A unified, goal-oriented, hybridized reduced basis method and generalized polynomial chaos algorithm for partial differential equations with random inputs

Jiahua Jiang* Yanlai Chen* Akil Narayan†

Abstract

The generalized Polynomial Chaos (gPC) method using stochastic collocation is one of the most popular computational approaches for solving partial differential equations (PDEs) with random inputs. The main hurdle preventing its efficient direct application for high-dimensional randomness is that the solution ensemble size grows exponentially in the number of inputs (the "curse of dimensionality"). In this paper, we design a weighted version of the reduced basis method (RBM) and synergistically integrate it into the gPC framework. The resulting algorithm is capable of speeding up the traditional gPC by orders of magnitude without degrading its accuracy. Perhaps most importantly, the relative efficiency improves as the parametric dimension increases demonstrating the potential of our method in significantly delaying the curse of dimensionality. Theoretical results as well as numerical evidence justify these findings.

1 Introduction

Computational methods for stochastic problems in uncertainty quantification (UQ) are an increasingly-important area of research and much recent effort in this direction has been rewarded with many promising developments. In particular, algorithms that quantify the effect of random input parameters on solutions to differential equations have seen rapid advancement. One of the most widely used methods in this context is the generalized Polynomial Chaos (gPC) method [32], which constructs a parametric response surface using a polynomial representation. This method exploits parametric regularity of the system to achieve fast convergence rates [30]. With gPC, stochastic solutions are represented as expansions in orthogonal polynomials of the input random parameters, and so many stochastic algorithms concentrate on computation of the expansion coefficients in a gPC representation. Stochastic collocation is a popular non-intrusive approach to compute these coefficients, using a collection of interpolation or quadrature nodes in parameter space [31]. This requires one to query an expensive yet deterministic computational solver once for each parameter node. However, when the dimension of the random parameter is large, the number of required parameter nodes (and hence the number of computational solves) grows exponentially fast. This is a manifestation of the "curse of dimensionality".

*Department of Mathematics, University of Massachusetts Dartmouth, 285 Old Westport Road, North Dartmouth, MA 02747, USA. Emails: {jjiang, yanlai.chen}@umassd.edu. J. Jiang and Y. Chen are partially supported by National Science Foundation grant DMS-1216928.

†Scientific Computing and Imaging (SCI) Institute and Department of Mathematics, University of Utah, 72 S Campus Drive, Salt Lake City, UT 84112, USA. Email: akil@sci.utah.edu. A. Narayan is partially supported by NSF DMS-1552238, AFOSR FA9550-15-1-0467, and DARPA N660011524053

One popular strategy that combats the computational burden arising from multiple queries of an expensive model is model order reduction, which includes proper orthogonal decomposition (POD) methods, Krylov subspace methods, and reduced basis methods (RBM). We refer to [5] for a recent survey detailing some of these methods. Model reduction strategies allow one to replace an expensive computational model with an inexpensive yet accurate emulator for which performing a large number of queries is computationally feasible.

Such an approach appeals to the same motivation as POD methods: although the random inputs live in a high-dimensional space, the output of interest (such as the full solution field or integrated quantities of interest) frequently lie in a low-dimensional manifold [6, 9, 25]. The search for, identification, and exploitation of this low-dimensional manifold are the central goals of many model order reduction strategies. Assuming such a low-dimensional manifold exists, then it may be possible to build a reduced-complexity emulator and consequently form the sought accurate gPC approximation. In this paper we employ the RBM model reduction strategy, for which [6, 16, 28] are good references with [1, 2, 4, 18] the appropriate historical references.

The Reduced Basis Method performs a projection onto "snapshots", i.e., a small and carefully chosen selection of the most representative high-fidelity solutions. These snapshots are selected via a greedy algorithm that appeals to an *a posteriori* error estimate [17, 28]. The computational methods that one uses to compute high-fidelity snapshots include typical legacy solvers, like collocation or finite element discretizations. The ingredient in RBM that allows for computational savings is the "offline-online" decomposition. The offline stage is the more expensive part of the algorithm where a small number of parameter values are chosen and the snapshots are generated by executing the expensive high-fidelity computational model at these parameter locations. (Typically $\mathcal{O}(10)$ such evaluations are necessary.) The preparation completed during the offline stage allows very efficient evaluation of an emulator of the high-fidelity model during the online stage. During the online stage, each evaluation of the emulator can typically be computed 100-1000 times faster than evaluation of the original expensive model. One of the major benefits of RBM that we exploit in this paper is that the RBM model reduction is *rigorous*: Certifiable error bounds accompany construction of the emulator in the offline stage [28].

The idea of utilizing the RBM for problems in a general uncertainty quantification framework is not new [7, 8, 10, 14, 20, 27]. However in this paper our ultimate aim is to form a gPC expansion. The use of the RBM in this context and the exploration of its effectiveness in high-dimensional random space are underdeveloped to the best of our knowledge. A naïve stochastic collocation method is computationally infeasible in high-dimensional parameter spaces, even when employing a sparse grid of economical cardinality. But the hybrid gPC-RBM algorithm we propose is able to reduce the computational complexity to a manageable load, and thus enables construction of the gPC approximation in high-dimensional parameter spaces with rigorous error bounds.

This paper introduces, refines, and extends the idea of combining a goal-oriented Reduced Basis Method with a generalized Polynomial Chaos expansion. Our framework is *goal-oriented*: the construction of the approximation is optimized with a user-specifiable quantity of interest in mind. The algorithm is *rigorous*: we can guarantee an error tolerance for general quantities of interest. Our numerical results indicate that our method improves in performance (efficiency) as the parametric dimension increases – this suggests that our method is particularly useful for delaying the curse of dimensionality.

In section 2, we introduce the general framework of a PDE with stochastic input data. The two major ingredients in our approach, gPC and RBM, are likewise discussed. In section 3 we introduce the hybrid algorithm which is analyzed in section 4. Our numerical results are collected in section 5, which verify the efficiency and convergence of the hybrid algorithm.

2 Background

In this section, we introduce the necessary background material of the hybrid algorithm, namely, generalized Polynomial Chaos and the Reduced Basis Method.

2.1 Problem setting

Probability framework: Let $\boldsymbol{\mu} = (\mu_1, ...\mu_K)$ be a K-variate random vector with independent components on a complete probability space $(\Omega, \mathcal{B}, \mathcal{P})$, with Ω the sample space equipped with the σ-algebra $\mathcal{B}$, and $\mathcal{P}$ a probability measure. For $\Gamma_i = \mu_i(\Omega)$ the state space of μ_i, the probability density function of the random variable μ_i is denoted $\rho_i : \Gamma_i \to \mathbb{R}^+$. Since the components of $\boldsymbol{\mu}$ are mutually independent, then

$$\boldsymbol{\rho}(\boldsymbol{\mu}) = \Pi_{i=1}^{K} \rho_i(\mu_i) \tag{2.1}$$

is the joint probability density function of random vector $\boldsymbol{\mu}$. The image of $\boldsymbol{\mu}$ is

$$\boldsymbol{\Gamma} = \bigoplus_{i=1}^{K} \Gamma_i \subset \mathbb{R}^K$$

Partial Differential Equation with random parameters: Let $D \subset R^d \, (d = 1, 2, 3)$ be an open set in the physical domain with boundary ∂D, and $\boldsymbol{x} = (x_1, ...x_d) \in D$ be a point in this set. We consider the problem of finding the solution $u : D \times \boldsymbol{\Gamma} \to \mathbb{R}$ of the following stochastic PDE:

$$\begin{cases} \mathcal{L}(\boldsymbol{x}, u, \boldsymbol{\mu}) = f(\boldsymbol{x}, \boldsymbol{\mu}), & \forall \, (\boldsymbol{x}, \boldsymbol{\mu}) \in D \times \Gamma, \\ \mathcal{B}(\boldsymbol{x}, u, \boldsymbol{\mu}) = g(\boldsymbol{x}, \boldsymbol{\mu}), & \forall \, (\boldsymbol{x}, \boldsymbol{\mu}) \in \partial D \times \Gamma. \end{cases} \tag{2.2}$$

Here $\mathcal{L}$ is a differential operator defined on domain D and $\mathcal{B}$ is a boundary operator defined on the boundary ∂D. The functions f and g represent the forcing term and the boundary conditions, respectively.

We require the problem (2.2) to have well-posed solutions in a Hilbert space X. We thus assume $u(\cdot\,; \boldsymbol{\mu}) \in X$ almost surely. The Hilbert space X is equipped with inner product $(\cdot, \cdot)_X$ and induced norm $\| \cdot \|_X$. A straightforward example is furnished when (2.2) corresponds to a canonical elliptic partial differential equation: X satisfies $H_0^1(D) \subset X \subset H^1(D)$, with H^1 the space of functions whose L^2-derivatives are square-integrable over D, and H_0^1 the space of functions in H^1 whose support is compact in D.

In most applications, one has access to a deterministic computational solver that, for each *fixed* value of $\boldsymbol{\mu}$, produces an approximate, discrete solution to (2.2). We assume that for this fixed $\boldsymbol{\mu}$, such a computational solver produces the discrete solution $u^{\mathcal{N}}$, which has $\mathcal{N}$ degrees of freedom. This discrete solution is obtained by solving a discretized version of (2.2). For a fixed $\boldsymbol{\mu} \in \boldsymbol{\Gamma}$, this is given by

$$\begin{cases} \mathcal{L}^{\mathcal{N}}(u^{\mathcal{N}}, \boldsymbol{\mu}) = f^{\mathcal{N}}(\boldsymbol{\mu}), \\ \mathcal{B}^{\mathcal{N}}(u^{\mathcal{N}}, \boldsymbol{\mu}) = g^{\mathcal{N}}(\boldsymbol{\mu}). \end{cases} \tag{2.3}$$

Standard legacy discretizations, such as finite element or spectral collocation solvers, can be written in this way. The continuous Hilbert space X is replaced with its discrete Hilbert space counterpart $X_{\mathcal{N}}$, with norm $\| \cdot \|_{X_{\mathcal{N}}}$.

As before, we assume that $u^{\mathcal{N}}(\boldsymbol{\mu}) \in X_{\mathcal{N}}$ almost surely. We will need an additional assumption that the norm of the solution is uniformly bounded as a function of the parameter. I.e.,

$$\left\| u^{\mathcal{N}}(\cdot, \boldsymbol{\mu}) \right\|_{X_{\mathcal{N}}} \leqslant U, \qquad \forall \boldsymbol{\mu} \in \boldsymbol{\Gamma} \tag{2.4}$$

This assumption is satisfied for many practical problems of interest. For example, for a linear elliptic operator $\mathcal{L}^{\mathcal{N}}(u^{\mathcal{N}})$, boundedness of the solution is a simple consequence of the bilinear weak form being coercive and the linear form being continuous [23]. In our setting, uniform coercivity of the bilinear form and uniform continuity of the linear form with respect to μ would be sufficient to guarantee the uniform boundedness (2.4).

When introducing the discretized PDE (2.3) we assume that, for any $\boldsymbol{\mu} \in \boldsymbol{\Gamma}$, $u^{\mathcal{N}}(\boldsymbol{\mu}) \approx u(\boldsymbol{x}, \boldsymbol{\mu})$, where the approximation has an acceptable level of accuracy. In practice, one requires $\mathcal{N} \gg 1$ to achieve this.

In what follows we will usually treat $\boldsymbol{\mu}$ as a parameter rather than as an explicitly random quantity. This is a standard approach, and is without loss since all of our statements can be framed in the language of probability by appropriate change of notation. (E.g., ρ-weighted integrals are expectations.)

2.2 Generalized Polynomial Chaos

The Generalized Polynomial Chaos method is a popular technique for solving stochastic PDE and representing stochastic processes. The main idea of the gPC method is to seek an approximation of the exact solution of the PDE (2.2) by assuming that the dependence on $\boldsymbol{\mu}$ is efficiently represented by a $\boldsymbol{\mu}$-polynomial. If u depends smoothly on $\boldsymbol{\mu}$, then exponential convergence with respect to the polynomial degree can be achieved [30]. Computational implementations of gPC use an expansion in an orthogonal polynomial basis; as a consequence, quantities of interest such as expected value and variance can be efficiently evaluated directly from expansion coefficients.

2.2.1 gPC basis

Consider one-dimensional parameter space Γ_i corresponding to the random variable μ_i. If μ_i has finite moments of all orders, then there exists a collection of orthonormal polynomials $\{\phi_m(\cdot)\}_{m=0}^{\infty}$, with ϕ_m a polynomial of degree m, such that

$$\mathbb{E}\left[\phi_m(\mu_i)\phi_n(\mu_i)\right] = \int_{\Gamma_i} \rho_i(\mu_i)\phi_m(\mu_i)\phi_n(\mu_i)d\mu_i = \delta_{m,n}$$

where $\delta_{m,n}$ is the Kronecker delta function. The type of orthogonal polynomial basis $\{\phi_m\}$ depends on the distribution of μ_i. For instance, if μ_i is uniformly distributed in $[-1, 1]$, its probability density function ρ_i is a constant and $\{\phi_m\}_{m=0}^{\infty}$ is the set of orthonormal Legendre polynomials. Several well-studied orthogonal polynomial families correspond to standard probability distributions [30]. The correspondence for common probability distributions is shown in Table 1.

For the K-dimensional case ($K > 1$), an orthogonal polynomial family associated to the full tensor-product density $\rho(\boldsymbol{\mu})$ can be formed from products of univariate polynomials:

$$\Phi_\alpha(\boldsymbol{\mu}) = \phi_{\alpha_1}(\mu_1)...\phi_{\alpha_K}(\mu_K),$$

where $\alpha = (\alpha_1, \ldots, \alpha_K) \in \mathbb{N}_0^K$ is a multi-index. The degree of Φ_α is $|\alpha| = \sum_{k=1}^K \alpha_k$. Note that the $\boldsymbol{\Phi}_\alpha$ defined in this manner are indeed orthonormal:

$$\int \Phi_\alpha(\boldsymbol{\mu})\Phi_\beta(\boldsymbol{\mu})\boldsymbol{\rho}(\boldsymbol{\mu})d\boldsymbol{\mu} = \prod_{k=1}^{K} \int \phi_{\alpha_k}\phi_{\beta_k}\rho_k(\mu_k)\mathrm{d}\mu = \prod_{k=1}^{K} \delta_{\alpha_k,\beta_k} = \delta_{\alpha,\beta}$$

Random variable distribution	gPC polynomial basis	Support
Gaussian	Hermite	$(-\infty, \infty)$
Gamma	Laguerre	$[0, \infty)$
Beta	Jacobi	$[-1, 1]$
Uniform	Legendre	$[-1, 1]$
Poisson	Charlier	$\{0, 1, 2, 3, ...\infty\}$
Binomial	Krawtchouk	$\{0, 1, 2, 3, ..., N\}$
Negative Binomial	Meixner	$\{0, 1, 2, 3, ...\infty\}$
Hypergeometric	Hahn	$\{0, 1, 2, 3, ..., N\}$

Table 1: Various probability distributions with their corresponding gPC polynomial family and support.

A standard polynomial space to consider in the multivariate setting is the *total degree* space, formed from the span of all Φ_α whose degree is less than a given $P \in \mathbb{N}$:

$$\mathcal{U}_K^P \equiv \text{span}\left\{\Phi_\alpha \;\middle|\; |\alpha| \leqslant P\right\}$$

The dimension of $\mathcal{U}_K^P$ is

$$\dim(\mathcal{U}_K^P) = \left(\begin{array}{c} K+P \\ K \end{array} \right), \tag{2.5}$$

which grows comparably to P^K for large K. In what follows, we will index multivariate orthonormal polynomials as either Φ_α with $\alpha \in \mathbb{N}_0^K$ satisfying $|\alpha| \leqslant P$, or Φ_m with $m \in \mathbb{N}$ satisfying $1 \leqslant m \leqslant \dim \mathcal{U}_K^P$. To achieve this, we assume any ordering of multi-indices α that preserves a partial ordering of the total degree (for example, graded lexicographic ordering).

2.2.2 gPC approximation and quadrature

The $L^2_\rho(\Gamma)$-optimal gPC approximation of the solution $u(\boldsymbol{x}, \boldsymbol{\mu})$ to (2.2) in the space $\mathcal{U}_K^P$ is the $L^2_\rho(\Gamma)$-orthogonal projection onto $\mathcal{U}_K^P$, given by

$$u_K^P(\boldsymbol{x}, \boldsymbol{\mu}) = \sum_{m=1}^{M} \widetilde{u}_m(\boldsymbol{x})\Phi_m(\boldsymbol{\mu}), \quad M = \dim(\mathcal{U}_K^P) = \left(\begin{array}{c} K+P \\ K \end{array} \right). \tag{2.6}$$

The Fourier coefficient functions $\widetilde{u}_m (1 \leqslant m \leqslant M)$ are defined as

$$\widetilde{u}_m(\boldsymbol{x}) = \int u(\boldsymbol{x}, \boldsymbol{\mu})\Phi_m(\boldsymbol{\mu})\boldsymbol{\rho}(\boldsymbol{\mu})d\boldsymbol{\mu}. \tag{2.7}$$

For any $\boldsymbol{x} \in D$, the mean-square error in this finite-order projection is

$$\begin{aligned} \mathcal{E}_{\text{gPC}}(\boldsymbol{x}) &= \left\| u(\boldsymbol{x}, \boldsymbol{\mu}) - u_K^P(\boldsymbol{x}, \boldsymbol{\mu}) \right\|_{L^2_{\boldsymbol{\rho}}(\boldsymbol{\Gamma})} \\ &= \left(\int_{\boldsymbol{\Gamma}} (u(\boldsymbol{x}, \boldsymbol{\mu}) - u_K^P(\boldsymbol{x}, \boldsymbol{\mu}))^2 \boldsymbol{\rho}(\boldsymbol{\mu}) d\boldsymbol{\mu} \right)^{1/2}. \end{aligned} \tag{2.8}$$

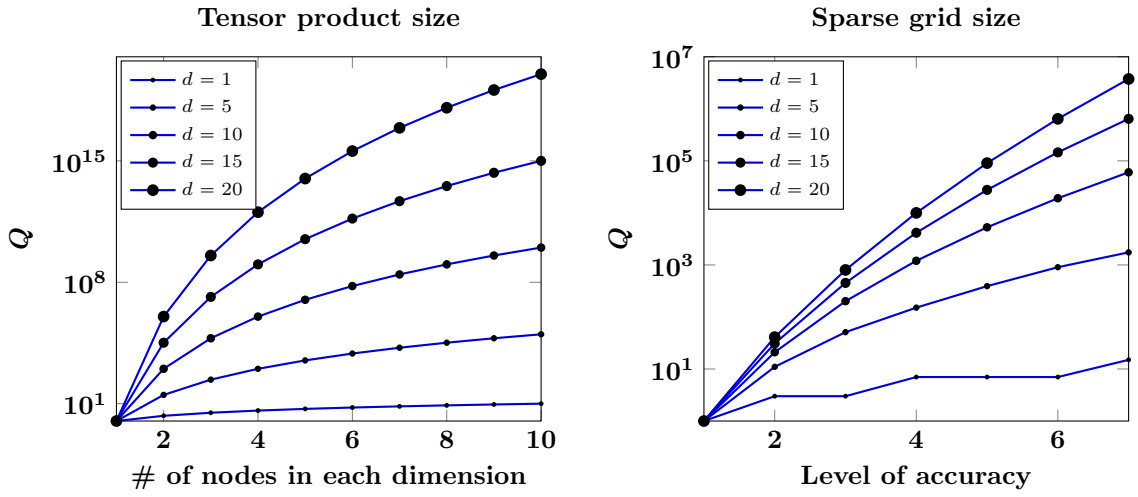

Figure 1: Tensor-product vs sparse grid (Gauss-Patterson-based sparse grid) quadrature rule sizes

Note that this error is usually not achievable in practice: The Fourier coefficients $\widetilde{u}_m$ cannot be computed without essentially full knowledge of the solution u. Therefore, one frequently resorts to approximating these coefficients. One popular non-intrusive method is quadrature-based stochastic collocation, where the integral in (2.7) is approximated by a quadrature rule.

Toward that end, let $\{\boldsymbol{\mu}^q, w_q\}_{q=1}^Q$ denote quadrature nodes and weights, respectively, for a quadrature rule that implicitly defines a new empirical probability measure:

$$\int_{\boldsymbol{\Gamma}} f(\boldsymbol{\mu})\rho(\boldsymbol{\mu})\mathrm{d}\boldsymbol{\mu} \approx \sum_{q=1}^{Q} w_q f\left(\boldsymbol{\mu}^q\right). \tag{2.9}$$

For example, two common choices for quadrature rules are tensor-product Gauss quadrature rules, and Gauss-Patterson-based sparse grid quadrature rules. Each of these rules can effectively integrate polynomials of high degree, but the requisite size of the quadrature rule Q is large in high dimensions: See Figure 1.

With this quadrature rule, the Fourier coefficients can be approximated by

$$\widetilde{u}_m \approx \widehat{u}_m = \sum_{q=1}^{Q} u\left(\boldsymbol{x}, \boldsymbol{\mu}^q\right) \Phi_m(\boldsymbol{\mu}^q) w_q. \tag{2.10}$$

The advantage of this formulation is that we need only compute the quantities $u\left(\cdot, \boldsymbol{\mu}^q\right)$, which are a collection of solutions to a deterministic PDE. Since this is all done in the context of a computational solver given by (2.3), one will replace the continuous solution $u(\cdot, \boldsymbol{\mu})$ with the discrete solution $u^{\mathcal{N}}(\boldsymbol{\mu})$.

Then a straightforward stochastic collocation approach first collects the solution ensemble from the computational solver,

$$\left\{u_q^{\mathcal{N}}\left(\boldsymbol{x}\right)\right\}_{q=1}^{Q} \triangleq \left\{u^{\mathcal{N}}\left(\boldsymbol{x}, \boldsymbol{\mu}^q\right)\right\}_{q=1}^{Q} \tag{2.11}$$

then computes the approximate Fourier coefficients

$$\widehat{u}_m^{\mathcal{N}} = \sum_{q=1}^{Q} u_q^{\mathcal{N}}(\boldsymbol{x})\,\Phi_m(\boldsymbol{\mu}^q) w_q, \tag{2.12}$$

and finally forms the full approximation

$$u(\boldsymbol{x},\boldsymbol{\mu}) \approx u^{\mathcal{N}}(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{m=1}^{M} \widehat{u}_m^{\mathcal{N}}(\boldsymbol{x})\Phi_m(\boldsymbol{\mu}). \tag{2.13}$$

Note that, in order for the quadrature approximation (2.12) to be reasonably accurate, the number of quadrature points Q should be comparable with M. We already know from (2.5) that M scales like P^K. A canonical type of problem for the system (2.2) and its resulting discretization (2.3) is a linear elliptic PDE. In this case, the cost of obtaining each $u_q^{\mathcal{N}}$ requires at least $\mathcal{O}(\mathcal{N})$ computational effort. (In some cases $\mathcal{O}(\mathcal{N}^3)$ effort is required.) Q solves of the PDE are required, with each solve costing at least $\mathcal{O}(\mathcal{N})$ work. Since $Q \sim M \sim P^K$, then in the best-case scenario the total work scales like $\mathcal{O}\left(\mathcal{N}P^K\right)$. Thus, the requisite computational effort for a straightforward stochastic collocation method is infeasible when the random parameter $\boldsymbol{\mu}$ is high-dimensional.

However, if one could construct the approximation (2.13), then it is usually extremely accurate. The focus of this paper is to *inexpensively* achieve an approximation whose error is comparable to that from the projected gPC coefficients (2.13). The rest of this paper provides an algorithmic method that allows one to approximate (2.13) in a computationally feasible manner. The essential ingredient is replacement of $u_q^{\mathcal{N}}$ by an accurate surrogate that is much cheaper to compute.

2.2.3 Quantities of Interest

In many UQ scenarios, one is not necessarily interested in the entire solution field $u(\boldsymbol{x},\boldsymbol{\mu})$, but rather some other quantity of interest derived from it. We introduce a functional $\mathcal{F}$ that serves to map the solution u to the quantity of interest (the "goal"). Our theoretical results require two assumptions on the quantity of interest map $\mathcal{F}$: that it has affine dependence on an M-term gPC expansion, and that the affine terms are Lipschitz continuous in the sense described below. We demonstrate in this section that these assumptions are not restrictive. Our construction exploits the well-known property of gPC that common quantities of interest such as the mean field and variance field can be exactly recovered by simple manipulation of the gPC coefficients [30].

The first assumption we make is that $\mathcal{F}$ has affine dependence on an M-term gPC expansion, specifically

$$\mathcal{F}\left[u_M\right] = \mathcal{F}\left[\sum_{m=1}^{M} \widehat{u}_m(\boldsymbol{x})\phi_m(\boldsymbol{\mu})\right] = \sum_{m=1}^{M} \theta_{\mathcal{F}}(\widehat{u}_m(\boldsymbol{x}))\mathcal{F}\left[\phi_m(\boldsymbol{\mu})\right]. \tag{2.14}$$

It is not hard to show that typical quantities of interest satisfy this condition on $\mathcal{F}$ with simple coefficient functions $\theta_{\mathcal{F}}$:

- $\mathcal{F}$ is expected value $\mathbb{E}$ and $\theta_{\mathcal{F}}$ is the identity function,

$$\mathcal{F}\left[u_M\right] = \mathbb{E}\left[u_M(\boldsymbol{x},\boldsymbol{\mu})\right] = \sum_{m=1}^{M} \widehat{u}_m(\boldsymbol{x})\mathbb{E}\left[\phi_m(\boldsymbol{\mu})\right] = \widehat{u}_1 \tag{2.15a}$$

- $\mathcal{F}$ is the variance field, with $\theta_{\mathcal{F}}$ the quadratic function $\theta_{\mathcal{F}}(v) = v^2$,

$$\mathcal{F}\left[u_M\right] = \mathrm{var}(u_M(\boldsymbol{x},\boldsymbol{\mu})) = \sum_{m=1}^{M} (\widehat{u}_m(\boldsymbol{x}))^2 \mathrm{var}\left[\phi_m(\boldsymbol{\mu})\right] = \sum_{m=2}^{M} (\widehat{u}_m(\boldsymbol{x}))^2 \tag{2.15b}$$

- $\mathcal{F}$ is the norm-squared operator $\|\cdot\|^2_{L^2_\rho}$ defined in (2.8) and $\theta_\mathcal{F}$ is again the quadratic $\theta_\mathcal{F}(v) = v^2$,

$$\mathcal{F}[u_M] = ||u_M(\boldsymbol{x},\boldsymbol{\mu})||^2_{L^2_\rho} = \sum_{m=1}^{M} (\widehat{u}_m(\boldsymbol{x}))^2 ||\phi_m(\boldsymbol{\mu})||_{L^2_\rho} = \sum_{m=1}^{M} (\widehat{u}_m(\boldsymbol{x}))^2 \tag{2.15c}$$

Thus, our assumption (2.14) is not too restrictive.

Our theoretical results also require the second assumption that the functional $\theta_\mathcal{F}$ is Lipschitz continuous with Lipschitz constant C_{Lip}, i.e., that

$$\|\theta_\mathcal{F}(v) - \theta_\mathcal{F}(w)\| \leqslant C_{\text{Lip}} \|v - w\|, \tag{2.16}$$

for all appropriate inputs v and w. For $\mathcal{F} = \mathbb{E}$, this constant is $C_{\text{Lip}} = 1$. For the latter cases of $\mathcal{F} = \text{var}$ and $\mathcal{F}[\cdot] = \|\cdot\|_{L^2_\rho}$ where $\theta_\mathcal{F}(v) = v^2$, then $C_{\text{Lip}} = 2U$, where U is the uniform bound in (2.4).

2.3 Reduced Basis Method (RBM)

The reduced basis method is one of the most widely used model order reduction strategies to solve a parameterized PDE with a large number of different parameter configurations. RBM seeks to form an approximation u^N satisfying

$$u^N(\boldsymbol{\mu}) \approx u^\mathcal{N}(\boldsymbol{\mu}), \qquad\qquad \boldsymbol{\mu} \in \boldsymbol{\Gamma},$$

such that the surrogate u^N can be computed with an algorithm whose complexity depends only on N, in contrast to the full solution $u^\mathcal{N}$ whose complexity depends on $\mathcal{N} \gg N$. In this section, we present the key ingredients of RBM, including the greedy algorithm for the construction of the reduced basis space, the *a posteriori* error estimate, and the efficient offline-online computational decomposition. The RBM algorithm will be a central part of the novel hybrid approach that we present in Section 3.

2.3.1 Reduced basis approximation

We take the general problem (2.2) for presentation of reduced basis approximations. To simplify the presentation, we assume that the boundary condition is homogeneous Dirichlet and the differential operator $\mathcal{L}$ is linear, affine with respect to functions of $\boldsymbol{\mu}$. However, there are constructive remedies available for non-affine and non-linear operators [3, 19, 26].

We recall from the discussion in Section 2.1 that a computational solver in (2.3) uses $\mathcal{N} \gg 1$ degrees of freedom to produce $u^\mathcal{N}_q(\boldsymbol{x})$, which is deemed an acceptably accurate approximation to $u(\boldsymbol{x},\boldsymbol{\mu}^q)$. In the RBM context, this approximation is called the *truth solution* or *truth approximation* and we will use this terminology when appropriate. The starting point for developing computational reduced basis methods is to replace the expensive truth solution with an inexpensive reduced-order solution. We briefly describe the standard method for accomplishing below. For simplicity we assume a linear operator $\mathcal{L}$, but note that extensions to nonlinear problems exist [19].

Assume that a dense training set of parameter samples $\Xi \in \boldsymbol{\Gamma}$ is given such that the $\boldsymbol{\mu}$-variation of the solution u is accurately captured by the resulting truth solution ensemble $\{u^\mathcal{N}(\boldsymbol{x},\boldsymbol{\mu}) : \boldsymbol{\mu} \in \Xi\}$. In the framework of this paper, we take Ξ to be the quadrature rule nodal set introduced in (2.9), that is, $\Xi = \{\boldsymbol{\mu}^q\}_{q=1}^Q$.

For any given reduced-order dimension $N \ll \mathcal{N}$, we build the N-dimensional reduced basis space X^N by a greedy algorithm. The reduced basis space is constructed as a span of "snapshots" (i.e., truth solutions) based on judiciously chosen samples from the training set Ξ [24]

$$X^N = \text{span}\{u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\nu}^1), ..., u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\nu}^N)\}, \tag{2.17}$$

where $\{\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^N\} \subset \Xi$. For a fixed $\boldsymbol{\mu}^* \in \boldsymbol{\Gamma}$, the idea of RBM is to approximate the truth solution $u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^*)$ by its RB surrogate, which is formed as an element of X^N. The RBM surrogate $u^N(\boldsymbol{x}, \boldsymbol{\mu}^*)$ can thus be represented as

$$u^N(\boldsymbol{x}, \boldsymbol{\mu}^*) = \sum_{k=1}^{N} c_k(\boldsymbol{\mu}^*) u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^k) \tag{2.18}$$

By exploiting the linearity of the operator, RBM seeks to find coefficients $c_k(\boldsymbol{\mu}^*)$ such that the residual of (2.3) using the solution u^N,

$$\sum_{k=1}^{N} c_k(\boldsymbol{\mu}^*) \mathcal{L}^{\mathcal{N}}(\boldsymbol{x}, u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^k), \boldsymbol{\mu}^*) - f^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^*),$$

is as small as possible. The meaning of "small" is made precise by the prescription of an appropriate projection operator $\mathcal{P}$ such that the following holds

$$\mathcal{P}\Big(\sum_{k=1}^{N} c_k(\boldsymbol{\mu}^*) \mathcal{L}^{\mathcal{N}}(\boldsymbol{x}, u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^k), \boldsymbol{\mu}^*) \Big) = \mathcal{P}\Big(f^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^*) \Big). \tag{2.19}$$

Concrete examples of this abstract projection operator are the continuous L^2 projection onto X^N, a discrete ℓ^2 projection (least-squares) on the spatial mesh, or an empirical interpolation procedure [12, 13].

2.3.2 *A posteriori* error estimate

Error estimates play a crucial role in computational procedures for RBM algorithms. These estimates allow one to choose the parameter values $\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^N$ in (2.17) in an efficient and accurate way. Let $R : D \times \boldsymbol{\Gamma} \to \mathbb{R}$ denote the (Riesz representation of the) truth discretization residual in (2.3) with the reduced-order solution u^N, defined as

$$R_N(\cdot; \boldsymbol{\mu}) = f^{\mathcal{N}}(\cdot\,; \boldsymbol{\mu}) - \mathcal{L}^{\mathcal{N}}(u^N(\boldsymbol{x}, \boldsymbol{\mu}); \boldsymbol{\mu}). \tag{2.20}$$

The goal is to use knowledge of R_N to quantify the error in an RBM surrogate. The surrogate error is given by $e(\boldsymbol{x}, \boldsymbol{\mu}) := u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}) - u^N(\boldsymbol{x}, \boldsymbol{\mu})$ and, due to linearity of $\mathcal{L}^N$, satisfies

$$\mathcal{L}^{\mathcal{N}}(e(\boldsymbol{x}, \boldsymbol{\mu}), \boldsymbol{\mu}) = R(\cdot; \boldsymbol{\mu}) \tag{2.21}$$

To obtain a computable error bound for $e(\boldsymbol{x}, \boldsymbol{\mu})$, we let $\beta_{LB}(\boldsymbol{\mu})$ be a lower bound for the smallest eigenvalue of $\mathcal{L}^{\mathcal{N}}(\boldsymbol{\mu})^T \mathcal{L}^{\mathcal{N}}(\boldsymbol{\mu})$.

$$0 < \beta_{LB}(\boldsymbol{\mu}) \leqslant \min_{v} \frac{v^T \left(\mathcal{L}^{\mathcal{N}}(\boldsymbol{\mu})\right)^T \mathcal{L}^{\mathcal{N}}(\boldsymbol{\mu}) v}{\|v\|_{X_{\mathcal{N}}}} \tag{2.22}$$

Here $\mathcal{L}^{\mathcal{N}}(\boldsymbol{\mu})$ should be understood as the matrix representation of $\mathcal{L}^{\mathcal{N}}(\cdot\,; \boldsymbol{\mu})$. The relations (2.21) and (2.22) can be used to conclude [13]

$$\|e(\boldsymbol{x}, \boldsymbol{\mu})\|_{X_{\mathcal{N}}} \leqslant \frac{\|R_N(\boldsymbol{x}, \boldsymbol{\mu})\|_{X_{\mathcal{N}}}}{\sqrt{\beta_{LB}(\boldsymbol{\mu})}} \qquad \forall \boldsymbol{\mu} \in \Xi \tag{2.23}$$

Therefore, we can define a rigorous *a posteriori* error estimate as

$$\Delta_N(\boldsymbol{\mu}) = \frac{\|R_N(\boldsymbol{x},\boldsymbol{\mu})\|_{X_\mathcal{N}}}{\sqrt{\beta_{LB}(\boldsymbol{\mu})}} \geqslant \|e(\boldsymbol{x},\boldsymbol{\mu})\|_{X_\mathcal{N}} \qquad \forall \boldsymbol{\mu} \in \Xi. \tag{2.24}$$

We see that it is computationally tractable to bound the RBM surrogate error so long as we can efficiently compute the residual R_N along with β_{LB}. The residual can indeed be computed with $\mathcal{O}(N)$ complexity (see Section 2.3.4). The efficient and feasible evaluation of $\beta_{LB}(\boldsymbol{\mu})$ can be accomplished via the successive constraint linear optimization method (SCM) [11, 17, 21, 22] with the marginal computational cost for each $\boldsymbol{\mu}$ independent of the truth solution complexity $\mathcal{N}$. With the ability to efficiently compute Δ_N, we can describe the greedy algorithm for choosing the RBM parameter snapshot locations $\{\boldsymbol{\nu}^k\}$.

2.3.3 Greedy Algorithm

Given the training parameter set Ξ, the greedy algorithm deals with an optimization problem in a greedy way, seeking a new parameter $\boldsymbol{\nu}^{k+1} \in \Xi$ such that

$$\boldsymbol{\nu}^{k+1} = \underset{\boldsymbol{\mu}\in\Xi}{\operatorname{argmax}}\, \Delta_k(\boldsymbol{\mu}) \tag{2.25}$$

Guided by the error estimate $\Delta_k(\boldsymbol{\mu})$, the parameter values that accurately represent the solution manifold will not be omitted.

Algorithm 1 Greedy algorithm for construction of an RBM approximation

1: Input: training set Ξ;
2: Input: stopping criterion tolerance ϵ_{tol};
3: Randomly select the first sample $\boldsymbol{\mu}^{\mathbf{1}} \in \Xi$;
4: Obtain truth solution $u^\mathcal{N}\left(\boldsymbol{x},\boldsymbol{\nu}^1\right)$, set $X_1 = \text{span}\left\{u^\mathcal{N}(\boldsymbol{x},\boldsymbol{\nu}^1)\right\}$;
5: Set $N = 1$ and $\Delta_{\max} = \infty$;
6: **while** $\Delta_{\max} > \epsilon_{\text{tol}}$ **do**
7: **for** each $\mu \in \Xi$ **do**
8: Obtain RBM solution $u^N(\boldsymbol{x},\boldsymbol{\mu})$ by computing $c_j(\boldsymbol{x})$ that satisfy (2.19)
9: Compute *a posteriori* error estimate $\Delta_N(\boldsymbol{\mu})$
10: **end for**
11: Set $\boldsymbol{\nu}^{N+1} = \text{argmax}_{(\boldsymbol{\mu}\in\Xi)}\, \Delta_N(\boldsymbol{\mu})$, and $\Delta_{\max} = \Delta_N(\boldsymbol{\nu}^{N+1})$;
12: Obtain truth solution $u^\mathcal{N}(\boldsymbol{x},\boldsymbol{\nu}^{N+1})$ from (2.3) at $\boldsymbol{\mu} = \boldsymbol{\nu}^{N+1}$;
13: Augment the reduced basis space $X_{N+1} = \text{span}\left\{X_N \cup \{u^\mathcal{N}(\boldsymbol{x},\boldsymbol{\nu}^{N+1})\}\right\}$;
14: Set $N \leftarrow N+1$
15: **end while**

This process is repeated until the maximum of the error estimate is sufficiently small. At every step we choose the parameter whose surrogate error is largest.

Note that we need to query the truth solution at lines 4 and 12 of Algorithm 1. However, we only require N such solutions, with $N \ll |\Xi|$. Lines 8 and 9 can be completed efficiently with only $\mathcal{O}(N)$ complexity, as we describe in the next section.

2.3.4 Offline-online decomposition

The offline-online computational decomposition is the central idea that makes RBM so effective [29]. The core idea that permits this decomposition is that in many cases the operator $\mathcal{L}^{\mathcal{N}}$ and forcing term $f^{\mathcal{N}}$ in (2.3) exhibit *affine* dependence on the parameter, i.e.,

$$\mathcal{L}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_L} \boldsymbol{\theta}_q^L(\boldsymbol{\mu})\mathbb{L}_q, \qquad\qquad f(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \boldsymbol{\theta}_q^f(\boldsymbol{\mu})f_q(\boldsymbol{x}) \tag{2.26}$$

where the coefficient functions $\boldsymbol{\theta}_q^L$ and $\boldsymbol{\theta}_q^f$ depend only on the parameter $\boldsymbol{\mu}$, and the operators $\mathbb{L}_q$ and f_q are parameter-independent. (We recall again that the RBM can likewise handle non-affine problems [19].)

This affine assumption allows the RBM algorithm to be divided into two stages: *offline* and *online*. The offline stage has complexity that is $\mathcal{N}$-dependent and thus is expensive, but it is done only once. The online stage complexity is independent of $\mathcal{N}$, and thus it is inexpensive and is computationally feasible for a large number of inputs.

That the offline portion is $\mathcal{N}$-dependent is not surprising; what is remarkable is that the online portion, computation of the RBM surrogate u^N, can be accomplished with complexity depending only on $N \ll \mathcal{N}$. The essential idea is that by using the formula (2.18) along with the affine assumptions (2.26), then the discrete PDE (2.3) at any value $\boldsymbol{\mu}^*$ has the form

$$\sum_{k=1}^{N} c_k(\boldsymbol{\mu}^*) \sum_{q=1}^{Q_L} \boldsymbol{\theta}_q^L(\boldsymbol{\mu}^*)\underline{\underline{\mathbb{L}_q(u^{\mathcal{N}}(\boldsymbol{x},\boldsymbol{\mu}^k))}} = \sum_{q=1}^{Q_f} \boldsymbol{\theta}_q^f(\boldsymbol{\mu}^*)\underline{\underline{f_q(\boldsymbol{x})}}$$

We note that only the terms that are double-underlined require $\mathcal{N}$-dependent complexity to evaluate. However, these terms do *not* depend on $\boldsymbol{\mu}^*$, and so they may be computed and stored during the offline stage. Thus, for any $\boldsymbol{\mu}^*$, determining the RBM coefficients $c_k(\boldsymbol{\mu}^*)$ via the projection in (2.19) may be performed with a complexity that depends only on N, Q_L and Q_f. We refer to [13, 29] for more details.

3 Hybrid Algorithm

We recall the discussion from the end of section 2.2.2 that stochastic collocation can be computationally burdensome when the random parameter dimension K is large. The results in Figure 1 indicate that the number of stochastic collocation nodes easily exceeds the current capacity of computational power for high dimensions. For instance, a 20-dimensional problem using a 5-point-per-dimension quadrature rule requires $5^{20} \approx 10^{14}$ nodes. Although a sparse grid approach is more efficient than a tensor product grid, such a grid still has more than one million parameter values, requiring more than one million solves of (2.3). Therefore it still takes a onerous amount of time in practical engineering problems to achieve reasonable accuracy. Our approach ameliorates the cost-per-solve by using a gPC-goal-oriented variant of the RBM algorithm.

In this section, we develop a reduced basis method utilizing a modified *a posteriori* error estimate in the traditional RB greedy algorithm. We design this RBM-gPC hybrid to efficiently and accurately construct a gPC surrogate for the PDE system (2.3).

We avoid the direct hybridization of RBM and gPC, where RBM as introduced in section 2.3 would be used to give equal weight to all parameter values. This naïve approach would result in a gPC approximation that converges slowly in the L^2_ρ norm, requiring a large N (the dimension of the RBM surrogate). We exploit the observation that each $u(\boldsymbol{x},\boldsymbol{\mu}^q)$ associated with parameter

value $\boldsymbol{\mu}^q$ should have some quantitative measure of importance as indicated by the probability density $\rho(\boldsymbol{\mu}^q)$. This idea was explored [27], but our version differs notably from earlier methods since we do not explicitly use ρ as a weight for the *a posteriori* error estimate.

Instead, we propose and analyze a new "goal-oriented reduced basis method" adopting a *weighted a posteriori* error estimate. Rigorous error analysis is provided to validate the use of the weighted approach and the resulting gPC construction.

3.1 Weighted *a posteriori* error estimate

Appropriate design of the error estimate can critically determine the performance of any reduced basis method, particularly so for our goal-oriented approach. The approximate gPC coefficient formula (2.10) is the w_q-weighted inner product between the polynomial Φ_m and $u^{\mathcal{N}}$. Thus, the error estimate should likewise be weighted using the quadrature weight w_q. We emphasize again that our strategy is different from using the probability density function ρ as done in [27]; even in simple one-dimensional cases, it is easy to see that $w_q \not\approx \rho(\boldsymbol{\mu}^q)$ (cf., e.g., Gaussian quadrature or Clenshaw-Curtis quadrature).

We introduce the following cheap, dependable, and tight evaluation of reduced basis approximation, a weighted *a posteriori* error estimate $\Delta_N^w(\boldsymbol{\mu})$, given by

$$\Delta_N^w(\boldsymbol{\mu}^q) = \frac{\|R_N(\cdot,\boldsymbol{\mu}^q)\|_{X_{\mathcal{N}}}}{\sqrt{\beta_{\mathrm{LB}}(\boldsymbol{\mu}^q)}}\sqrt{Q|w_q|}, \qquad q = 1,\ldots,Q, \tag{3.1}$$

where $\beta_{LB}(\boldsymbol{\mu}^q)$ is a lower bound for the smallest eigenvalue, and R_N is the PDE residual of the order-N surrogate as defined in (2.20). Note that the novel quantity is the factor $\sqrt{Q|w_q|}$; since w_q corresponds to a Q-point normalized quadrature rule the quantity $Q|w_q|$ has $\mathcal{O}(1)$ magnitude, in principle. The absolute value bars are necessary in general because sparse grid quadrature rules can have negative weights. For a tensor-product Gaussian quadrature rule, the weights are all positive.

Having defined this weighted *a posteriori* error estimate, we can bound the error between the m-th truth Fourier coefficient $\widehat{u}_m^{\mathcal{N}}$ in (2.12) and its surrogate $\widehat{u}_m^N$

$$\widehat{u}_m^N = \sum_{q=1}^{Q} u^N(\boldsymbol{\mu}^q)\Phi_m(\boldsymbol{\mu}^q)w_q. \tag{3.2}$$

The precise estimate is below in section 4.

3.2 Goal-oriented greedy algorithm

Given the training parameter set Ξ and a current RB selection $\{\boldsymbol{\nu}^1,\ldots,\boldsymbol{\nu}^k\}$, the goal-oriented greedy algorithm, stated in Algorithm 2, aims to construct the reduced basis space in hierarchical manner by finding a new parameter $\boldsymbol{\nu}^{k+1} \in \Xi$ such that

$$\boldsymbol{\nu}^{k+1} = \underset{\boldsymbol{\mu}\in\Xi}{\operatorname{argmax}}\, \Delta_k^w(\boldsymbol{\mu}) \tag{3.3}$$

Guided by the weighted *a posteriori* error estimate $\Delta_k^w(\boldsymbol{\mu})$, the greedy algorithm chooses parameters by weighting them with the (square root of the) quadrature weights as shown in (3.1). We show later in Theorem 4.2 and Corollary 4.3 that this weighting allows one to guarantee that the gPC approximation that is formed from the RBM surrogates is within a user-defined tolerance of the gPC-truth approximation.

Another major difference in the goal-oriented algorithm is that the tolerance criterion is tuned to the quantity of interest of the gPC surrogate. At each stage with k RB snapshots, we compute the error estimate

$$\varepsilon = C_{Q,M}\sqrt{\frac{1}{Q}\sum_{q=1}^{Q}(\Delta_k^w)^2(\boldsymbol{\mu}^q)}, \text{ with } C_{Q,M} = \sum_{m=1}^{M}\sqrt{\sum_{q=1}^{Q}|w_q|\Phi_m^2(\boldsymbol{\mu}^q)}\,|\mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right]|\,, \tag{3.4}$$

where $\mathcal{F}$ denotes the quantity of interest as introduced in (2.14). Note that the constant $C_{Q,M}$ is computable *independent* of the solution u, and depends only on the choice of quadrature rule and quantity of interest. (See Lemma 4.6, and the discussion following Corollary 4.3.) As we show in Corollary 4.3, it turns out that ε is an upper bound on the error in the quantity of interest defined by $\mathcal{F}$ between the inexpensive RBM surrogate and the full expensive gPC stochastic collocation approximation.

Algorithm 2 Goal-oriented greedy algorithm

Input: training set Ξ with associated quadrature weights w_q;
Input: stopping criterion tolerance ε_{tol};
Input: goal-oriented constant $C_{Q,M} = \sum_{m=1}^{M} B_{Q,m}\mathcal{F}\left[\Phi_m\left(\boldsymbol{\mu}\right)\right]$.
Randomly select the first sample $\boldsymbol{\mu}^{\mathbf{1}} \in \Xi$;
Obtain truth solution $u^{\mathcal{N}}\left(\boldsymbol{x}, \boldsymbol{\mu}^1\right)$, set $X_1 = \text{span}\left\{u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^1)\right\}$;
Set $k = 1$ and $\varepsilon = \infty$;
while $\varepsilon > \varepsilon_{\text{tol}}$ **do**
    for each $\mu \in \Xi$ **do**
        Obtain RBM solution $u^k(\boldsymbol{x}, \boldsymbol{\mu})$ by computing $c_j(\boldsymbol{x})$ that satisfy (2.19)
        Compute weighted *a posteriori* error estimate $\Delta_k^w(\boldsymbol{\mu})$ from (3.1)
    end for
    Choose $\boldsymbol{\mu}^{k+1} = \text{argmax}_{(\boldsymbol{\mu}\in\Xi)}\,\Delta_k^w(\boldsymbol{\mu})$;
    augment the reduced basis space $X_{k+1} = X_k \cup \{u^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{\mu}^{k+1})\}$;
    Calculate $\Delta^{\text{sum}} = \sum_{\boldsymbol{\mu}\in\Xi}(\Delta_N^w)^2(\boldsymbol{\mu})$;
    Set $\varepsilon = C_{Q,M}\sqrt{\frac{1}{|\Xi|}\Delta^{\text{sum}}}$.;
    Set $N \leftarrow N + 1$;
end while

3.3 Goal-oriented hybridized RBM-gPC algorithm

We are now ready to present the unified, goal-oriented hybrid method; summary pseudocode is shown in Algorithm 3. The main idea of the hybrid algorithm is leveraging the goal-oriented RBM to alleviate the computational burden of obtaining the gPC coefficients. One major tool to accomplish that goal is splitting the work of solving (2.2) into two stages, one offline and one online.

The offline stage is Algorithm 2: first the RBM space is built by scanning the quadrature node set Ξ and evaluating the weighted *a posteriori* error estimate $\Delta_k^w\left(\boldsymbol{\mu}^q\right)$. Algorithm 2 utilizes the expensive PDE solver, solving (2.3) a total of N times over the parameter set $\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^N$. The offline phase stops when the computed error indicator ε defined in (3.4) falls below the user-defined tolerance ε_{tol}.

The online stage proceeds as indicated on line 5 of Algorithm 3. In this phase, the RBM surrogate that was constructed in the offline phase is evaluated several times to compute the approximate gPC coefficients $\left\{\widehat{u}_m^N\right\}_{m=1}^M$. This portion of the algorithm is *much* faster than naïve evaluation of (2.12); it is in this section of the procedure where the hybrid algorithm reaps computational savings compared to a traditional stochastic collocation approach.

Once the approximate coefficients $\widehat{u}_m^N$ are collected, the quantity of interest $\mathcal{F}\left[u_M^N\right]$ may be evaluated. As we show in the next section, a significant benefit of using the hybrid approach is that the error in this computed quantity of interest can be rigorously controlled by the user-defined input tolerance ε_{tol}.

Thus, the hybrid algorithm *both* achieves significant computational savings in construction of a gPC approximation *and* provides strict error bounds on quantities of interest.

Algorithm 3 Hybrid Algorithm

1: Input: the general stochastic PDE (2.2) and a tolerance ε_{tol}.

2: **Offline procedure:**

3: Set Ξ as the training set and use the goal-oriented greedy procedure with tolerance ε_{tol} (Algorithm 2) to compute N reduced basis elements $\{u^{\mathcal{N}}(\boldsymbol{x},\boldsymbol{\mu}^j)\}_{j=1}^N$.

4: **end Offline procedure**

5: Using the online RBM surrogate $u^N(\boldsymbol{x},\boldsymbol{\mu})$, compute M-term gPC coefficients and approximation of u^N:

$$\widehat{u}_m^N = \sum_{q=1}^{Q} w_q u^N\left(\boldsymbol{x},\boldsymbol{\mu}^q\right)\Phi_m\left(\boldsymbol{\mu}^q\right)$$

$$u_M^N(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{m=1}^{M} \hat{u}_m^N(\boldsymbol{x})\Phi_m(\boldsymbol{\mu}).$$

6: Output: Solution u_M^N, or quantity of interest $\mathcal{F}[u_M^N] = \sum_{m=1}^M \theta_{\mathcal{F}}[\widehat{u}_N^M(\boldsymbol{x})]\mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right]$.

4 Analysis of the hybrid algorithm

In this section, we show the convergence of this goal-oriented gPC-RBM algorithm. More precisely, we show that the error committed by the hybrid RBM algorithm is controlled by the user-defined input parameter ε_{tol} in Algorithm 3. Given a P-th order M-term gPC projection (2.6), the m-th truth Fourier coefficient $\widehat{u}_m^{\mathcal{N}}$ is evaluated by (2.12) and its surrogate $\widehat{u}_m^{\mathcal{N}}$ by (3.2). The truth and reduced basis stochastic solutions are then, respectively,

$$u_M^{\mathcal{N}}(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{m=1}^{M} \hat{u}_m^{\mathcal{N}}(\boldsymbol{x})\Phi_m(\boldsymbol{\mu}), \tag{4.1}$$

$$u_M^N(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{m=1}^{M} \hat{u}_m^N(\boldsymbol{x})\Phi_m(\boldsymbol{\mu}). \tag{4.2}$$

The properties of the RBM algorithm allow us to bound the error between the computationally expensive $\mathcal{F}\left[u_M^{\mathcal{N}}\right]$ and the efficiently computable hybrid surrogate $\mathcal{F}\left[u_M^N\right]$. This bound is our main

theoretical result and is shown in Theorem 4.2.

To begin, we first need to control the error between the full stochastic collocation gPC coefficients $u_m^{\mathcal{N}}(\boldsymbol{x})$ and its RBM surrogate $u_m^N(\boldsymbol{x})$.

Lemma 4.1. *Given a Q-point quadrature rule $\{\boldsymbol{\mu}^q, w_q\}_{q=1}^Q$ and an N-dimensional reduced basis approximation $u^N(\boldsymbol{x},\boldsymbol{\mu})$ for the solution $u(\boldsymbol{x},\boldsymbol{\mu})$ to the PDE (2.2), the error in the m-th gPC coefficient is given by*

$$\left\|\widehat{u}_m^{\mathcal{N}}(\boldsymbol{x}) - \widehat{u}_m^N(\boldsymbol{x})\right\|_{X_{\mathcal{N}}} \leqslant B_{Q,m}\sqrt{\frac{1}{Q}\sum_{q=1}^{Q}(\Delta_N^w)^2\left(\boldsymbol{\mu}^q\right)},$$

where $B_{Q,m}$ is the uncentered second moment of $\Phi_m(\boldsymbol{\mu})$ under the discrete measure defined by the quadrature rule:

$$B_{Q,m} = \sqrt{\sum_{q=1}^{Q}|w_q|\Phi_m^2(\boldsymbol{\mu}^q)}. \tag{4.3}$$

Proof. We use the quadrature representation for these functions:

$$\begin{aligned}
\left\|\widehat{u}_m^{\mathcal{N}}(\boldsymbol{x}) - \widehat{u}_m^N(\boldsymbol{x})\right\|_{X_{\mathcal{N}}} &= \left\|\sum_{q=1}^{Q} w_q\Phi_m\left(\boldsymbol{\mu}^q\right)\left[u^{\mathcal{N}}\left(\boldsymbol{x},\boldsymbol{\mu}^q\right) - u^N\left(\boldsymbol{x},\boldsymbol{\mu}^q\right)\right]\right\|_{X_{\mathcal{N}}} \\
&\leqslant \sum_{q=1}^{Q}\left|\sqrt{|w_q|}\Phi_m\left(\boldsymbol{\mu}^q\right)\right|\left(\sqrt{|w_q|}\left\|e(\boldsymbol{x},\boldsymbol{\mu}^q)\right\|_{X_{\mathcal{N}}}\right) \\
&\leqslant \sqrt{\sum_{q=1}^{Q}|w_q|\Phi_m^2\left(\boldsymbol{\mu}^q\right)}\sqrt{\sum_{q=1}^{Q}|w_q|\left\|e(\boldsymbol{x},\boldsymbol{\mu}^q)\right\|_{X_{\mathcal{N}}}^2} \\
&\leqslant B_{Q,m}\sqrt{\sum_{q=1}^{Q}\frac{1}{Q}(\Delta_N^w)^2\left(\boldsymbol{\mu}^q\right)}
\end{aligned}$$

□

This result bounds the error in each gPC coefficient as a product of two terms: the first term $B_{Q,m}$ indicates the accuracy of the quadrature rule, which is computable independent of the solution u. The second term is an average of the weighted *a posteriori* error estimate over parameter space.

We can now bound the RBM error in the quantity of interest.

Theorem 4.2. *Given an M-term gPC projection (2.6) and an N-dimensional reduced basis approximation (2.19), the error in the quantity of interest computed from the RBM-gPC approximation u_M^N, and that computed from the truth gPC approximation $u_M^{\mathcal{N}}$ is*

$$\left\|\mathcal{F}\left[u_M^N\right] - \mathcal{F}\left[u_M^{\mathcal{N}}\right]\right\|_{X_{\mathcal{N}}} \leqslant C_{\mathrm{Lip}}\,C_{Q,M}\sqrt{\frac{1}{Q}\sum_{q=1}^{Q}\left[\Delta_N^w(\boldsymbol{\mu}^q)\right]^2}, \tag{4.4}$$

where C_{Lip} is the Lipschitz constant of $\mathcal{F}$ defined in (2.16)*, and $C_{Q,M}$ is a constant independent of u, defined by*

$$C_{Q,M} = \sum_{m=1}^{M} B_{Q,m} \left| \mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right] \right|, \tag{4.5}$$

with $B_{Q,m}$ defined in (4.3)*.*

Proof. We begin by using our assumption (2.14) regarding the affine dependence of $\mathcal{F}$ on a gPC representation:

$$\mathcal{F}\left[u_M^N\right] - \mathcal{F}\left[u_M^{\mathcal{N}}\right] = \sum_{m=1}^{M} \left(\theta_{\mathcal{F}}(\widehat{u}_m^N(\boldsymbol{x})) - \theta_{\mathcal{F}}(\widehat{u}_m^{\mathcal{N}}(\boldsymbol{x}))\right) \mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right].$$

Applying triangle inequality and Lipschitz continuity of $\theta_{\mathcal{F}}(\cdot)$, we have

$$\left\| \mathcal{F}\left[u_M^N\right] - \mathcal{F}\left[u_M^{\mathcal{N}}\right] \right\|_{X_{\mathcal{N}}} \leqslant C_{\text{Lip}} \sum_{m=1}^{M} \left\| \widehat{u}_M^{\mathcal{N}}(\boldsymbol{x}) - \widehat{u}_M^N(\boldsymbol{x}) \right\|_{X_{\mathcal{N}}} \left| \mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right] \right|.$$

Using Lemma 4.1 on the right-hand side, we obtain

$$\left\| \mathcal{F}\left[u_M^N\right] - \mathcal{F}\left[u_M^{\mathcal{N}}\right] \right\|_{X_{\mathcal{N}}} \leqslant C_{\text{Lip}} \sum_{m=1}^{M} B_{Q,m} \left| \mathcal{F}\left[\Phi_m(\boldsymbol{\mu})\right] \right| \sqrt{\frac{1}{Q} \sum_{q=1}^{Q} (\Delta_N^w)^2(\boldsymbol{\mu}^q)},$$

and this proves (4.4). □

Corollary 4.3. *The output gPC approximation u_M^N from Algorithm 3 satisfies*

$$\left\| \mathcal{F}\left[u_M^N\right] - \mathcal{F}\left[u_M^{\mathcal{N}}\right] \right\|_{X_{\mathcal{N}}} \leqslant C_{\text{Lip}}\, \varepsilon$$

where ε is defined in (3.4).

Remark 4.4. The Lipschitz constant C_{Lip} is trivially 1 when $\mathcal{F}[\cdot]$ is the expected value operator. For the other two cases listed in (2.15b) and (2.15c), it is finite as long as we have uniform stability with respect to the parameter $\boldsymbol{\mu}$ for the computational solver (2.3). As the discussion around (2.4) indicates, this is a standard assumption, and in that case $C_{\text{Lip}} = 2U$.

Remark 4.5. We emphasize that $C_{Q,M}$ is a scalar whose value is *independent* of the solution u, its truth discretization $u^{\mathcal{N}}$, or the RBM surrogate u^N. It depends only on the choice of quadrature rule, the gPC order, and the choice of what quantity of interest is to be computed.

The coefficient $C_{Q,M}$ is not analytically computable in general since it depends on the chosen quadrature rule in parameter space. However, the next lemma shows that for quadrature rules of sufficiently high accuracy, $B_{Q,m} \equiv 1$.

Lemma 4.6. *Let $\{\Phi_1, \ldots, \Phi_M\}$ span the degree-P isotropic total-degree space, so that $M = \binom{K+P}{P}$. Assume that the quadrature rule* (2.9) *corresponds to an isotropic tensor product quadrature rule with q points in each dimension, totaling $Q = q^K$ nodes. If $P < q$, then $B_{Q,m} \equiv 1$ for $m = 1 \ldots, M$.*

Quantity of interest	$B_{Q,m}$	$C_{Q,M}$	Assumptions
Mean value, $\mathcal{F} = \mathbb{E}$	1	1	The quadrature rule (2.9) exactly integrates the constant function, all weights are positive
Variance, $\mathcal{F} = \text{var}$	$1 - \delta_{m,1}$	$M - 1$	Assumptions of Lemma 4.6
L^2_ρ-norm squared, $\mathcal{F}[\cdot] = \|\cdot\|^2_{L^2_\rho}$	1	M	Assumptions of Lemma 4.6
Variance, $\mathcal{F} = \text{var}$	$\sqrt{\sum_{q=1}^Q \lvert w_q \rvert \mathbf{\Phi}_m^2(\boldsymbol{\mu}_q)}$	$\sum_{m=2}^M \sqrt{\sum_{q=1}^Q \lvert w_q \rvert \mathbf{\Phi}_m^2(\boldsymbol{\mu}_q)}$	None
L^2_ρ-norm squared, $\mathcal{F}[\cdot] = \|\cdot\|^2_{L^2_\rho}$	$\sqrt{\sum_{q=1}^Q \lvert w_q \rvert \mathbf{\Phi}_m^2(\boldsymbol{\mu}_q)}$	$\sum_{m=1}^M \sqrt{\sum_{q=1}^Q \lvert w_q \rvert \mathbf{\Phi}_m^2(\boldsymbol{\mu}_q)}$	None

Table 2: Summary of explicit values for bounding constants $B_{Q,m}$ and $C_{Q,M}$ for common quantity of interest operators $\mathcal{F}$. In the table, $\delta_{j,k}$ is the Kronecker delta function.

Proof. This is a simple consequence of the fact that a q-point Gaussian quadrature rule exactly integrates polynomials up to degree $2q - 1$. With α the size-K multi-index corresponding to the linear index m, then

$$\Phi_m^2(\boldsymbol{\mu}) = \prod_{k=1}^K \phi_{\alpha_k}^2(\mu_k), \qquad\qquad \alpha_k \leqslant |\alpha| = P.$$

Thus, in dimension k, we have $\deg \phi_{\alpha_k}^2 \leqslant 2P < 2q$. Additionally, a Gauss quadrature rule has all positive weights. Therefore, the quadrature rule integrates the polynomial in each dimension exactly, so $B_{Q,m}^2 = \sum_{j=1}^Q w_j \Phi_m^2\left(\boldsymbol{\mu}^j\right) = \mathbb{E}\Phi_m^2(\boldsymbol{\mu}) = 1$. □

Of course, similar statements about $B_{Q,m}$ can be made for non-total-degree or anisotropic spaces so long as one has a good understanding of the quadrature rule. Even if one cannot analytically derive values for $B_{Q,m}$, it is easily and inexpensively computable by applying the quadrature rule to the gPC basis Φ_m^2.

Thus under certain assumptions, the constants $C_{Q,M}$ and $B_{Q,m}$ are explicitly computable. We summarize some of these results in Table 2. However, in some cases we do not have explicit formulas for $B_{Q,m}$, but these constants can be easily computed, based on for example sparse grid code [15]. In Figures 2, we take a widely used sparse grid with two typical types of gPC bases as examples, where w_q are the weights of 4 dimensional Gauss-Patterson-based sparse grid and Φ_m are chosen as Legendre polynomials and Jacobi polynomials ($\alpha = 1, \beta = 1$) individually.

5 Numerical results

In this section, we present numerical results to illustrate the accuracy of the proposed hybrid approach and its efficiency compared to the conventional gPC method. The PDE with random inputs is the following linear elliptic equation posed on the spatial domain $D = [-1, 1] \times [-1, 1]$ with homogeneous Dirichlet boundary conditions for simplicity.

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{x}, \boldsymbol{\mu}) \nabla u(\boldsymbol{x}, \boldsymbol{\mu})) = f & \text{in} \quad D \times \mathbf{\Gamma}, \\ u(\boldsymbol{x}, \boldsymbol{\mu}) = 0 & \text{on} \quad \partial D \times \mathbf{\Gamma}. \end{cases} \tag{5.1}$$

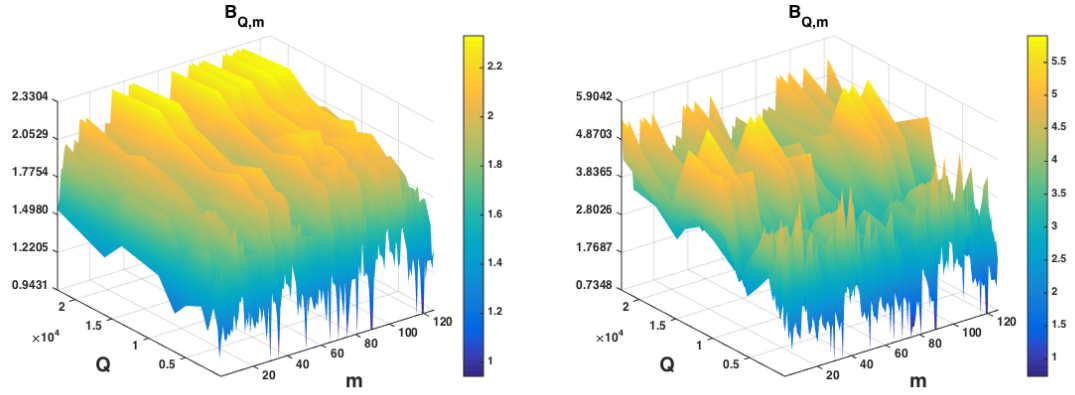

Figure 2: Value of $B_{Q,m}$ with Legendre polynomial (left) and Jacobi polynomial (right). Q varies with the accuracy level of sparse grid (level 7 to level 16). m records the sequence of gPC bases with *degree* $\leqslant 5$

The stochastic coefficient $a(\boldsymbol{x}, \boldsymbol{\mu})$ is defined as:

$$a(\boldsymbol{x}, \boldsymbol{\mu}) = A + \sum_{k=1}^{K} \frac{\cos(30 * \mu_k - 1)}{k^2} \cos(kx) \sin(ky),$$

where K is the parameter dimension and A is a positive constant ensuring $a > 0$, so that the equation is uniformly elliptic on D. We take as the right hand side $f = 1$. As we mentioned in (2.14), the output of interest $\mathcal{F}$ is defined as a certain functional of the solution over the physical domain D. Here, we explore the following two cases:

- $\mathcal{F} = \mathbb{E}$. By (2.15a), the error of the mean value we observe is

$$\xi_{\text{mean}} = \left\| \mathcal{F}[u_M^{\mathcal{N}}] - \mathcal{F}[u_M^N] \right\|_{\ell^2(D)} = \left\| \widehat{u}_1^{\mathcal{N}}(\boldsymbol{x}) - \widehat{u}_1^N(\boldsymbol{x}) \right\|_{\ell^2(D)} \tag{5.2}$$

- $\mathcal{F} = \|\cdot\|_{L^2_\rho}^2$. By (2.15c), the error of norm-squared operator we evaluate is

$$\xi_{\text{norm}} = \left\| \mathcal{F}[u_M^{\mathcal{N}}] - \mathcal{F}[u_M^N] \right\|_{\ell^2(D)} = \left\| \sum_{m=1}^{M} (\widehat{u}_m^{\mathcal{N}}(\boldsymbol{x}))^2 - \sum_{m=1}^{M} (\widehat{u}_m^N(\boldsymbol{x}))^2 \right\|_{\ell^2(D)} \tag{5.3}$$

In our numerical experiments, we test problem (5.1) with $A = 5$ and $K = 2, 4, 6$. We use the 5^{th} total degree space

$$\mathcal{U}_K^5 \equiv \text{span}\left\{ \Phi_\alpha \;\middle|\; |\alpha| \leqslant 5 \right\} \tag{5.4}$$

for gPC approximation. We adopt a tensor product quadrature rule for lower dimensional case $K = 2$, and sparse grid for the higher dimensional cases $K = 4, 6$. See Table 3 for the numbers of nodes Q and gPC polynomial bases M for each K. To obtain truth approximations $u_M^{\mathcal{N}}$, we implement a spectral collocation solver on a $\mathcal{N}_x \times \mathcal{N}_x$ ($\mathcal{N}_x = 35$) grid. The online solver of our goal-oriented reduced basis method is the least squares reduced collocation method developed in [13]. We test the algorithm for two different probability distributions for the random variable

K	2	4	6
Q(K)	1,600	22,401	367,041
M(K)	21	126	462

Table 3: The number of quadrature nodes and gPC polynomial bases for different dimension K.

$\boldsymbol{\mu}$, namely, the uniform distribution and Beta distribution with shape parameters $\alpha = 2, \beta = 2$.

We plot in Figure 3 the expected value $\mathbb{E}(u_M^{\mathcal{N}})$. The error estimates ε defined in (3.4) with $\mathcal{F} = \mathbb{E}$ and $\mathcal{F} = \|\cdot\|_{L^2_\rho}^2$ are plotted in Figure 4 and Figure 5 respectively, both displaying exponential convergence. We then evaluate the actual error in the quantity of interest of the resulting surrogate solution, as defined in (5.2) and (5.3). These are shown in Figure 6 for the two choices, $\mathbb{E}(\cdot)$ and $\|\cdot\|_{L^2_\rho}^2$. We note that both ξ_{mean} and ξ_{norm} have a clear exponential trend in convergence for both cases of probability distributions. Finally, we measure the efficiency of the hybrid algorithm by calculating the ratio of the runtime between those of the hybrid and the traditional gPC approach. This is plotted in Figure 7. Here the time for the hybrid algorithm includes *both* offline and online time. It is clear that the proposed hybrid gPC-RBM method can reach a high level of accuracy (Figure 6) while significantly alleviating the computational burden (Figure 7). Moreover, we observe that the efficiency is increasing as K gets larger. Therefore, this alleviation is more significant for high-dimensional problems, indicating great potential of the hybrid approach for even higher parametric dimensions.

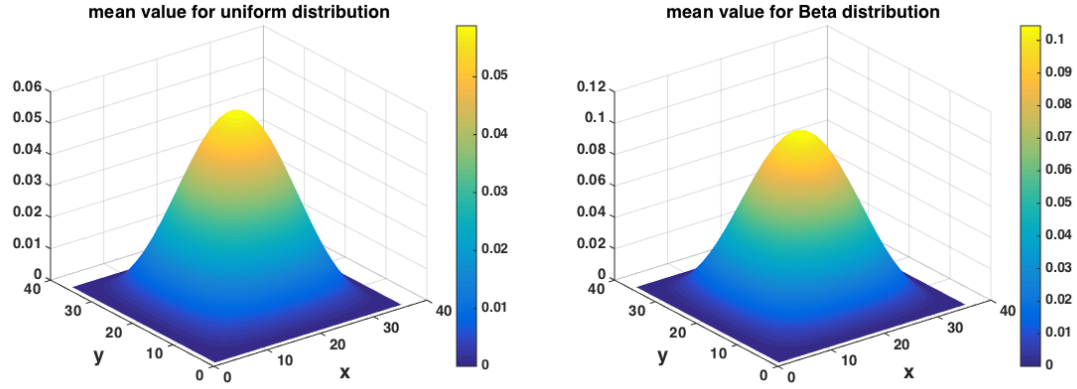

Figure 3: Expected value when $K = 4$ for uniform (left) and Beta (right) distribution.

6 Conclusion

We propose, analyze, and numerically test a hybridized RBM-gPC algorithm. It is based on a newly designed weighted RBM enabling a particular greedy algorithm tailored for any applicable quantity of interest in the context of uncertainty quantification. The final algorithm is analyzed to be reliable, and tested to be accurate. Most importantly, its efficiency is increasing with respect to the dimension of the randomness in the partial differential equation.

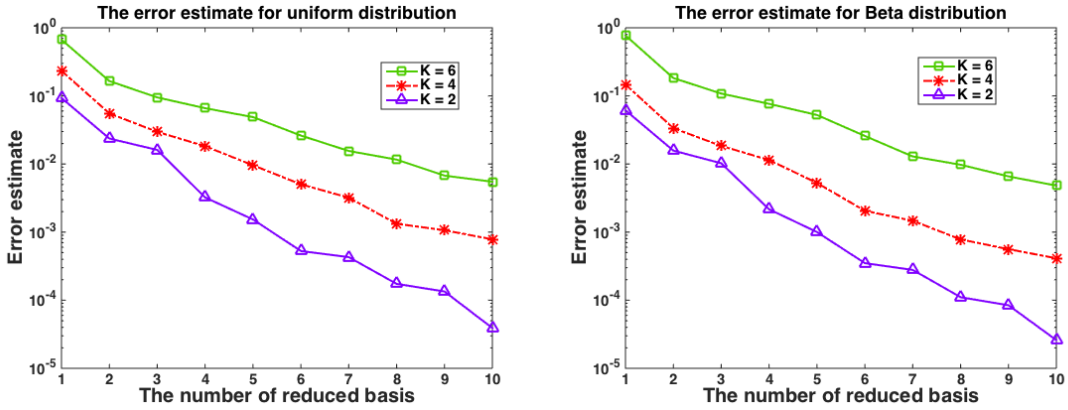

Figure 4: Error estimate ε with $\mathcal{F} = \mathbb{E}$ for uniform distribution (left) and Beta distribution (right).

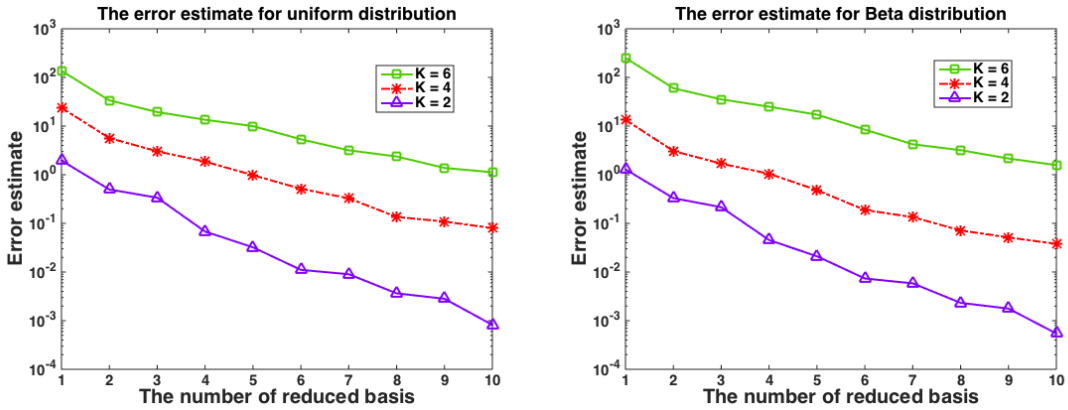

Figure 5: Error estimate ε with $\mathcal{F} = \|\cdot\|_{L^2_\rho}^2$ for uniform distribution (left) and Beta distribution (right).

References

[1] A.K.Noor and J.M.Peters, *Reduced basis technique for nonlinear analysis of structures*, AIAA **18** (1980), no. 4, 455–462.

[2] B. O. Almroth, P. Stern, and F. A. Brogan, *Automatic choice of global shape functions in structural analysis*, AIAA Journal **16** (1978), 525–528.

[3] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique **339** (2004), no. 9, 667–672.

[4] A. Barrett and G. Reddien, *On the reduced basis method*, Z. Angew. Math. Mech. **75** (1995), no. 7, 543–549. MR MR1347913 (97a:65053)

[5] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems*, SIAM Review **57** (2015), no. 4, 483–531.

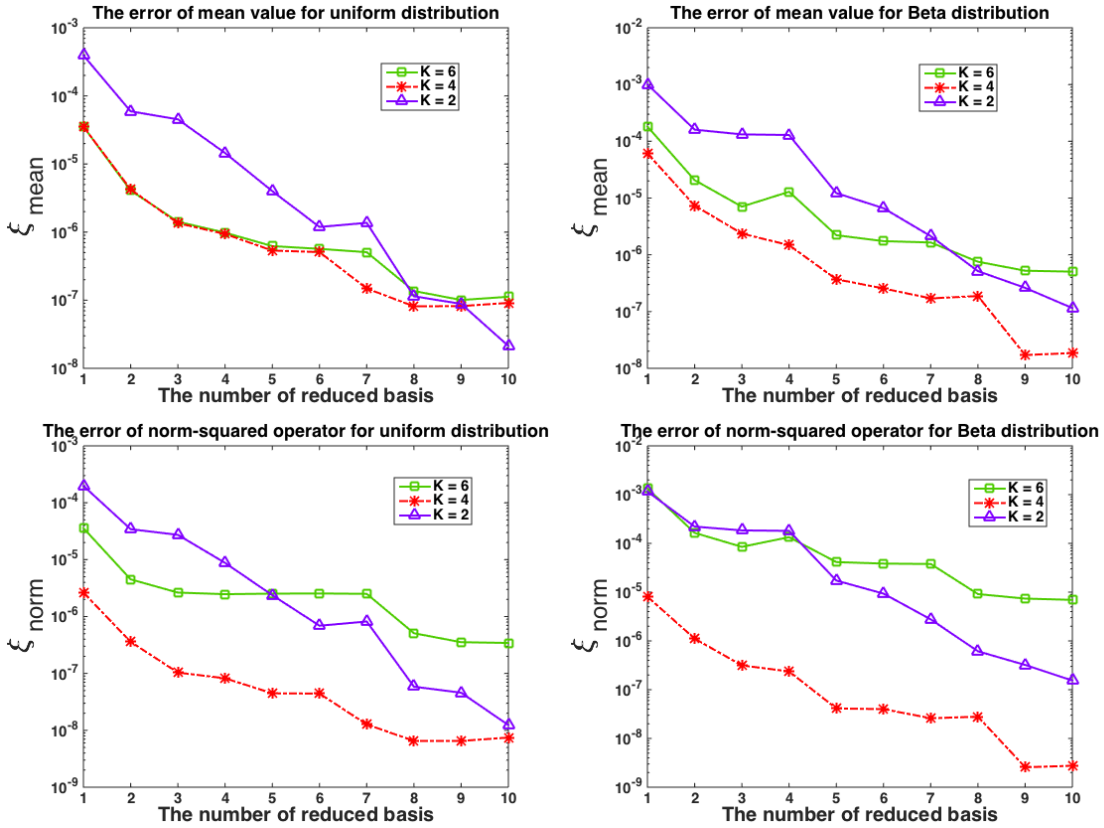

Figure 6: Error of the expected value (top) and norm-squared operator (bottom) for uniform distribution (left) and Beta distribution (right).

[6] P. Binev, A.Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM Journal of Mathematical Analysis **43** (2011), no. 3, 1457–1472.

[7] S. Boyaval, C. Le Bris, T. Lelièvre, Y. Maday, N.C. Nguyen, and A.T. Patera, *Reduced basis techniques for stochastic problems*, Archives of Computational Methods in Engineering **17** (2010), no. 4, 435–454.

[8] S. Boyaval, C. Le Bris, Y. Maday, N. C. Nguyen, and A. T. Patera, *A reduced basis approach for variational problems with stochastic parameters: application to heat conduction with variable Robin coefficient*, Comput. Methods Appl. Mech. Engrg. **198** (2009), no. 41-44, 3187–3206. MR 2571337 (2010m:65010)

[9] A. Buffa, Y. Maday, A. T. Patera, C. Prudhomme, and G. Turinici, *A priori convergence of the Greedy algorithm for the parametrized reduced basis method*, ESAIM: Mathematical Modelling and Numerical Analysis **46** (2012), no. 03, 595–603.

[10] P. Chen and A. Quarteroni, *A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods*, Journal of Computational Physics **298** (2013), 176–193.

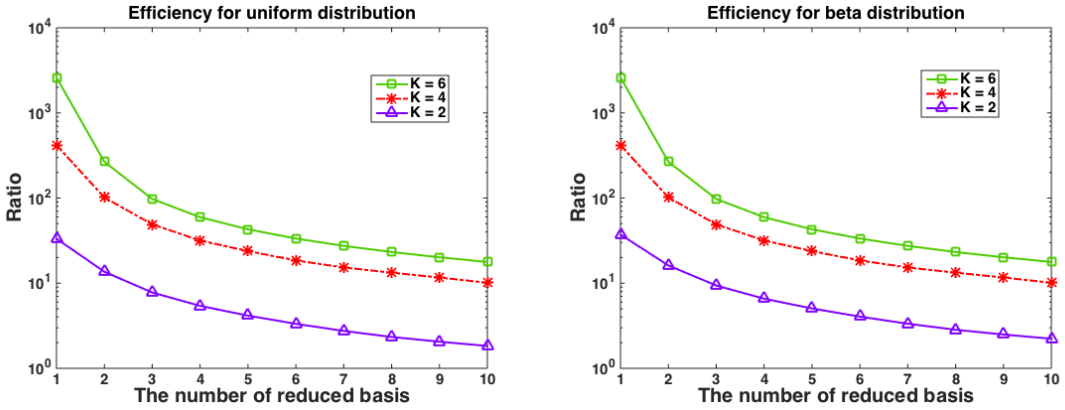

Figure 7: Efficiency of the hybrid algorithm for uniform distribution (left) and Beta distribution (right).

[11] Y. Chen, *A certified natural-norm successive constraint method for parametric inf-sup lower bounds*, Applied Numer. Math. **99** (2016), 98–108.

[12] Y. Chen, S. Gottlieb, and Y. Maday, *Parametric analytical preconditioning and its applications to the reduced collocation methods.*, C. R. Acad. Sci. Paris, Ser. I **352** (2014), 661 – 666.

[13] Y. Chen and S. Gottlieb, *Reduced Collocation Methods: Reduced Basis Methods in the Collocation Framework*, Journal of Scientific Computing **55** (2012), no. 3, 718–737 (en).

[14] H. C. Elman and Q. Liao, *Reduced basis collocation methods for partial differential equations with random coefficients*, SIAM/ASA J. Uncertainty Quantification **1** (2013), no. 1, 192–217.

[15] F. Heiss and V. Winschel, *http://www.sparse-grids.de/*

[16] D.B.P. Huynh G. Rozza and A.T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Archives of Computational Methods in Engineering **15** (2008), no. 3, 229–275.

[17] _____, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Archives of Computational Methods in Engineering **15** (2008), no. 3, 229–275.

[18] M.A. Grepl and A.T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis **39** (2005), no. 1, 157–181.

[19] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis **41** (2007), no. 03, 575–605.

[20] B. Haasdonk, K. Urban, and B. Wieland, *Reduced basis methods for parameterized partial differential equations with stochastic influences using the Karhunen-Loève expansion*, SIAM/ASA J. Uncertain. Quantif. **1** (2013), no. 1, 79–105. MR 3283879

[21] D.B.P. Huynh, D.J. Knezevic, Y. Chen, J.S. Hesthaven, and A.T. Patera, *A natural-norm successive constraint method for inf-sup lower bounds*, CMAME **199** (2010), 1963–1975.

[22] D.B.P. Huynh, G. Rozza, S. Sen, and A.T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, C. R. Acad. Sci. Paris, Série I. **345** (2007), 473 – 478.

[23] C. Johnson, *Numerical solution of partial differential equations by the finite element method*, Cambridge University Press, Cambridge, 1987. MR 925005 (89b:65003a)

[24] Y. Maday, *Reduced basis method for the rapid and reliable solution of partial differential equations*, International Congress of Mathematicians. Vol. III, Eur. Math. Soc., Zürich, 2006, pp. 1255–1270. MR 2275727 (2007m:65099)

[25] Y. Maday, A. T. Patera, and G. Turinici, *A Priori Convergence Theory for Reduced-Basis Approximations of Single-Parameter Elliptic Partial Differential Equations*, Journal of Scientific Computing **17** (2002), no. 1-4, 437–446 (en).

[26] N. C. Nguyen, A. T. Patera, and J. Peraire, *A best points interpolation method for efficient approximation of parametrized functions*, International Journal for Numerical Methods in Engineering **73** (2008), no. 4, 521–543 (en).

[27] A. Quarteroni P. Chen and G. Rozza, *A weighted reduced basis method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis **51** (2013), no. 6, 3163–3185.

[28] A.T. Patera and G. Rozza, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*, Copyright MIT (2007), 229–275.

[29] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations*, Archives of Computational Methods in Engineering **15** (2008), no. 3, 229–275 (en).

[30] D. Xiu, *Fast Numerical Methods for Stochastic Computations: A Review*, Communications in Computational Physics **5** (2009), no. 2-4, 242–272.

[31] D. Xiu and J. S. Hesthaven, *High-Order Collocation Methods for Differential Equations with Random Inputs*, SIAM Journal on Scientific Computing **27** (2005), no. 3, 1118–1139.

[32] D. Xiu and G. E. Karniadakis, *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal on Scientific Computing **24** (2002), no. 2, 619–644.