$ as follows:

$$< f(\boldsymbol{x}), g(\boldsymbol{x}) >_{\mathcal{H}'_G} = \sum_{i=1}^{N}\sum_{j=1}^{N'} f_i g_j k(X_i, X'_j). \tag{A6}$$

$< \cdot, \cdot >_{\mathcal{H}'_G}$ is a well-defined inner product by checking the following conditions:

1. Symmetry: $< f, g >_{\mathcal{H}'_G} = \sum_{i,j} f_i g_j k(X_i, X'_j) = \sum_{j,i} g_j f_i k(X'_j, X_i) =< g, f >_{\mathcal{H}'_G}$;
2. Bi-linearity:

$$\begin{aligned}< t_1 f^{(1)} + t_2 f^{(2)}, g >_{\mathcal{H}'_G} =& \sum_{l=1}^{N}\sum_{j=1}^{N'} f_l g_j k(X_l, X'_j) \\ =& a\sum_{i=1}^{N_1}\sum_{j=1}^{N'} f_i^{(1)} g_j k(X_i^{(1)}, X'_j) + b\sum_{i=1}^{N_2}\sum_{j=1}^{N'} f_i^{(2)} g_j k(X_i^{(2)}, X'_j) \\ =& a < f^{(1)}, g >_{\mathcal{H}'_G} + b < f^{(2)}, g >_{\mathcal{H}'_G};\end{aligned}$$

3. Positive-definiteness: It is obvious that $< f, f >_{\mathcal{H}'_G} = \boldsymbol{f}^T K \boldsymbol{f} \geq 0$ with the equality iff $f = 0$.

□

Appendix A.3. Proofs of Proposition 3

Proof. Firstly, we can prove that the reproducing formula holds for the space $\mathcal{H}'_G$. For any X, $k(X, \boldsymbol{x})$ is a function of $\boldsymbol{x}$ and belongs to $\mathcal{H}'_G$. Furthermore, we have:

$$< f(\cdot), k(\boldsymbol{x}, \cdot) >_{\mathcal{H}'_G} = \sum_{i=1}^{N} f_i k(\boldsymbol{x}, X_i) = f(\boldsymbol{x}). \tag{A7}$$

The above reproducing property is valid for any $f \in \mathcal{H}'_G$; thus, it is still valid for the closure $\mathcal{H}_G$ in the sense of generalizing the above equation as $< f(\cdot), k(\boldsymbol{x}, \cdot) >_{\mathcal{H}_G} = f(\boldsymbol{x})$. Then, we need to prove that $\mathcal{H}_G$ is unique. Suppose that we have another Hilbert space $\mathcal{H}_{G'}$, which is possibly an RKHS of the kernel $k(\cdot, \cdot)$, then, for a specific $\boldsymbol{X}$, we can get:

$$< k(\boldsymbol{x}, \boldsymbol{X}), k(\boldsymbol{x}', \boldsymbol{X}) >_{\mathcal{H}'_G} = k(\boldsymbol{x}, \boldsymbol{x}') =< k(\boldsymbol{x}, \boldsymbol{X}), k(\boldsymbol{x}', \boldsymbol{X}) >_{\mathcal{H}_{G'}} . \tag{A8}$$

This proves that the two inner products are the same on $\mathcal{H}'_G$; then, $\mathcal{H}_{G'}$ must contain $\mathcal{H}_G$ because it is the closure of $\mathcal{H}'_G$. $\mathcal{H}_{G'}$ must be equivalent to $\mathcal{H}_G$, otherwise we can find a nonzero element $f \in \mathcal{H}_{G'} - \mathcal{H}_G$ such that it is orthogonal to $\mathcal{H}_G$. However, we can always get that $f =< f, k(\cdot, \boldsymbol{X}) >_{\mathcal{H}_{G'}} \equiv 0$ for a particular $\boldsymbol{X}$, which is a contradiction. □

Appendix A.4. Proofs of Theorem 1

Proof. Define a weighted l^2 space such that:

$$l^2_\lambda = \left\{ h \,\middle|\, < h, h >_{l^2_\lambda} = \sum_l \lambda_l h_l^2 < \infty \right\}. \tag{A9}$$

It is clear that $\{c_l(\boldsymbol{X})\} \in l^2_{1/\lambda}$ for $c_l(\boldsymbol{X})$ defined in Equation (12). $l^2_{1/\lambda}$ is the completion of the span of all $\{c_i(\boldsymbol{X})\}$, then $\mathcal{H}_G$ is isometrically isomorphic to $l^2_{1/\lambda}$, because firstly, $c_l(\boldsymbol{X})$ is identically determined by $k(\cdot, \cdot)$ and $\boldsymbol{X}$, secondly, according to Equation (13), $< f_{\boldsymbol{X}}(\boldsymbol{x}), g_{\boldsymbol{X}'}(\boldsymbol{x}) >_{\mathcal{H}_G} = \sum_l c_l(\boldsymbol{X}) c_l(\boldsymbol{X}') / \lambda_l$. On the other hand, there exists a linear map such that:

$$T : l^2_{1/\lambda} \to \mathcal{H}_P, \quad T(c) = \sum_l c_l \boldsymbol{\phi}_l. \tag{A10}$$

This is a surjective map for every $f \in \mathcal{H}_P$, $\{f_i\} \in l^2_{1/\lambda}$. Now, we need to prove that the projection T is injective. Assume there exist c and c' such that $T(c) = T(c')$, then:

$$0 = \|T(c) - T(c')\|^2_{\mathcal{H}_P} =< T(c - c'), T(c - c') >_{\mathcal{H}_P} = \sum_l (c_i - c'_i)^2 / \lambda_l < \infty, \tag{A11}$$

which proves $c = c'$. Meanwhile,

$$< f(\boldsymbol{x}), g(\boldsymbol{x}) >_{\mathcal{H}_P} = \sum_l f_l g_l / \lambda_l, \quad \{f_l\}, \{g_l\} \in l^2_{1/\lambda}. \tag{A12}$$

The inner products remain equal, so it is also clear that $\mathcal{H}_P$ is isometrically isomorphic to $l^2_{1/\lambda}$.

To summarize, we can say that $\mathcal{H}_G$ and $\mathcal{H}_P$ are isomorphic by establishing a Hilbert space $l^2_{1/\lambda}$ to connect them; or it can be said that PCE and GP with the Mercer kernel generate surrogates in the same Hilbert space. □

References

1. Schwefel, H.P.P. *Evolution and Optimum Seeking: The Sixth Generation*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1993.
2. Santner, T.J.; Williams, B.J.; Notz, W.I. *The Design and Analysis of Computer Experiments*; Springer Science & Business Media: Berlin, Germany, 2013.
3. Hurtado, J.; Barbat, A.H. Monte Carlo techniques in computational stochastic mechanics. *Arch. Comput. Methods Eng.* **1998**, *5*, 3–29.
4. Conti, S.; O'Hagan, A. Bayesian emulation of complex multi-output and dynamic computer models. *J. Stat. Plan. Inference* **2010**, *140*, 640–651.
5. Higdon, D.; Gattiker, J.; Williams, B.; Rightley, M. Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **2008**, *103*, 570–583.
6. Balci, O. Verification, validation, and certification of modeling and simulation applications. In Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, New Orleans, LA, USA, 7–10 December 2003.
7. Rubino, G.; Tuffin, B. *Rare Event Simulation Using Monte Carlo Methods*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2009.
8. Sundar, V.; Shields, M.D. Surrogate-enhanced stochastic search algorithms to identify implicitly defined functions for reliability analysis. *Struct. Saf.* **2016**, *62*, 1–11.
9. Shan, S.; Wang, G.G. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct. Multidiscip. Optim.* **2010**, *41*, 219–241.
10. Fadale, T.D.; Nenarokomov, A.V.; Emery, A.F. Uncertainties in parameter estimation: The inverse problem. *Int. J. Heat Mass Transf.* **1995**, *38*, 511–518.
11. Liang, B.; Mahadevan, S. Error and uncertainty quantification and sensitivity analysis in mechanics computational models. *Int. J. Uncertain. Quantif.* **2011**, *1*, 147–161.
12. De Cursi, E.S.; Sampaio, R. *Uncertainty Quantification and Stochastic Modeling with Matlab*; Elsevier: Amsterdam, The Netherlands, 2015.
13. Friedman, J.H. Multivariate adaptive regression splines. *An. Stat.* **1991**, *19*, 1–67.
14. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.J.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *9*, 155–161.
15. Oparaji, U.; Sheu, R.J.; Bankhead, M.; Austin, J.; Patelli, E. Robust artificial neural network for reliability and sensitivity analyses of complex non-linear systems. *Neural Netw.* **2017**, *96*, 80–90.
16. Sun, Z.; Wang, J.; Li, R.; Tong, C. LIF: A new kriging based learning function and its application to structural reliability analysis. *Reliab. Eng. Syst. Saf.* **2017**, *157*, 152–165.
17. Ghanem, R.; Spanos, P.D. *Stochastic Finite Elements: A Spectral Approach*; Springer: Berlin, Germany, 1991.
18. Xiu, D.; Karniadakis, G.E. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **2002**, *24*, 619–644.
19. Xiu, D.; Hesthaven, J.S. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **2005**, *27*, 1118–1139.
20. Xiu, D. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*; Princeton University Press: Princeton, NJ, USA, 2010.
21. Le Maître, O.P.; Reagan, M.T.; Najm, H.N.; Ghanem, R.G.; Knio, O.M. A stochastic projection method for fluid flow: II. Random process. *J. Comput. Phys.* **2002**, *181*, 9–44.
22. Ghiocel, D.M.; Ghanem, R.G. Stochastic finite-element analysis of seismic soil-structure interaction. *J. Eng. Mech.* **2002**, *128*, 66–77.
23. Kennedy, M.C.; O'Hagan, A. Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2001**, *63*, 425–464.
24. Cressie, N. Statistics for spatial data: Wiley series in probability and statistics. *Wiley-Intersci. N. Y.* **1993**, *15*, 105–209.
25. MacKay, D.J. Introduction to Gaussian processes. *NATO ASI Ser. F Comput. Syst. Sci.* **1998**, *168*, 133–166.
26. Rasmussen, C.E. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*; Springer: Berlin, Germany, 2004; pp. 63–71.
27. Constantine, P.G.; Wang, Q. Residual minimizing model interpolation for parameterized nonlinear dynamical systems. *SIAM J. Sci. Comput.* **2012**, *34*, A2118–A2144.

28. Quiñonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.
29. Schobi, R.; Sudret, B.; Wiart, J. Polynomial-chaos-based kriging. *Int. J. Uncertain. Quantif.* **2015**, *5*, 171–193.
30. Schöbi, R.; Sudret, B.; Marelli, S. Rare Event Estimation Using Polynomial-Chaos kriging. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part A Civ. Eng.* **2017**, *3*, D4016002.
31. Schöbi, R.; Sudret, B. Combining polynomial chaos expansions and kriging for solving structural reliability problems. In Proceedings of the 7th International Conference on Computational Stochastic Mechanics (CSM7), Santorini, Greece, 15–18 June 2014.
32. Schöbi, R.; Sudret, B. PC-kriging: A new meta-modeling method and its applications to quantile estimation. In Proceedings of the 17th IFIP Working Group 7.5 Conference on Reliability and Optimization of Structural Systems, Huangshan, China, 3–7 July 2014.
33. Aronszajn, N. Theory of reproducing kernels. *Trans. Am. Math. Soc.* **1950**, *68*, 337–404.
34. Kullback, S. *Information Theory and Statistics*; Courier Corporation: North Chelmsford, MA, USA, 1997.
35. Echard, B.; Gayton, N.; Lemaire, M. AK-MCS: An active learning reliability method combining kriging and Monte Carlo simulation. *Struct. Saf.* **2011**, *33*, 145–154.
36. Dubourg, V. Adaptive Surrogate Models for Reliability Analysis and Reliability-Based Design Optimization. Ph.D. Thesis, Université Blaise Pascal-Clermont-Ferrand II, Aubière, France, 2011.
37. Kibble, W. An extension of a theorem of Mehler's on Hermite polynomials. *Math. Proc. Camb. Philos. Soc.* **1945**, *41*, 12–15.
38. Rasmussen, C.E.; Nickisch, H. Gaussian processes for machine learning (GPML) toolbox. *J. Mach. Learn. Res.* **2010**, *11*, 3011–3015.
39. Niederreiter, H. *QuasiMonte Carlo Methods*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.
40. Ishigami, T.; Homma, T. An importance quantification technique in uncertainty analysis for computer models. In Proceedings of the First International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 3–5 December 1990; pp. 398–403.
41. Marrel, A.; Iooss, B.; Van Dorpe, F.; Volkova, E. An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput. Stat. Data Anal.* **2008**, *52*, 4731–4744.
42. Oakley, J.E.; O'Hagan, A. Probabilistic sensitivity analysis of complex models: A Bayesian approach. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2004**, *66*, 751–769.