East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2022-244.241022

Vol. **13**, No. 2, pp. 398-419
May 2023

An Adaptive Non-Intrusive Multi-Fidelity Reduced Basis Method for Parameterized Partial Differential Equations

Yuanhong Chen, Xiang Sun, Yifan Lin and Zhen Gao*

School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China.

Received 23 August 2022; Accepted (in revised version) 24 October 2022.

Abstract. An adaptive non-intrusive multi-fidelity reduced basis method for parameterized partial differential equations is developed. Based on snapshots with different fidelity, the method reduces the number of high-fidelity snapshots in the regression model training and improves the accuracy of reduced-order model. One can employ the reduced-order model built on the low-fidelity data to adaptively identify the important parameter values for the high-fidelity evaluations under a given tolerance. The multi-fidelity reduced basis is constructed based on the high-fidelity snapshot matrix and the singular value decomposition of the low-fidelity snapshot matrix. Coefficients of such multi-fidelity reduced basis are determined by projecting low-fidelity snapshots on the low-fidelity reduced basis and using the Gaussian process regression. The projection method is more accurate than the regression method, but it requires low-fidelity snapshots. The regression method trains the Gaussian process regression only once but with slightly lower accuracy. Numerical tests show that the proposed multi-fidelity method can improve the accuracy and efficiency of reduced-order models.

AMS subject classifications: 65M10, 78A48

Key words: Multi-fidelity method, non-intrusive, reduced-order model, Gaussian process regression, adaptive sampling.

1. Introduction

Mathematical models described by parameterized partial differential equations (PDEs) attract substantial attention as substitutes for physical experiments in many scientific and engineering applications. Meanwhile, there are applications — e.g. uncertainty quantification and optimization design, which involve numerous evaluations of PDEs with different parameter values. However, the direct solution of large-scale systems is challenging work because of high computational costs and storage limitations. Therefore, constructing

*Corresponding author. *Email addresses:* chenyuanhong@stu.ouc.edu.cn (Y. Chen), sunxiang@ouc.edu.cn (X. Sun), linyifan@stu.ouc.edu.cn (Y. Lin), zhengao@ouc.edu.cn (Z. Gao)

reduced-order models (ROMs) which provide an accurate and efficient approximation of the corresponding full-order models has become an important task — cf. [20,30,38,39].

Here we focus on a very popular method in reduced-order modeling — viz. the reduced basis method (RBM) [4,6,9,11,45,50]. Generally, RBM can be split into offline and online stages. In the offline stage, reduced basis functions are extracted from given high-fidelity snapshots by various reduction tools such as the proper orthogonal decomposition (POD) [3, 28]. In the online stage, the reduced order solution for a new parameter value can be restored by a linear combination of reduced bases. Therefore, an ideal selection of the reduced basis must satisfy the condition that the space expanded by the reduced basis is an accurate approximation to the solution space. Once a reduced basis is constructed, a classical intrusive approach — e.g. the Galerkin projection method, is often employed to determine the reduced coefficients [10, 22, 25, 41]. However, the intrusive methods are equation-dependent and the use of source codes is usually prohibited. Besides, the ROM approximation of nonlinear problems is not always stable [27]. Therefore, non-intrusive methods have been developed in different research areas [2, 13, 17, 18, 21, 35, 47].

The non-intrusive methods are equation-free data-driven approaches such that the governing equations are treated as black boxes. In these methods, the reduced coefficients are computed by interpolation or regression methods. In particular, Gaussian process regression (GPR) [15,17,18,47] and artificial neural networks [13,21] are ones of the most used regression methods, which perform well in reduced-order modeling. In these works, the mapping from the parameter space to the reduced coefficients is built by using high-fidelity data alone. However, in order to obtain acceptable results such an approach requires a large amount of high-fidelity data and requires heavy computational time. To mitigate this issue, many multi-fidelity models have been developed [23,24,29,31–33,37,40,49,51,53]. They are based on the fact that low-fidelity models are less accurate, but contain large-scale structures of the system. Therefore, one can reduce the requirement of high-fidelity data by using multi-fidelity data in reduced-order modeling. Thus, Kast *et al.* [23] proposed a multi-fidelity ROM based on a multi-fidelity GPR [5] with inputs from different levels of accuracy. The low-fidelity data are assimilated via an interpolation approach inspired by bi-fidelity reconstruction [33], which is a linear combination of high-fidelity snapshots. Lu and Zhu [29] presented a bi-fidelity data-assisted neural network in reduced-order modeling. The method generated the high-fidelity reduced basis using POD and learnt the high-fidelity reduced coefficients using a shallow multi-layer perception by incorporating the features extracted from the low-fidelity data as the input features. A multi-fidelity approach applied to physics-informed neural networks was proposed in [37].

We note that the finding of appropriate parameters is the key step in the construction of multi-fidelity ROMs since it can enhance the accuracy and efficiency. However, the traditional sampling methods of reduced-order modeling require a large number of sample points, especially in the case of high-dimensional parameter spaces. Therefore, it is important to establish a sampling method that can identify the important parameter values. The greedy sampling methods were introduced in [8, 14, 16, 19, 44, 44], where important parameter values have been sequentially selected from the parameters enabling the largest error of the ROM. Xiao *et al.* [46] used the sparse grid to generate important samples.

In the adaptive POD model [52], the samples were selected adaptively with the goal of minimizing error in the concerned area.

The aim of this paper is to develop an adaptive multi-fidelity reduced basis model using a certain amount of low-fidelity data coupled with a few high-fidelity data. Different from the bi-fidelity reconstruction in [33], we do not directly select high-fidelity snapshots as the bases. Instead, we minimize the error indicator computed by low-fidelity ROM to adaptively select relatively important parameter values and construct a multi-fidelity reduced basis based on POD. In order to ensure that the low-fidelity reduced basis represent all low-fidelity snapshots as accurately as possible, an iterative sampling method is first proposed to identify the important parameter values. High-fidelity snapshots are then generated at the selected important parameter values, and a multi-fidelity reduced-basis is obtained by coupling the low-fidelity ROM and the high-fidelity snapshots. The corresponding combination coefficients are the same as those computed by the projection of the low-fidelity snapshot onto the space of the low-fidelity reduced-basis. This allows us to compute them efficiently due to the orthogonality of low-fidelity reduced-basis. At the online stage, we provide two strategies to recover the combination coefficients. On the one hand, we can compute them just by projecting the low-fidelity data at a given new parameter onto the low-fidelity reduced-basis. On the other hand, one can use a GPR model which is trained on the data from the low-fidelity snapshots to approximate them for a given new parameter. Although the projection method is more accurate than the regression method, it requires low-fidelity snapshots. The regression method only needs to train the GPR only once, but with slightly lower accuracy. The numerical results demonstrate that the proposed multi-fidelity method can provide an accurate and efficient ROM by using an amount of low-fidelity data coupled with a few relatively important high-fidelity data. The reduced-order solution can be recovered by the mentioned multi-fidelity reduced-basis model with acceptable accuracy. Moreover, we demonstrate that the method can be effective in alleviating the demand for abundant high-fidelity snapshots.

This paper is organized as follows. In Section 2, we introduce the problem. A non-intrusive RBM combining the POD and GPR is described in Section 3. In Section 4, we study an adaptive multi-fidelity reduced basis method and then compare it with the single-fidelity regression model in Section 5. Finally, we draw conclusions in Section 6.

2. Problem Description

The parameterized PDEs considered in this work has the following form:

$$\frac{\partial Q(\boldsymbol{x},t;\boldsymbol{\mu})}{\partial t}+\mathscr{N}\big[Q(\boldsymbol{x},t;\boldsymbol{\mu})\big]+\mathscr{L}\big[Q(\boldsymbol{x},t;\boldsymbol{\mu})\big]=0,\quad (\boldsymbol{x},t,\boldsymbol{\mu})\in\Omega\times[0,T]\times\mathscr{P},\qquad (2.1)$$

where $Q(\boldsymbol{x},t;\boldsymbol{\mu})$ is the system solution, $\Omega\subset\mathbb{R}^d$ is the spatial space, $\boldsymbol{\mu}$ denotes the parameter vector from the parameter space $\mathscr{P}\subset\mathbb{R}^m$, and $\mathscr{N}$ and $\mathscr{L}$ are nonlinear and linear differential operators. Note that for steady-state problems, the Eq. (2.1) does not depend on the time.

We are interested in developing a non-intrusive ROM for the system (2.1), which would use a certain amount of low-fidelity data coupled with high-fidelity data. The fidelity of the model for the system represented by this equation is defined by a discretization scheme in spatial domain. Generally, for a fixed numerical approach, a fine discretization scheme represents a high-fidelity model, while a coarse discretization scheme denotes a low-fidelity model. Assume that $\boldsymbol{Q}_h(t,\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and $\boldsymbol{Q}_l(t,\boldsymbol{\mu}) \in \mathbb{R}^{N_l}$ respectively represent the high- and low-fidelity solutions of the Eq. (2.1). It is also assumed that $N_l \ll N_h$ and the number of high-fidelity simulations needed in the reduced-order model is as small as possible. To this end, we have to combine the information extracted from the low-fidelity model and selected high-fidelity data in the reduced-order modeling. The detailed procedure of this approach is described below.

Remark 2.1. In the cases of time-dependent problems, the time t is taken as an additional parameter. For the sake of simplicity, we still use $\boldsymbol{\mu}$ to denote the combination of parameters and time. The discrete solution of the system is denoted by $\boldsymbol{Q}_h(\boldsymbol{\mu})$ or $\boldsymbol{Q}_l(\boldsymbol{\mu})$, where $\boldsymbol{\mu}$ belongs to $\mathscr{P}$ for steady-state problems and to $[0,T]\times\mathscr{P}$ for time-dependent problems.

3. A Data-Driven RBM

To set the stage for the later discussion, we briefly review a data-driven RBM based on the POD and GPR methods, where POD is used to extract the reduced basis from a set of high-fidelity snapshots, and GPR is employed to approximate the corresponding reduced coefficients. With this method, one can obtain the following ROM:

$$\boldsymbol{Q}_h(\boldsymbol{\mu}) \approx \sum_{i=1}^{n} \alpha_i(\boldsymbol{\mu})\,\boldsymbol{\psi}_i = \mathbf{V}_h \boldsymbol{\alpha}_h(\boldsymbol{\mu}),$$

where $\boldsymbol{\psi}_i \in R^{N_h}$ is the reduced basis vector, $\mathbf{V}_h$ is a matrix with the reduced bases as columns, and $\boldsymbol{\alpha}_h(\boldsymbol{\mu}) = [\alpha_1(\boldsymbol{\mu}), \alpha_2(\boldsymbol{\mu}), \ldots, \alpha_n(\boldsymbol{\mu})]^{\mathrm{T}}$ denotes the coefficient vector that depends on $\boldsymbol{\mu}$.

3.1. Proper orthogonal decomposition

In this section, we briefly introduce the POD for the reduced basis extraction in a data-driven RBM.

Suppose we have a parameter set $\mathscr{D}_s = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_{N_s}\}$ and its corresponding high-fidelity snapshots. The high-fidelity snapshot matrix is defined by

$$\mathbf{S}_h = [\boldsymbol{Q}_h(\boldsymbol{\mu}_1) \,|\, \boldsymbol{Q}_h(\boldsymbol{\mu}_2) \,|\, \ldots \,|\, \boldsymbol{Q}_h(\boldsymbol{\mu}_{N_s})] \in \mathbb{R}^{N_h \times N_s}.$$

The goal of the POD is to find a matrix $\mathbf{V}_h \in R^{N_h \times n}$ such that

$$\left\|\mathbf{S}_h - \mathbf{V}_h\mathbf{V}_h^{\mathrm{T}}\mathbf{S}_h\right\|_F^2 = \min_{\mathbf{W}\in\mathbf{Y}_n} \left\|\mathbf{S}_h - \mathbf{W}_h\mathbf{W}_h^{\mathrm{T}}\mathbf{S}_h\right\|_F^2,$$

where $\mathbf{Y}_n = \{\mathbf{W} \in \mathbb{R}^{N_h \times n} : \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}_n\}$, and $\|\cdot\|_F$ is the Frobenius norm. The minimization problem is usually reduced to an eigenvalue problem solved by the singular value decomposition of $\mathbf{S}_h$ — i.e.

$$\mathbf{S}_h = \mathbf{U}_h \mathbf{\Sigma}_h \mathbf{Z}_h^{\mathrm{T}},$$

where $\mathbf{U}_h \in \mathbb{R}^{N_h \times r}$, $\mathbf{Z}_h \in \mathbb{R}^{N_s \times r}$ are orthogonal matrices, r is the rank of $\mathbf{S}_h$, and the matrix $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ contains the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. It can be proved that the solution of the above minimization problem is $\mathbf{V}_h = \mathbf{U}_h[:, 1:n]$, and the following error estimate holds:

$$\left\|\mathbf{S}_h - \mathbf{V}_h \mathbf{V}_h^{\mathrm{T}} \mathbf{S}_h\right\|_F^2 = \sum_{i=n+1}^{r} \sigma_i^2.$$

Assume that the singular values decay rapidly. Then the number of reduced basis n can then be chosen according to the condition

$$\sum_{i=1}^{n} \sigma_i^2 \Big/ \sum_{i=1}^{r} \sigma_i^2 \geq 1 - \delta \tag{3.1}$$

with a specified $\delta > 0$. In addition, the reduced coefficient $\boldsymbol{\alpha}_h(\boldsymbol{\mu}_i)$ can be obtained by the projection of the high-fidelity solution $\boldsymbol{Q}_h(\boldsymbol{\mu}_i)$ on the reduced basis — i.e.

$$\boldsymbol{\alpha}_h(\boldsymbol{\mu}_i) = \mathbf{V}_h^{\mathrm{T}} \boldsymbol{Q}_h(\boldsymbol{\mu}_i). \tag{3.2}$$

There are many ways to recover the reduced coefficients. In this work, a GPR is adopted to directly build the mapping from the parameters to reduced coefficients. Once the regression model is established, the high-fidelity snapshot for a new parameter value can be directly predicted at the online stage at a low cost.

3.2. Gaussian process regression

A Gaussian process (GP) is a collection of random variables, any finite number of which obeys the joint Gaussian distribution [18]. In GPR, the prior distribution of regression function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ is assumed to be a GP defined by

$$f(\boldsymbol{x}) \sim \mathrm{GP}\big(0, \kappa(\boldsymbol{x}, \boldsymbol{x}')\big), \quad y = f(\boldsymbol{x}) + \varepsilon, \quad \varepsilon \sim \mathrm{GP}\big(0, \chi^2\big),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is input vector, κ is the semi-positive definite kernel, $y \in \mathbb{R}$ denotes the corresponding output, and χ is the standard deviation of the Gaussian noise term ε. Suppose there is a training set

$$D_{tr} = \big\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_k, y_k)\big\} = (\mathbf{X}, \mathbf{y}),$$

where $\mathbf{X} := [\boldsymbol{x}_1 | \boldsymbol{x}_2 | \ldots | \boldsymbol{x}_k] \in \mathbb{R}^{d \times k}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_k\}$. The prior distribution of $\mathbf{y}$ can be given as

$$\mathbf{y}|\mathbf{X} \sim \mathcal{N}(0, \mathbf{K}_y), \quad \mathbf{K}_y = \mathrm{Cov}[\mathbf{y}|\mathbf{X}] = \kappa(\mathbf{X}, \mathbf{X}) + \chi^2 \mathbf{I}_M.$$

For a new test input $\boldsymbol{x}^*$, the joint distribution of an observed value $\mathbf{y}$ and the predicted value f_* can be obtained as

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}_y & \mathbf{K}_* \\ \mathbf{K}_*^{\mathrm{T}} & \mathbf{K}_{**} \end{bmatrix}\right),$$

where $\mathbf{K}_* = \kappa(\mathbf{X}, \boldsymbol{x}_*)$, $\mathbf{K}_{**} = \kappa(\boldsymbol{x}_*, \boldsymbol{x}_*)$. Based on the prior joint density on the observations $\mathbf{y}$ and f_*, the posterior distribution of f_* has the following form:

$$\begin{aligned} &f_*|\boldsymbol{x}_*, \mathbf{X}, \mathbf{y} \sim \mathrm{GP}(m^*, C^*), \\ &m^* = \mathbf{K}_*^{\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{y}, \quad C^* = \mathbf{K}_{**} - \mathbf{K}_*^{\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{K}_*. \end{aligned}$$

The hyperparameters contained in the kernel function can be determined by the maximum likelihood method [46]. Besides, the kernel function here is the radial basis function, which is implemented by using the `Python` library [42].

After obtaining the data from a single-fidelity model, we construct a common reduced-order model by using the single-fidelity GPR-based RBM (SF-GPR). At the offline stage, an input set $\mathscr{D}_s = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{N_s}\}$ is first generated, and then the high-fidelity snapshots are computed for each $\boldsymbol{\mu}_i \in \mathscr{D}_s$. Next, the high-fidelity snapshot matrix $\mathbf{S}_h$ is assembled using these snapshots and the reduced basis is extracted by the POD of the snapshot matrix. The reduced coefficients are now computed by (3.2). Based on the training data $\{(\boldsymbol{\mu}_i, \mathbf{V}_h^{\mathrm{T}} \mathbf{Q}_h(\boldsymbol{\mu}_i))\}_{i=1}^{N_s}$, the regression function $\tilde{\boldsymbol{\alpha}}_h$ is trained using the GPR to recover the output $\boldsymbol{\alpha}_h(\boldsymbol{\mu}^*)$ at the new input point $\boldsymbol{\mu}^*$ at the online stage. Accordingly, the SF-GPR solution is

$$\tilde{\mathbf{Q}}_h(\boldsymbol{\mu}^*) = \mathbf{V}_h \tilde{\boldsymbol{\alpha}}_h(\boldsymbol{\mu}^*).$$

The offline and online stages of SF-GPR are described in detail in Algorithms 3.1 and 3.2, respectively.

Algorithm 3.1 SF-GPR (Offline Stage).

1: Sample an input set $\mathscr{D}_s = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{N_s}\}$.
2: Run the high-fidelity model for each $\boldsymbol{\mu}_i \in \mathscr{D}_s$ to form a high-fidelity snapshot matrix $\mathbf{S}_h$.
3: Perform POD for $\mathbf{S}_h$ to get a POD basis $\mathbf{V}_h$.
4: For each $\boldsymbol{\mu}_i \in \mathscr{D}_s$, compute the reduced coefficient vector $\boldsymbol{\alpha}_h(\boldsymbol{\mu}_i) = \mathbf{V}_h^{\mathrm{T}} \mathbf{Q}_h(\boldsymbol{\mu}_i)$.
5: Train a GPR model with the input $\boldsymbol{\mu}$ and output $\boldsymbol{\alpha}_h(\boldsymbol{\mu})$.

Algorithm 3.2 SF-GPR (Online Stage).

1: Evaluate the trained mapping at the new input point $\boldsymbol{\mu}^*$ to predict the coefficient $\tilde{\boldsymbol{\alpha}}_h(\boldsymbol{\mu}^*)$.
2: Compute the approximate solution for given $\boldsymbol{\mu}^*$: $\tilde{\mathbf{Q}}_h(\boldsymbol{\mu}^*) = \mathbf{V}_h \tilde{\boldsymbol{\alpha}}_h(\boldsymbol{\mu}^*)$.

4. Adaptive Multi-Fidelity RBM

As was already shown, given a set of high-fidelity snapshots one can build a ROM approximating the original complex model. However, in order to obtain a good ROM, a large

number of high-fidelity snapshots are needed. This poses challenges in both the computation of high-fidelity snapshots and data storage. To address this issue, we first consider an adaptive sampling method based on a low-fidelity model. After that, a multi-fidelity model combining the low-fidelity model and the high-fidelity data at the parameter values obtained by the adaptive sampling method is constructed.

4.1. Adaptive sampling strategy

In this section, we introduce an adaptive sampling strategy that utilizes a low-fidelity model to select the important parameter values for the problems considered. Though a low-fidelity model is considered, it still contains the main characteristics of the problem. Therefore, the adaptive sampling based on a low-fidelity model is feasible.

The overall procedure for the proposed adaptive sampling is presented in Algorithm 4.1. First, an initial parameter set $\mathscr{D}_s$ for the construction of POD basis is generated. Note that the initial samples are generally selected at the boundary of the parameter space. Then, a candidate set $\mathscr{D}_{te}$ containing M candidated sample points is sampled, and for each $\boldsymbol{\mu} \in \mathscr{D}_s$, the corresponding low-fidelity snapshot is calculated to form a low-fidelity snapshot matrix $\mathbf{S}_l$. After that, the low-fidelity basis $\mathbf{V}_l$ is obtained by the POD of $\mathbf{S}_l$.

Define the generalized error ϵ_m of the low-fidelity ROM at each parameter $\boldsymbol{\mu}_m \in \mathscr{D}_{te}$ as

$$\epsilon(\boldsymbol{\mu}_m) = \left\| \mathbf{Q}_l(\boldsymbol{\mu}_m) - \mathbf{V}_l \mathbf{V}_l^{\mathrm{T}} \mathbf{Q}_l(\boldsymbol{\mu}_m) \right\|_2 .$$

The important point $\boldsymbol{\mu}^*$ is selected as the one that satisfies $\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \mathscr{D}_{te}} \varepsilon(\boldsymbol{\mu})$. Repeatedly updating $\mathscr{D}_s$, $\mathbf{S}_l$ with $\boldsymbol{\mu}^*$ and $\mathbf{Q}_l(\boldsymbol{\mu}^*)$ until the maximum error $\max_{\boldsymbol{\mu} \in D_{te}} \varepsilon(\boldsymbol{\mu}) \leq \varepsilon^{tol}$, we finally obtain the set of important parameter values.

Algorithm 4.1 Adaptive sampling.

1: Initialize the set $\mathscr{D}_s = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_p\}$ and sample a candidate set $\mathscr{D}_{te} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M\}$.
2: Compute the low fidelity snapshot $\mathbf{Q}_l(\boldsymbol{\mu}_i)$ for each $\boldsymbol{\mu}_i \in \mathscr{D}_s$, and assemble the low-fidelity snapshot matrix $\mathbf{S}_l$.
3: Perform POD for $\mathbf{S}_l$ to get the low-fidelity basis $\mathbf{V}_l$.
4: **while** $\varepsilon_{\max} > \varepsilon^{tol}$ **do**
5: $\quad \mathscr{D}_s = \mathscr{D}_s \bigcup \{\boldsymbol{\mu}^*\}$;
6: $\quad \mathbf{S}_l = [\mathbf{S}_l \mid \mathbf{Q}_l(\boldsymbol{\mu}^*)]$;
7: $\quad$ Perform POD for $\mathbf{S}_l$ to get the low-fidelity basis $\mathbf{V}_l$;
8: $\quad$ **for** $m = 1$ to M **do**
9: $\quad\quad \varepsilon(\boldsymbol{\mu}_m) = \left\| \mathbf{Q}_l(\boldsymbol{\mu}_m) - \mathbf{V}_l \mathbf{V}_l^{\mathrm{T}} \mathbf{Q}_l(\boldsymbol{\mu}_m) \right\|_2$
10: $\quad$ Find $\boldsymbol{\mu}^*$ satisfying $\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \mathscr{D}_{te}} \varepsilon(\boldsymbol{\mu})$, and set $\varepsilon_{\max} = \varepsilon(\boldsymbol{\mu}^*)$.
11: Compute the high-fidelity snapshot for each $\boldsymbol{\mu}_i \in \mathscr{D}_s$, and assemble the high-fidelity snapshot matrix $\mathbf{S}_h$.

4.2. Multi-fidelity RBM

4.2.1. Multi-fidelity surrogate model

The main idea of the multi-fidelity surrogate model is to build a surrogate model by combining low-fidelity model and high-fidelity model to reduce the requirement of high-fidelity samples by compensating for expensive high-fidelity samples with cheap low-fidelity samples. Assume that a parameter set $\mathscr{D}_s$, low-fidelity snapshot matrix $\mathbf{S}_l$ and high-fidelity matrix $\mathbf{S}_h$ have been obtained by Algorithm 4.1. Finding the SVD

$$\mathbf{S}_l = \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{Z}_l^{\mathrm{T}} \tag{4.1}$$

of $\mathbf{S}_l$, we obtain $\mathbf{U}_l = \mathbf{S}_l \mathbf{Z}_l \mathbf{\Sigma}_l^{-1}$. As mentioned in Section 3.1, the low-fidelity basis $\mathbf{V}_l$ consists of the first n columns of $\mathbf{U}_l$. In other words, it can be represented as a linear combination of low-fidelity snapshots. Therefore, we can use the high-fidelity snapshot matrix to improve the accuracy of $\mathbf{U}_l$, viz.

$$\mathbf{U}_{rec} = \mathbf{S}_h \mathbf{Z}_l \mathbf{\Sigma}_l^{-1} \tag{4.2}$$

and define a multi-fidelity basis as the first n columns of $\mathbf{U}_{rec}$, i.e. $\mathbf{V}_{mf} = \mathbf{U}_{rec}[:, 1:n]$.

The corresponding multi-fidelity coefficients are computed by projecting the low-fidelity solution onto the low-fidelity reduced basis,

$$\boldsymbol{\alpha}_l(\boldsymbol{\mu}) = \mathbf{V}_l^{\mathrm{T}} \mathbf{Q}_l(\boldsymbol{\mu}). \tag{4.3}$$

The multi-fidelity surrogate model is then given by

$$\mathbf{Q}_{mf}(\boldsymbol{\mu}) = \mathbf{V}_{mf}\, \boldsymbol{\alpha}_l(\boldsymbol{\mu}). \tag{4.4}$$

4.2.2. MF-GPR

The projection method has to provide low-fidelity snapshots for every evaluation of the combination coefficients, which leads to an increase in computational time. We present an alternative strategy for the recovery of the coefficients mentioned — viz. the Gaussian process regression. Since these coefficients depend on the low-fidelity snapshots only, we can generate a large set of training data. After that, in order to avoid the repetitive computation of low-fidelity model, we can choose GPR for learning the mapping of parameters to combination coefficients.

To construct a multi-fidelity reduced-order model, we first employ Algorithm 4.1 to respectively compute the low- and high-fidelity snapshot matrixes $\mathbf{S}_l$ and $\mathbf{S}_h$ for the final selected important parameter values in $\mathscr{D}_{te}$. After that, a multi-fidelity reduced basis can be obtained by the Eqs. (4.1)-(4.2). The corresponding multi-fidelity coefficients for $\boldsymbol{\mu} \in \mathscr{D}_{te}$ are computed as $\boldsymbol{\alpha}_l(\boldsymbol{\mu}_i) = \mathbf{V}_l^{\mathrm{T}} \mathbf{Q}_l(\boldsymbol{\mu}_i)$. Let $\mathscr{P}_{tr} = \{\boldsymbol{\alpha}_l(\boldsymbol{\mu}_i) | \boldsymbol{\mu} \in \mathscr{D}_{te}\}$. A GPR model $\tilde{\boldsymbol{\alpha}}_l(\boldsymbol{\mu})$ for the multi-fidelity reduced basis can be trained by using the dataset $\{\mathscr{D}_{te}, \mathscr{P}_{tr}\}$. Thus, for a new input point $\boldsymbol{\mu}^*$, the multi-fidelity coefficients are $\tilde{\boldsymbol{\alpha}}_l(\boldsymbol{\mu}^*)$. Then the solution approximated by the multi-fidelity reduced-order model is given by

$$\tilde{\mathbf{Q}}_{mf}(\boldsymbol{\mu}^*) = \mathbf{V}_{mf}\, \tilde{\boldsymbol{\alpha}}_l(\boldsymbol{\mu}^*). \tag{4.5}$$

The detailed procedures at offline- and online-stage are presented in Algorithms 4.2 and 4.3, respectively.

Algorithm 4.2 MF-GPR (Offline Stage).

1: Determine the SVD $\mathbf{S}_l = \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{Z}_l^{\mathrm{T}}$ of $\mathbf{S}_l$.
2: Compute the reconstruction matrix $\mathbf{U}_{rec} = \mathbf{S}_h \mathbf{Z}_l \mathbf{\Sigma}_l^{-1}$ and obtain the multi-fidelity basis $\mathbf{V}_{mf} = \mathbf{U}_{rec}[:, 1:n]$.
3: Evaluate the reduced coefficient vector $\boldsymbol{\alpha}_l(\boldsymbol{\mu}_i) = \mathbf{V}_l^{\mathrm{T}} \mathbf{Q}_l(\boldsymbol{\mu}_i)$ for each $\boldsymbol{\mu}_i \in \mathscr{D}_{te}$.
4: Train a GPR model with input $\boldsymbol{\mu}$ and output $\boldsymbol{\alpha}_l(\boldsymbol{\mu})$.

Algorithm 4.3 MF-GPR (Online Stage).

1: Evaluate the trained mapping at the new input point $\boldsymbol{\mu}^*$ to predict the coefficient $\tilde{\boldsymbol{\alpha}}_l(\boldsymbol{\mu}^*)$.
2: Compute the approximate solution for given $\boldsymbol{\mu}^*$: $\tilde{\mathbf{Q}}_{mf}(\boldsymbol{\mu}^*) = \mathbf{V}_{mf}\, \tilde{\boldsymbol{\alpha}}_l(\boldsymbol{\mu}^*)$.

4.2.3. Implementation of algorithms

The implementation of the algorithms consists of the following steps:

Step 1. Sample a candidate set $\mathscr{D}_{te} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M\}$.

Step 2. Evaluate the low-fidelity snapshot for each $\boldsymbol{\mu}_i \in \mathscr{D}_{te}$, and select the important point set $\mathscr{D}_s$.

Step 3. Evaluate the high-fidelity snapshot for each $\boldsymbol{\mu}_i \in \mathscr{D}_s$.

Step 4. Evaluate the multi-fidelity reduced basis and its corresponding coefficients by combining low- and high-fidelity snapshots at the important points in $\mathscr{D}_s$.

Step 5. Train GPR model for the multi-fidelity coefficients.

Step 6. Online prediction.

In addition, in order to obtain an online approximation to a high-fidelity snapshot, we propose the following two alternative strategies:

S1. Compute the output coefficients of the trained GPR for the new input point, and recover the approximation solution using Eq. (4.5).

S2. Compute the low-fidelity snapshot for the new input point, and recover the approximation solution using Eqs. (4.3) and (4.4).

In strategy S2, the high-fidelity solution is approximated by finding the corresponding low-fidelity solution, so that Step 5 is not needed.

In Step 2, we have to find M low-fidelity snapshots at the i-th iteration in Algorithm 4.1. Besides, it is needed $\mathcal{O}(N_l {N_{s_i}}^2)$ time to determine the singular value decomposition of a low-fidelity snapshot matrix and $\mathcal{O}(N_l n_i M)$ time to find the error estimator. Here, N_{s_i} and n_i respectively denote the number of bases and import points in the current iteration. In Step 3, we run N_s high-fidelity simulations. For strategy S1, the computational costs of finding the multi-fidelity basis and the corresponding reduced coefficient vector in Step 4 are $\mathcal{O}(N_h N_s r + N_h r)$ and $\mathcal{O}(N_l n M)$, respectively. In Step 5, the computational complexity for the training of GPR is $\mathcal{O}(M^3 n)$. In Step 6, we compute the output of GPR at a new location and need $\mathcal{O}(nN_h)$ time to restore the high-fidelity snapshot at this location. For strategy S2, the computation cost for the compution of the multi-fidelity basis in Step 4 is $\mathcal{O}(N_h N_s r + N_h r)$. In Step 6, we have to determine the low-fidelity snapshot at a new location and need $\mathcal{O}(n(N_h + N_l))$ time to restore the high-fidelity snapshot at this location. Furthermore, the number of iterations in Step 2 is referred to as n_{iter}, while T_{high} and T_{low} denote the time for the high-fidelity and low-fidelity simulations, respectively. The computational complexity of the implemented algorithm is listed in Table 1.

Table 1: Computational complexity in the implementation of the algorithms.

Step	Step 2	Step 3	Step 4	Step 5	Step 6
S1	$\sum_{i=1}^{n_{iter}} \mathcal{O}(N_l {N_{s_i}}^2 + N_l n_i M) + M T_{low}$	$N_s T_{high}$	$\mathcal{O}(N_h N_s r + N_h r + N_l n M)$	$\mathcal{O}(M^3 n)$	$\mathcal{O}(nN_h)$
S2	$\sum_{i=1}^{n_{iter}} \mathcal{O}(N_l {N_{s_i}}^2 + N_l n_i M) + M T_{low}$	$N_s T_{high}$	$\mathcal{O}(N_h N_s r + N_h r)$		$\mathcal{O}(n(N_h + N_l)) + T_{low}$

Remark 4.1. For large-scale problems, one can use the incremental SVD [7] to update the reduced basis in Algorithm 4.1. In each iteration, the SVD of a one-column bordered diagonal matrix is used instead of the low-fidelity snapshot matrix.

Remark 4.2. For steady-state PDEs, a high-fidelity solver only needs to run once to obtain a high-fidelity snapshot. However, for time-dependent PDEs, although only few time nodes are needed due to the adaptive sampling, we have to run the high-fidelity solver once and compute all the snapshots before time t_j until we obtain the snapshot at time t_j.

The key point in the multi-fidelity algorithm here is the construction of a multi-fidelity reduced basis and the multi-fidelity coefficients. Let $\mathbf{S}_{mf}$ be the reconstructed multi-fidelity solution matrix

$$\mathbf{S}_{mf} = \left[\boldsymbol{Q}_{mf}(\boldsymbol{\mu}_1)|\boldsymbol{Q}_{mf}(\boldsymbol{\mu}_2)|\ldots|\boldsymbol{Q}_{mf}(\boldsymbol{\mu}_{N_s})\right],$$

and matrix $\mathbf{B}_l$ be the first n columns of matrix $\mathbf{Z}_l \boldsymbol{\Sigma}_l^{-1}$. Recall that the multi-fidelity basis has the form $\mathbf{V}_{mf} = \mathbf{S}_h \mathbf{B}_l$, and the low-fidelity basis is $\mathbf{V}_l = \mathbf{S}_l \mathbf{B}_l$. Then $\mathbf{V}_{mf} = \mathbf{S}_h \mathbf{S}_l^{\dagger} \mathbf{V}_l$, where $\dagger$ denotes pseudo-inverse. Similar to the error estimate (3.1), we have

$$\|\mathbf{S}_h - \mathbf{S}_{mf}\|_F = \left\|\mathbf{S}_h - \mathbf{V}_{mf}\mathbf{V}_l^{\top}\mathbf{S}_l\right\|_F = \left\|\mathbf{S}_h - \mathbf{S}_h\mathbf{S}_l^{\dagger}\mathbf{V}_l\mathbf{V}_l^{\top}\mathbf{S}_l\right\|_F.$$

Therefore, the closer $\mathbf{V}_l\mathbf{V}_l^{\top}\mathbf{S}_l$ is to $\mathbf{S}_l$, the closer $\mathbf{S}_{mf}$ is to $\mathbf{S}_h$.

5. Numerical Results

We want to demonstrate the efficiency and accuracy of the proposed methods by four parameterized PDEs — viz. a one-dimensional stochastic elliptic equation, a two-dimensional advection diffusion reaction equation, a two-dimensional vorticity equation, and a two-dimensional Navier-Stokes equation. All dimensions are spatial ones. In the one-dimensional case, we show the accuracy and effectiveness of the method for the problems with a high-dimensional input parameter. In the two dimensional cases, we only consider the problems with one parameter to validate the effectiveness of the adaptive sampling strategy along with the accuracy and efficiency of the multi-fidelity method.

Note that in this section, the truncation criterion is set to $\delta = 10^{-10}$ in all the constructions of the reduced basis. Moreover, to evaluate the performance of the methods, we use the following types of errors:

1. Mean approximation error of the prediction given by SF-GPR, measured by the relative L_2 error with respect to the high-fidelity snapshots $\boldsymbol{Q}_h(\boldsymbol{\mu}_{test})$, i.e.

$$\epsilon_{SF-GPR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i}) - \tilde{\boldsymbol{Q}}_h(\boldsymbol{\mu}_{test,i})\|_2}{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i})\|_2}.$$

2. Mean approximation error of the prediction given by MF-GPR, measured by the relative L_2 error with respect to the high-fidelity snapshots $\boldsymbol{Q}_h(\boldsymbol{\mu}_{test})$, i.e.

$$\epsilon_{MF-GPR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i}) - \tilde{\boldsymbol{Q}}_{mf}(\boldsymbol{\mu}_{test,i})\|_2}{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i})\|_2}.$$

3. Mean approximation error of the multi-fidelity surrogate model

$$\epsilon_{MF-SUR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i}) - \boldsymbol{Q}_{mf}(\boldsymbol{\mu}_{test,i})\|_2}{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i})\|_2}$$

   for high-fidelity snapshots.

4. Mean relative POD projection error

$$\epsilon_{POD} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i}) - \mathbf{V}_h\mathbf{V}_h^{\mathrm{T}}\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i})\|_2}{\|\boldsymbol{Q}_h(\boldsymbol{\mu}_{test,i})\|_2}$$

   for high-fidelity snapshots.

To validate the effectiveness of the adaptive sampling method, we consider two kinds of projection error with respect to the sampling strategy. One is for the random sampling method referred to as POD-RAD, and the other is for the adaptive sampling method denoted by POD-ADA.

Example 5.1 (One-dimensional stochastic elliptic equation)**.** Consider the parameterized one-dimensional elliptic equation

$$\begin{aligned} -\big(a(x,\boldsymbol{\mu})Q_x(x)\big)_x &= 1, \quad x \in (0,1), \\ Q(0) = Q(1) &= 0, \end{aligned} \tag{5.1}$$

where the random diffusivity coefficient has the form

$$a(x,\boldsymbol{\mu}) = 1 + \frac{1}{2}\sum_{k=1}^{10} \frac{1}{k\pi}\cos(2k\pi x)\,\mu_k,$$

and $\mu_i \in [-1,1]$, $i = 1,\ldots,10$ are random parameters. The Eq. (5.1) is discretized in space by linear finite elements. According to [29], the low- and high-fidelity models are the Eq. (5.1) discretized with $N_l = 33$ and $N_h = 129$ grid points, respectively. Both models are solved by a finite element method.

In this case, the adaptive sampling is carried out with the error tolerance $\epsilon = 10^{-5}$, where $\mathscr{D}_s$ is initialized with two parameter values located at the boundary of the parameter space and $\mathscr{D}_{te}$ is filled with 3500 points from $[0,1]^{10}$ chosen by the Latin hypercube sampling. Finally, 22 important sample points are selected. The numerical results obtained by different methods are presented in Table 2 depicted as the mean relative L_2 errors tested on 180 randomly generated parameter values with different models. Comparing the projection errors of the adaptive sampling (POD-ADA) and the random sampling (POD-RAD), we observe that a more accurate ROM is built by the adaptive sampling method. This demonstrates the effectiveness of the adaptive sampling method proposed. Furthermore, for the same number of high-fidelity snapshots, MF-GPR outperforms SF-GPR. This is because the training data for the GPR in SF-GPR is too small. It also indicates the effectiveness of the proposed multi-fidelity method in that high-fidelity data can correct the low-fidelity reduce-order model and improve its accuracy. In addition, we note that the error of the MF-GPR is higher than that of the MF-SUR, so that the use of GPR to approximate the coefficients adds extra errors in the predictions.

Table 2: Example 5.1. Mean relative errors.

Model	POD-ADA	POD-RAD	MF-SUR	MF-GPR	SF-GPR
Error	1.50×10^{-5}	2.13×10^{-5}	1.99×10^{-5}	1.17×10^{-4}	5.50×10^{-3}

Example 5.2 (Two-dimensional advection diffusion reaction equation)**.** Consider the two-dimensional advection diffusion reaction equation

$$\begin{aligned} &\frac{\partial Q}{\partial t} - \nabla\cdot(\mu\nabla Q) + \mathbf{b}\cdot\nabla Q + 0.01Q = s(x,y,t;\mu), && (x,y,t) \in \Omega\times(0,2\pi), \\ &\mu\nabla Q\cdot\mathbf{n} = 0, && (x,y,t) \in \Gamma\times(0,2\pi), \\ &Q(x,y,0;\mu) = 0, && (x,y) \in \Omega, \end{aligned} \tag{5.2}$$

where

$$s(x,t;\mu)=\exp\left(-\left((x-0.5)^2+(y-0.5)^2\right)/0.07^2\right)$$

is the source term, $\Omega=[0,1]\times[0,1]$ the computational domain, $\Gamma=\partial\Omega$ the boundary of Ω, $\mathbf{n}$ the outward unit normal to $\partial\Omega$, μ a random parameter distributed on $[0.006,0.02]$, and $\mathbf{b}=(\cos(t),\sin(t))^{\mathrm{T}}$.

Similar to the above example, the Eq. (5.2) is solved by a finite element method. The low- and high-fidelity models are the Eq. (5.2) discretized with 1342 and 21472 finite elements, respectively. The simulations are carried out in MATLAB using the `redbKIT` library in [34]. In the adaptive sampling, the error tolerance is chosen as $\epsilon=5\times10^{-5}$. Besides, the important parameter set $\mathscr{D}_s$, candidate set $\mathscr{D}_{te}$ and test set are randomly sampled from the parameter space, and the test set contains no points in the candidate set. Here, $\mathscr{D}_s$ is initialized with two randomly selected parameter values at the boundary of the parameter space. Moreover, we generate 400 parameter values as the test set and 2500 parameter values as the candidate set. By the adaptive sampling strategy, 149 points are finally selected as important points among 2500 candidate points.

Fig. 1 presents the parameter values selected using the adaptive sampling method proposed and the random sampling method. Clearly, one can observe that the parameter values selected by the two sampling methods are very different, indicating the effectiveness of the adaptive sampling method. This is also verified in Table 3, where the mean relative error of the POD-ADA prediction is significantly smaller than for the POD-RAD prediction. We also note that MF-GPR with the high-fidelity snapshots evaluated at the selected parameter values, substantially outperforms the SF-GPR method. This shows the efficiency of the multi-fidelity method considered. In addition, MF-GPR and MF-SUR have different prediction errors. This is because that the GPR is the approximation of the multi-fidelity projection coefficient which is trained with the training data obtained from the low-fidelity data. The difference can be diminished by adding training data for the GPR or by using a more suitable kernel for the GPR. To further compare the performances of different methods, we

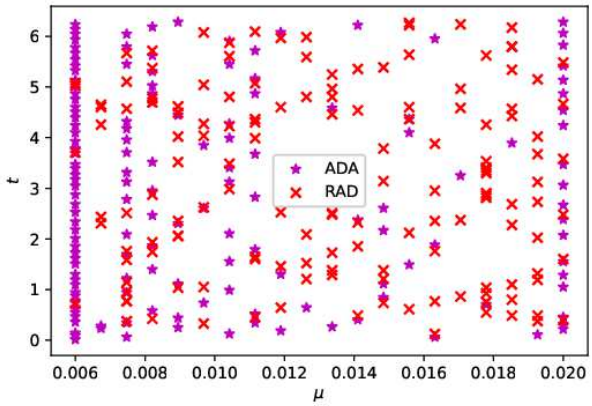

Figure 1: Example 5.2. Selected important parameter values obtained by adaptive sampling and common random sampling strategies.

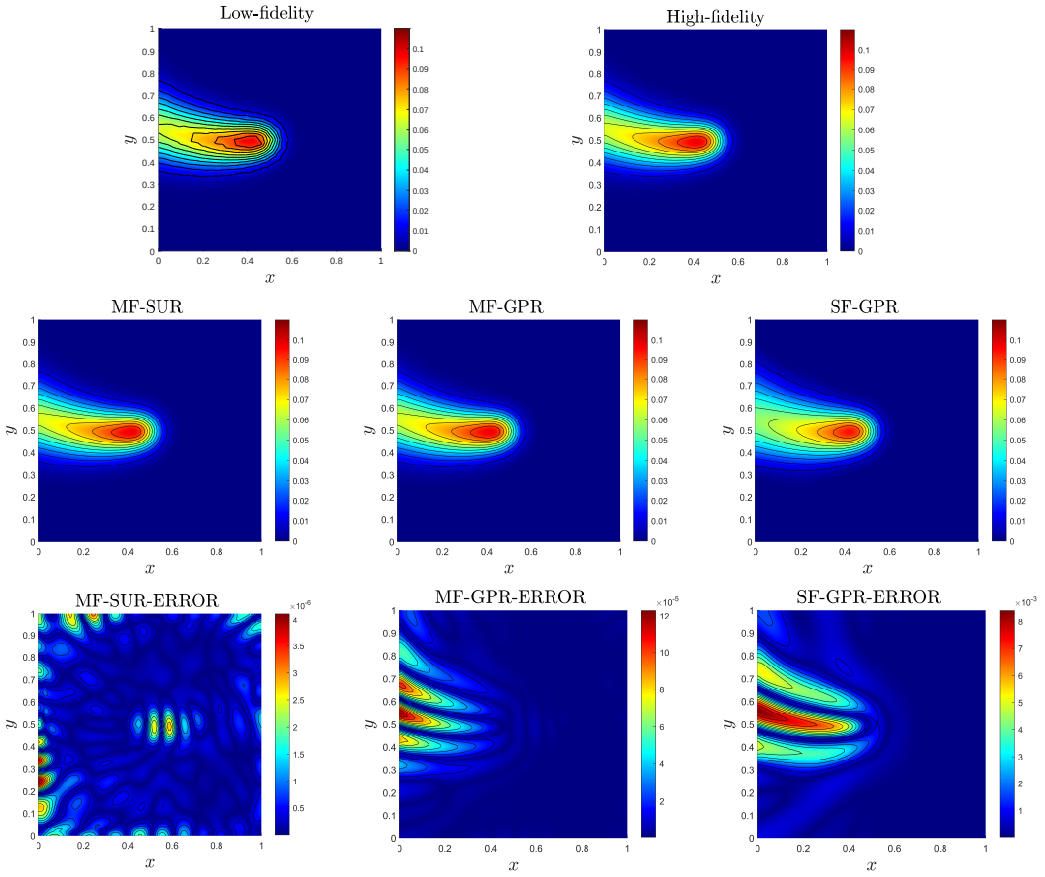

Figure 2: Example 5.2. Solutions of Eq. (5.2) evaluated by different methods (top and middle) and the corresponding absolute errors (bottom) with one randomly selected parameter value $\mu = 0.0064$ at $t = 3.2987$.

show the recovered solutions of the Eq. (5.2) and the corresponding absolute errors evaluated at a randomly sampled parameter value in Fig. 2. Note that the solutions of MF-GPR, MF-SUR and high-fidelity model are almost the same, whereas the SF-GPR trained with only the high-fidelity data at selected parameter values loses its accuracy. This indicates the effectiveness and accuracy of the multi-fidelity method used.

Table 3: Example 5.2. Mean relative errors.

Model	POD-ADA	POD-RAD	MF-SUR	MF-GPR	SF-GPR
Error	3.51×10^{-5}	1.48×10^{-3}	9.32×10^{-5}	8.90×10^{-4}	1.06×10^{-1}

Example 5.3 (Two-dimensional vorticity equation)**.** Consider the following two-dimensional vorticity equation — cf. [43],

$$\partial_t Q = \mu \Delta Q - (u \cdot \nabla) Q, \quad (x, y, t) \in [0, 2\pi] \times [0, 2\pi] \times [0, 50] \tag{5.3}$$

with smooth initial vorticity

$$Q(x,y,0) = \exp\left(-\frac{(x-\pi+\pi/5)^2+(y-\pi+\pi/5)^2}{0.3}\right) - \exp\left(\frac{-(x-\pi-\pi/5)^2+(y-\pi+\pi/5)^2}{0.2}\right) + \exp\left(\frac{-(x-\pi-\pi/5)^2+(y-\pi-\pi/5)^2}{0.4}\right)$$

for $\mu \in [0.002, 0.005]$. High-fidelity snapshots are solved using Fourier spectral method on a uniform grid of size $N_h = 128 \times 128$ in the spatial domain, while low-fidelity snapshots are solved by the same method but on a uniform grid of size $N_l = 16 \times 16$.

In this example, the tolerance for the adaptive sampling is set to $\epsilon = 1 \times 10^{-4}$. The important parameter set $\mathscr{D}_s$ is initialized with only two randomly generated parameter values at the parameter space boundary, the candidate set $\mathscr{D}_{te}$ is filled with 2500 randomly selected parameter values, and a set of 400 randomly selected parameter values which are out of $\mathscr{D}_{te}$ is taken as the test set. Finally, we select 93 important parameter values from $\mathscr{D}_{te}$ by Algorithm 4.1.

Fig. 3 demonstrates the selected parameter values obtained by adaptive and random samplings. Unlike random sampling, the parameter values chosen by adaptive sampling are more consistent with the physical properties of the vorticity. This is because the smaller the viscosity coefficient μ is, the stronger the nonlinearity of the solution becomes, and the more samples are needed to capture the nonlinearity of the equation solution.

To validate the accuracy of the methods, the mean relative errors of different methods in the predictions of Q for the test set are shown in Table 4. Similar to the previous examples, the multi-fidelity method MF-GPR substantially outperforms the single-fidelity method SF-GPR. Besides, POD-ADA method performs better than POD-RAD and the errors of MF-SUR

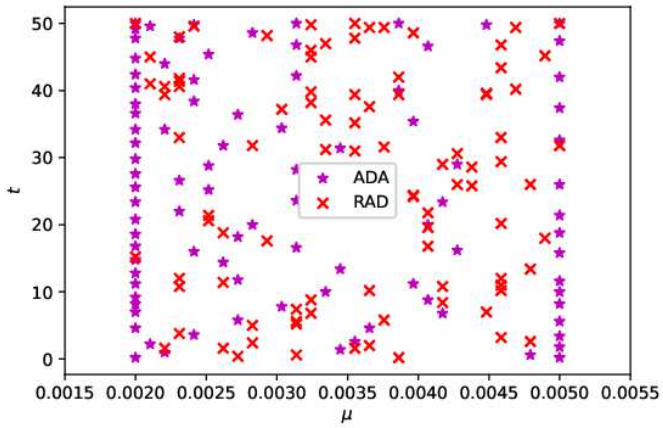

Figure 3: Example 5.3. Selected important parameter values obtained by adaptive sampling and common random sampling strategies.

Table 4: Example 5.3. Mean relative errors.

Model	POD-ADA	POD-RAD	MF-SUR	MF-GPR	SF-GPR
Error	1.06×10^{-4}	7.58×10^{-4}	5.62×10^{-4}	1.76×10^{-2}	1.01×10^{-1}

and POD-ADA are comparable. These further validate the accuracy and effectiveness of the proposed multi-fidelity method. In addition, the MF-SUR method performs better than MF-GPR. This is because of the extra error introduced by the GPR in the approximation of the multi-fidelity coefficients.

Fig. 4 shows the solutions of Eq. (5.3) obtained by different methods and the corresponding absolute errors for a randomly generated parameter value $\mu = 0.0021$ at $t = 17.4$. Note that MF-GPR captures almost all of the characteristics of the solution, although it has a larger error than MF-SUR. This implies the proposed methods can approximate the multi-fidelity coefficients accurately enough.

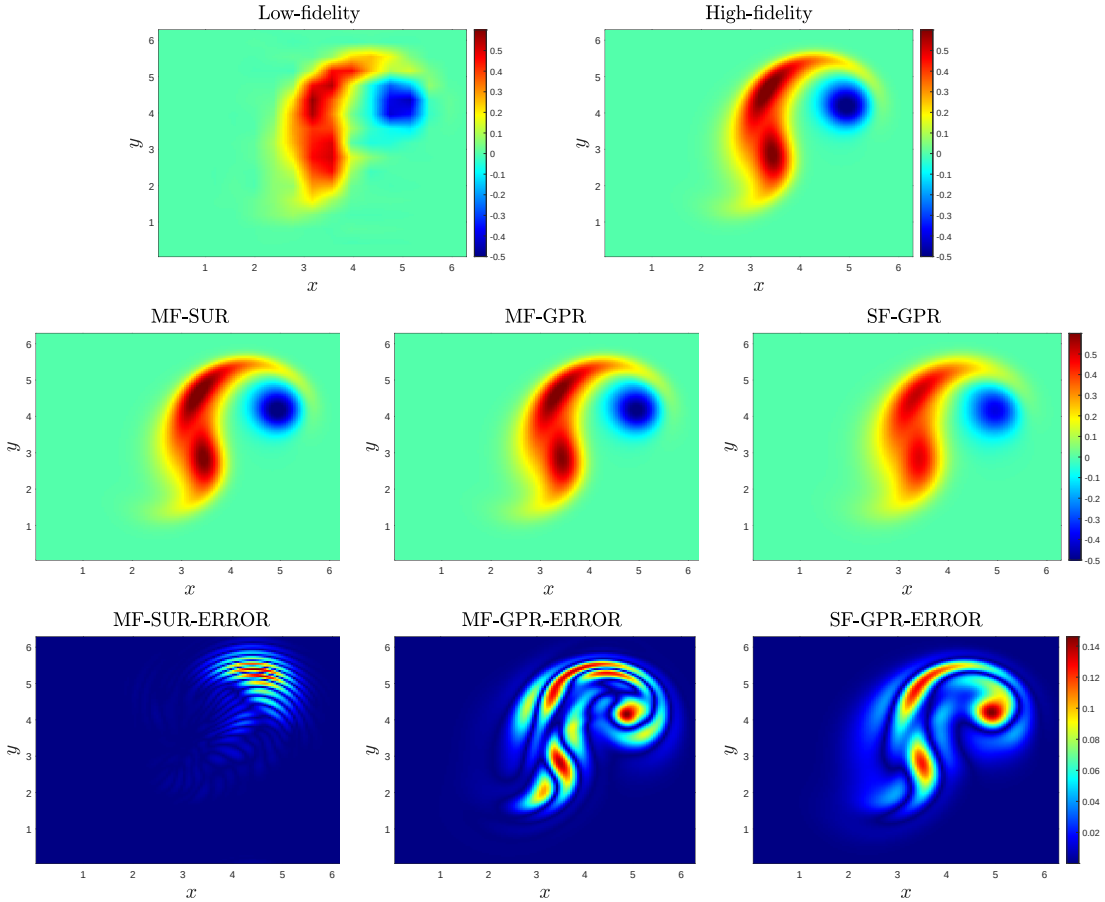

Figure 4: Example 5.3. Solutions of Eq. (5.3) obtained by different methods (top and middle) and the corresponding absolute errors (bottom) at $\mu = 0.0021$ and $t = 17.4$.

Example 5.4 (Two-dimensional Navier-Stokes Equation)**.** Consider a classical benchmark fluid problem — viz. the fluid flow around a cylinder, modeled by the following equations:

$$\begin{aligned} &\rho \mathbf{u}_t - \nu \Delta \mathbf{u} + \rho \mathbf{u} \nabla \mathbf{u} + \nabla p = \mathbf{0}, \\ &\nabla \cdot \mathbf{u} = 0, \quad t \in [0,5], \end{aligned} \tag{5.4}$$

where $\rho = 1.0$ is the fluid density, $\nu = 0.001$ the kinematic viscosity, $\mathbf{u}$ the velocity, and p the pressure. The problem is defined on the spatial domain $[0,2.2] \times [0,0.41]$ with the cylinder of the diameter 0.1 centered at $(0.2,0.2)$. The inflow boundary condition has the form

$$\mathbf{u}(0,y) = \left[\frac{4Uy(0.41-y)}{0.41^2}, 0 \right]^T,$$

where U is a random parameter uniformly distributed in $\mathscr{P} = [1.275, 1.5]$. The corresponding Reynolds number $Re = U_m L/\nu$, where $U_m = 2U/3$ and $L = 0.1$ denotes the characteristic length of the flow configuration, is therefore uniformly distributed in $[85, 100]$.

The Eq. (5.4) is solved by the finite element method [26]. The low- and high-fidelity models are the Eq. (5.4) discretized with 1207 and 19312 finite elements, respectively. To determine the error tolerance, we choose a slightly larger ϵ to balance the computational cost and accuracy in order to select less sample points when dealing with relatively complex problems. By setting the error tolerance $\epsilon = 4 \times 10^{-4}$, 171 parameter values among 2500 candidate parameter values are chosen by using Algorithm 4.1. Furthermore, 400 randomly generated samples not included in the candidate set, are employed as the test set. The numerical results are shown in Figs. 5-6 and Table 5.

Fig. 5 displays different important parameter values selected by both adaptive and random samplings. Similarly to the above considerations, the selected parameter values in the random sampling method are more uniformly distributed in the parameter space than in the adaptive sampling method. Furthermore, the parameter values selected by adaptive

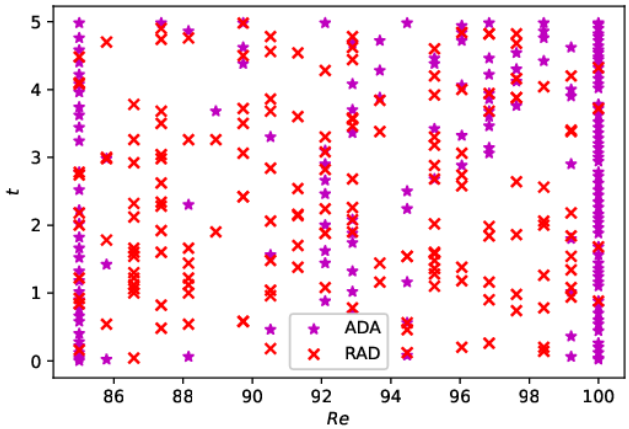

Figure 5: Example 5.4. Selected important parameter values obtained by adaptive sampling and common random sampling strategies.

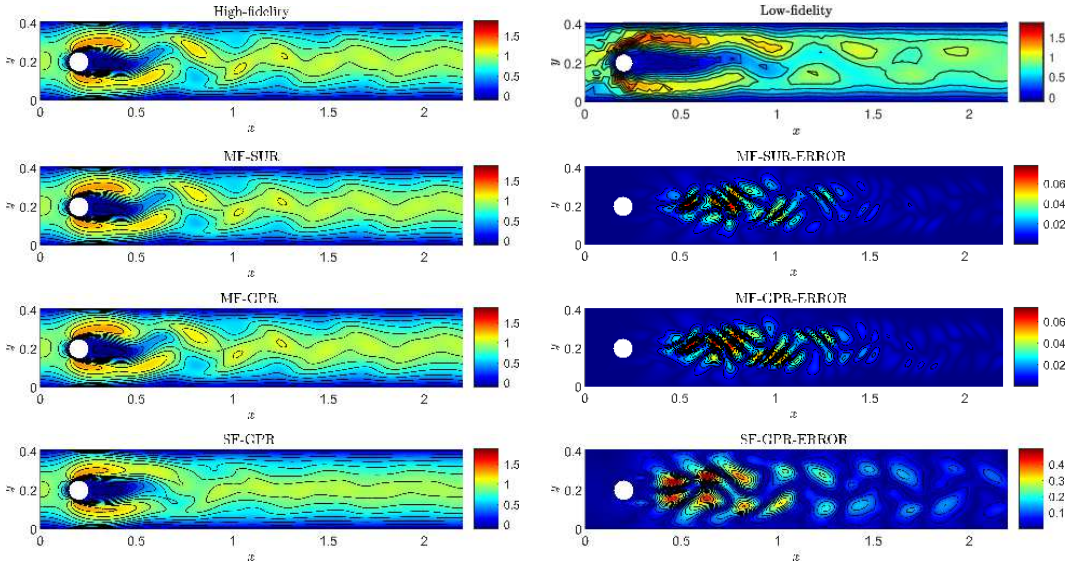

Figure 6: Example 5.4. The velocities computed using different methods and the corresponding absolute errors at $Re = 98.82$ and $t = 4.9$.

Table 5: Example 5.4. Mean relative errors.

Model	POD-ADA	POD-RAD	MF-SUR	MF-GPR	SF-GPR
Error	1.52×10^{-4}	5.85×10^{-4}	6.58×10^{-3}	6.98×10^{-3}	5.71×10^{-2}

sampling are mainly concentrated at the boundary of the parameter space, especially at the locations with large Reynolds numbers. This also demonstrates that the adaptive sampling strategy can select the most influential parameter values for the variation of the PDE solution. To study the accuracy of the proposed multi-fidelity method, we evaluate the mean relative errors of different methods in the prediction of velocity at the parameter values in the test set — cf. Table 5. Similar conclusions can be obtained as those in the previous examples, which demonstrate the effectiveness of the proposed method for the current complex two-dimension PDE equation. We also determine the velocities by using these two methods for the randomly generated parameter $Re = 98.82$ at $t = 4.9$ and show the corresponding high- and low-fidelity solutions in Fig. 6. Note that both MF-GPR and MF-SUR can capture almost all velocity features. The corresponding absolute errors obtained by different methods are shown in Fig. 6, demonstrate the accuracy and effectiveness of the adaptive multi-fidelity reduced basis method since the errors in MF-SUR and MF-GPR are much smaller than in SF-GPR.

6. Conclusion

We propose a non-intrusive RBM based on an adaptive sampling method and a multi-fidelity algorithm to deal with the heavy requirement of high-fidelity snapshots in a common

reduced-order modeling of parameterized PDEs. In particular, a set of important sample points for the high-fidelity snapshot evaluations are first selected by a low-fidelity ROM with an adaptive sampling strategy. After that, a multi-fidelity reduced basis is obtained by coupling the high-fidelity snapshot matrix with the low-fidelity ROM. The corresponding multi-fidelity reduced coefficients are approximated by using the GPR model or by the projections of low-fidelity data on the low-fidelity reduced basis. We apply these methods to four parameterized PDEs — viz. stochastic elliptic, advection diffusion reaction, vorticity and the Navier-Stokes equation. The results demonstrate the accuracy and efficiency of the methods. In addition, the adaptive sampling method under consideration can effectively identify points important to the model. Besides, for the same number of high-fidelity snapshots, our method performs much better than the single-fidelity one. Therefore, this multi-fidelity method can capture more informative features of the problems and achieve the desired accuracy with a reduced number of high-fidelity simulations. We also note that the method based on the projections is a bit more accurate than the one using the GPR model. However, extra low-fidelity snapshots are needed. A natural result is that the higher the similarity of the low- and high-fidelity models, the higher the accuracy of the multi-fidelity surrogate is. In the future, other low-fidelity models can be used to improve the accuracy of multi-fidelity model evaluation.

Acknowledgments

The authors are partially supported by the National Key Research and Development Program of China (2021YFF0704000), by the Taishan Scholars Program (tsqn202211059) and by the National Natural Science Foundation of China (12201592).

References

[1] N.M. Alexandrov, R.M. Lewis, C.R. Gumbert, L.L. Green and P.A. Newman, *Optimization with variable-fidelity models applied to wing design*, In: 38th Aerospace Sciences Meeting and Exhibit, **841**, (2000).

[2] F. Alsayyari, Z. Perkó, M. Tiberga, J.L. Kloosterman and D. Lathouwers, *A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems*, Comput. Meth. Appl. Mech. Eng. **373**, 113483 (2021).

[3] F. Ballarin, A. Manzoni, A. Quarteroni and G. Rozza, *Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations*, Int. J. Numer. Meth. Eng. **102**, 1136–1161 (2015).

[4] P. Benner, S. Gugercin and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev. **57**, 483–531 (2015).

[5] E.V. Bonilla, K.M. Chai and C. Williams, *Multi-task Gaussian process prediction*, Adv Neural Inf Process Syst. 153–160 (2008).

[6] S. Boyaval, C.L. Bris, T. Lelièvre, Y. Maday, N. Nguyen and A. Patera, *Reduced basis techniques for stochastic problems*, Arch. Comput. Methods Eng. **17**, 1–20 (2010).

[7] M. Brand, *Incremental singular value decomposition of uncertain data with missing values*, in: European Conference on Computer Vision, Springer, 707–720 (2002).

[8] T. Bui-Thanh and K. Willcox, *Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications*, AIAA J. **46**, 2520–2529 (2008).
[9] P. Chen, A. Quarteroni and G. Rozza, *Comparison between reduced basis and stochastic collocation methods for elliptic problems*, J. Sci. Comput. **59**, 187–216 (2014).
[10] M. Couplet, C. Basdevant and P. Sagaut, *Calibrated reduced-order POD-Galerkin system for fluid flow modelling*, J. Comput. Phys. **207**, 192–220 (2005).
[11] H. Elman and Q. Liao, *Reduced basis collocation methods for partial differential equations with random coefficients*, SIAM/ASA J. Uncertain. Quantificat. **1**, 192–217 (2013).
[12] L. Feng, G. Fu and Z. Wang, *A FOM/ROM Hybrid Approach for Accelerating Numerical Simulations*, J. Sci. Comput. **89**, (2021).
[13] Z. Gao, Q. Liu, J.S. Hesthaven, B.S. Wang, W.S. Don and X. Wen, *Non-intrusive reduced order modeling of convection dominated flows using artificial neural networks with application to Rayleigh-Taylor instability*, Commun. Comput. Phys. **30**, 97–123 (2021).
[14] O. Ghattas, T. Bui-Thanh and K. Willcox, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput. **30**, 3270–3288 (2008).
[15] O. Giulio, D. Nicola and R. Gianluigi, *A Gaussian Process Regression approach within a data-driven POD framework for engineering problems in fluid dynamics*, Math. Eng. **4**, 1–16 (2022).
[16] M. Grepl and A. Patera, *A posteriori error bounds for reduced basis approximations of parametrized parabolic partial differential equations*, ESAIM: Math. Model. Numer. Anal. **39**, 157–181 (2005).
[17] M. Guo and J.S. Hesthaven, *Data-driven reduced order modeling for time-dependent problems*, Comput. Meth. Appl. Mech. Eng. **345**, 75–99 (2019).
[18] M. Guo and J.S. Hesthaven, *Reduced order modeling for nonlinear structural analysis using Gaussian process regression*, Comput. Meth. Appl. Mech. Eng. **341**, 807–826 (2018).
[19] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM: Math. Model. Numer. Anal. **42**, 277–302 (2008).
[20] J.S. Hesthaven, G. Rozza and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, (2016).
[21] J.S. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks*, J. Comput. Phys. **363**, 55–78 (2018).
[22] A. Iollo, S. Lanteri and J.A. Désidéri, *Stability properties of POD-Galerkin approximations for the compressible Navier-Stokes equations*, Theor. Comput. Fluid Dyn. **13**, 377–396 (2000).
[23] M. Kast, M. Guo and J.S. Hesthaven, *A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems*, Comput. Meth. Appl. Mech. Eng. **364**, 112947 (2020).
[24] M.C. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, Biometrika, **87**, 1–13 (2000).
[25] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math. **90**, 117–148 (2001).
[26] M.G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, Springer, (2013).
[27] T. Lassila, A. Manzoni, A. Quarteroni and G. Rozza, *Model order reduction in fluid dynamics: Challenges and perspectives*, in: Reduced Order Methods for Modeling and Computational Reduction, Springer, **9**, 235–273 (2014).
[28] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee and C. Wu, *Proper orthogonal decomposition and its applications-part I: Theory*, J. Sound Vib. **252**, 527–544 (2002).
[29] C. Lu and X. Zhu, *Bifidelity Data-Assisted Neural Networks in Non-intrusive Reduced-Order Modeling*, J. Sci. Comput. **87**, 1–30 (2021).

[30] D.J. Lucia, P.S. Beran and W.A. Silva, *Reduced-order modeling: new approaches for computational physics*, Prog. Aeosp. Sci. **40**, 51–117 (2004).
[31] X. Meng and G.E. Karniadakis, *A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems*, J. Comput. Phys. **401**, 109020 (2019).
[32] M. Motamed, *A multi-fidelity neural network surrogate sampling method for uncertainty quantification*, Int. J. Uncertain Quan. **10**, 315–332 (2020).
[33] A. Narayan, C. Gittelson and D. Xiu, *A stochastic collocation algorithm with multifidelity models*, SIAM J. Sci. Comput. **36**, A495–A521 (2014).
[34] F. Negri, *redbKIT Version 2.2.*, `http://redbkit.github.io/redbKIT/`, (2016).
[35] S. Pawar, S. Rahman, H. Vaddireddy, O. San, A. Rasheed and P. Vedula, *A deep learning enabler for nonintrusive reduced order modeling of fluid flows*, Phys. Fluids. **31**, 085101 (2019).
[36] B. Peherstorfer, K. Willcox and M. Gunzburger, *Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization*, SIAM Rev. **60**, 550–591 (2018).
[37] M. Penwarden, S. Zhe, A. Narayan and R.M. Kirby, *Multifidelity Modeling for Physics-Informed Neural Networks (PINNs)*, J. Comput. Phys. **451**, 110844 (2021).
[38] G.R. Piao and H.C. Lee, *Distributed feedback control of the Benjamin-Bona-Mahony-Burgers equation by a Reduced-Order Model*, East Asian J. Appl. Math. **5**, 61–74 (2015).
[39] A. Quarteroni, A. Manzoni and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Springer, (2015).
[40] M. Raissi, P. Perdikaris and G.E. Karniadakis, *Inferring solutions of differential equations using noisy multi-fidelity data*, J. Comput. Phys. **335**, 736–746 (2017).
[41] C.W. Rowley, T. Colonius and R.M. Murray, *Model reduction for compressible flows using pod and Galerkin projection*, Physica D. **189**, 115–129 (2004).
[42] SheffieldML, *GPy – A Gaussian process framework in python*, `http://github.com/SheffieldML/GPy`, since 2012.
[43] M. Suzuki, *Fourier-spectral methods for Navier-Stokes equations in 2D*, `http://www.math.mcgill.ca/gantumur/math595f14/NSMashbat.eps`, (2014).
[44] K. Veroy and A. Patera, *Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced basis a posteriori error bounds*, Int. J. Numer. Methods Fluids. **47**, 773–788 (2005).
[45] K. Veroy, D. Rovas and A. Patera, *A posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: "Convex Inverse" bound conditioners*, ESAIM Control Optim. Calc. Var. **8**, 1007–1028 (2002).
[46] C.K. Williams and C.E. Rasmussen, *Gaussian processes for regression*, in: Advances in Neural Information Processing Systems, 514–520 (1996).
[47] C.K. Williams and C.E. Rasmussen, *Gaussian Processes for Machine Learning*, MIT Press, (2006).
[48] D. Xiao, F. Fang, C. Pain and I. Navon, *A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications*, Comput. Meth. Appl. Mech. Eng. **317**, 868–889 (2017).
[49] L. Yan and T. Zhou, *Adaptive multi-fidelity polynomial chaos approach to Bayesian inference in inverse problems*, J. Comput. Phys. **381**, 110–128 (2019).
[50] L. Yan and T. Zhou, *An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems*, Commun. Comput. Phys. **28**, 2180–2205 (2020).
[51] X. Yang, X. Zhu and J. Li, *When bifidelity meets cokriging: An efficient physics-informed multifidelity method*, SIAM J. Sci. Comput. **42**, A220–A249 (2020).
[52] J. Yu, C. Yan, Z.H. Jiang, W. Yuan and S.S. Chen, *Adaptive non-intrusive reduced order modeling*

for compressible flows, J. Comput. Phys. **397**, 108855 (2019).
[53] X. Zhu, E.M. Linebarger and D. Xiu, *Multi-fidelity stochastic collocation method for computation of statistical moments*, J. Comput. Phys. **341**, 386–396 (2017).