Journal of Computational Physics 309 (2016) 52–68

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Gaussian functional regression for output prediction: Model assimilation and experimental design

N.C. Nguyen *, J. Peraire

M.I.T., Department of Aeronautics and Astronautics, 77 Massachusetts Ave., Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 16 May 2015
Received in revised form 22 October 2015
Accepted 16 December 2015
Available online 29 December 2015

Keywords:
Gaussian functional regression
Multi-fidelity models
Reduced basis method
Model reduction
Experimental design

ABSTRACT

In this paper, we introduce a Gaussian functional regression (GFR) technique that integrates multi-fidelity models with model reduction to efficiently predict the input–output relationship of a high-fidelity model. The GFR method combines the high-fidelity model with a low-fidelity model to provide an estimate of the output of the high-fidelity model in the form of a posterior distribution that can characterize uncertainty in the prediction. A reduced basis approximation is constructed upon the low-fidelity model and incorporated into the GFR method to yield an inexpensive posterior distribution of the output estimate. As this posterior distribution depends crucially on a set of training inputs at which the high-fidelity models are simulated, we develop a greedy sampling algorithm to select the training inputs. Our approach results in an output prediction model that inherits the fidelity of the high-fidelity model and has the computational complexity of the reduced basis approximation. Numerical results are presented to demonstrate the proposed approach.

1. Introduction

Engineering analysis and design often require the prediction of outputs (quantities of interest) of a physical system for many different inputs. Typical outputs include maximum (or average) energies and forces, critical stresses or strains, flowrates or pressure drops, lifts, drags, and various local and global measures of concentration, temperature, and flux. The input typically characterizes physical properties and material, geometry, boundary conditions, and force fields or sources. The output is a function of the input that serves to identify a particular realization or configuration of the physical system. The input–output relationship thus encapsulates the behavior of the physical system relevant to the desired engineering context. More often that not, there exist several mathematical models with different fidelity and complexity for predicting the input–output relationship of the physical system. Typically, model complexity increases as a function of model fidelity. In other words, high-fidelity models are more accurate but more expensive than low-fidelity models. Therefore, it is desired in many applications to make use of all the available models for efficient prediction of the input–output relationship.

In this paper, we extend the Gaussian functional regression (GFR) framework first introduced in [14,17] to develop a *model assimilation* technique for parametrized mathematical models. We consider a set of high-fidelity (HF) model, low-fidelity (LF) model, and reduced order model. All the models we address in this paper are deterministic and steady-state partial differential equations. It is assumed that the HF model adequately captures the input–output relationship of the

* Corresponding author.
E-mail addresses: cuongng@mit.edu (N.C. Nguyen), peraire@mit.edu (J. Peraire).

http://dx.doi.org/10.1016/j.jcp.2015.12.035
0021-9991/

physical system in consideration. However, it can be very expensive to run the HF model many times because of its high computational cost. The LF model is obtained through an appropriate simplification of the HF model either by ignoring the nonlinear physics in the HF model or by linearizing the HF model around a fixed state. Although the LF model is less expensive than the HF model, it contains uncertainty because of the simplification made. In addition, a model reduction technique such as the reduced basis method [18,20,25] is employed to construct a reduced basis approximation of the LF model. Our goal is to effectively combine all of these models to make an efficient and accurate prediction of the input–output relationship of the HF model.

The use of multi-fidelity models to make an efficient inference about the input–output relationship of the HF model is not a new concept. Kennedy and O'Hagan [11] propose Gaussian process (GP) models to calculate the covariances and develop a Bayesian approach to predict the output of an expensive HF simulation code with the assistance of lower-fidelity simulation codes. This approach has been further extended by other researchers [7,22,21] and applied to multi-fidelity design optimization [4,5,9]. However, because the Bayesian Gaussian process approach treats all multi-fidelity simulation codes as *computational black box*, it is required to estimate the correlation between the HF model and the LF model.

Our approach differs from the Bayesian Gaussian process approach in that it makes full use of the mathematical structure of the LF model, while treating the HF model as black box. The LF model often contains useful information about the HF model because it captures some of the physics and the conditions of the HF model. As demonstrated in our recent work [17], the GFR method can yield much more accurate prediction than the standard Gaussian process regression [19,24,26] for the same number of observations because it makes use of the mathematical structure of the model. Furthermore, the mathematical structure of the LF model will be exploited to construct the reduced basis model for efficient prediction in real-time and many-query contexts [25].

In our approach, the role of the HF model is limited to generating numerical observations that are used to improve the fidelity of the LF model using the GFR method. Since the HF model is expensive, we would like to keep the number of observations as small as possible. The selection of inputs at which to obtain the numerical observations is an *experimental design problem* [26]. As it will become clear later, the uncertainty estimate is given by the posterior distribution. The experimental design problem can be posed as choosing the inputs to minimize the posterior distribution with some appropriate design criteria. Common design criteria include A-optimality which minimizes the average of the posterior variance, and G-optimality which minimizes the maximum value of the posterior variance. In sequential experimental design, we evaluate observations as they are collected and choose the next input in light of the previously selected observations. Herein we develop a greedy algorithm to select the training inputs in sequential fashion.

The present work offers several new contributions relative to the previous work [14,17]. This work aims at predicting the input–output relationship of parametrized PDE models, whereas the previous work focuses on state prediction for PDE models without parameters. This work consider multi-fidelity models including model reduction, while the previous work does not. Lastly, this work tackles the experimental design problem in parameter space and develops an algorithm, which is entirely different from that developed in [14].

The paper is organized as follows. In section 2, we present a mathematical description of our objectives. In Section 3, we introduce the GFR method to fuse the HF model with the LF model. In Section 4, we describe our approach to combining the GFR method with the RB method for efficient prediction of the HF output. In Section 5, we present numerical results to demonstrate the proposed approach. Finally, in Section 6, we provides some concluding remarks on future research.

2. Objectives

Let V^{hi} be an appropriate Hilbert space defined on a physical domain $\Omega \in \mathbb{R}^D$, a spatial point of which is denoted by $\boldsymbol{x}$. The weak formulation of a high-fidelity parametrized PDE model can be stated as follows. For a given input $\boldsymbol{\mu} \equiv (\mu_1, \ldots, \mu_P) \in \mathcal{D} \subset \mathbb{R}^P$ we seek a solution $u^{\text{hi}}(\boldsymbol{\mu}) \in V^{\text{hi}}$ such that

$$a^{\text{hi}}(u^{\text{hi}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = b^{\text{hi}}(v, \boldsymbol{\mu}), \quad \forall\, v \in V^{\text{hi}}. \tag{1}$$

Here $a^{\text{hi}} : V^{\text{hi}} \times V^{\text{hi}} \times \mathcal{D} \to \mathbb{R}$ is a parametrized operator which can be nonlinear with respect to the HF field variable $u^{\text{hi}}(\boldsymbol{\mu})$; and $b^{\text{hi}} : V^{\text{hi}} \times \mathcal{D} \to \mathbb{R}$ is a parametrized continuous linear functional. Both a^{hi} and b^{hi} may depend on the parameter input $\boldsymbol{\mu}$ which resides in the input space $\mathcal{D}$. We are interested in computing an output $\boldsymbol{s}^{\text{hi}}(\boldsymbol{\mu}) \in \mathbb{R}$ of the HF model (1), which is expressed as a linear functional of the state $u^{\text{hi}}(\boldsymbol{\mu})$ as

$$s^{\text{hi}}(\boldsymbol{\mu}) = c(u^{\text{hi}}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{2}$$

Here $c : V^{\text{hi}} \times \mathcal{D} \to \mathbb{R}$ is a linear functional which may also depend on the input vector $\boldsymbol{\mu}$. Note that we do not need to know the mathematical structure of the HF model, namely, a^{hi} and b^{hi}. The HF model can thus be considered as a black box.

Let us also assume that we are given the following LF model: For a given input $\boldsymbol{\mu} \in \mathcal{D}$ we seek $(u^{\text{lo}}(\boldsymbol{\mu}), s^{\text{lo}}(\boldsymbol{\mu})) \in V \times \mathbb{R}$ such that

$$a^{\text{lo}}(u^{\text{lo}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = b^{\text{lo}}(v, \boldsymbol{\mu}), \quad \forall\, v \in V, \tag{3a}$$

$$s^{\text{lo}}(\boldsymbol{\mu}) = c(u^{\text{lo}}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{3b}$$

Here the weak formulation (3a) results from an appropriate simplification of the HF model (1) and V is a suitable finite element approximation space. More specifically, we assume that $a^{\text{lo}}(\cdot,\cdot,\boldsymbol{\mu})$ is a parametrized bilinear form and $b^{\text{lo}}(\cdot,\boldsymbol{\mu})$ is a parametrized continuous linear functional. We shall suppress the superscript of both a^{lo} and b^{lo} to simplify the notation. In general, the output of the LF model $s^{\text{lo}}(\boldsymbol{\mu})$ may not be a good approximation to the output of the HF model $s^{\text{hi}}(\boldsymbol{\mu})$ because of the model simplification.

Finally, we assume that we are given a reduced basis space $V_N = \text{span}\{\zeta_n \in V, 1 \le n \le N\}$ of dimension N. We consider the following RB approximation to the LF model (3): Find $(u_N^{\text{lo}}(\boldsymbol{\mu}), s_N^{\text{lo}}(\boldsymbol{\mu})) \in V_N \times \mathbb{R}$ such that

$$a(u_N^{\text{lo}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = b(v, \boldsymbol{\mu}), \quad \forall\, v \in V_N, \tag{4a}$$

$$s_N^{\text{lo}}(\boldsymbol{\mu}) = c(u_N^{\text{lo}}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{4b}$$

Since the RB approximation is constructed upon the LF model (3), the RB output $s_N^{\text{lo}}(\boldsymbol{\mu})$ may not well approximate the HF output $s^{\text{hi}}(\boldsymbol{\mu})$ because of both the model simplification and the RB approximation error induced by $V_N \subset V$.

We suppose that we can run the HF model (1) at $\boldsymbol{\mu} = \boldsymbol{\mu}_m, m = 1, \ldots, M$ to obtain $s^{\text{hi}}(\boldsymbol{\mu}_1), \ldots, s^{\text{hi}}(\boldsymbol{\mu}_M)$, where $\boldsymbol{\mu}_m \in \mathcal{D}$, $m = 1, \ldots, M$ are training inputs. We wish to combine the numerical observations $\{(\boldsymbol{\mu}_m, s^{\text{hi}}(\boldsymbol{\mu}_m)), 1 \le m \le M\}$ with the LF model (3) and the RB model (4) to make an efficient inference about the HF output $s^{\text{hi}}(\boldsymbol{\mu})$ for any input $\boldsymbol{\mu} \in \mathcal{D}$. Our goal is twofold. First, we construct a statistical surrogate model that inherits the fidelity of the HF model and has the complexity of the RB model. Second, we develop an efficient algorithm for choosing the training inputs so as to reduce the prediction uncertainty. Toward our goal, we apply the GFR method described in the next section to build a stochastic model by fusing the LF model with the numerical observations from the HF model. We then incorporate the RB approximation described in Section 4 into the stochastic model to obtain the desired surrogate model and devise a greedy sampling to select the training inputs.

3. Gaussian functional regression

In this section, we introduce the GFR method for output prediction by combining the numerical observations from the HF model with the LF model. The main ingredients of the GFR method include (*i*) a Gaussian functional that is augmented with the linear PDE model to represent various sources of uncertainty in the model; (*ii*) a functional regression procedure that yields the posterior distribution of the output estimate in light of the numerical observations; and (*iii*) a selection of an appropriate covariance operator that controls the posterior distribution.

3.1. Stochastic model

Let $g : V \times \mathcal{D} \to \mathbb{R}$ be a parametrized linear functional. We consider the following model: For any input $\boldsymbol{\mu} \in \mathcal{D}$ we find $(u(\boldsymbol{\mu}), s(\boldsymbol{\mu})) \in V \times \mathbb{R}$ such that

$$a(u(\boldsymbol{\mu}), v, \boldsymbol{\mu}) + g(v, \boldsymbol{\mu}) = b(v, \boldsymbol{\mu}), \quad \forall v \in V, \tag{5a}$$

$$s(\boldsymbol{\mu}) = c(u(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{5b}$$

In order to quantify uncertainty in the original linear model (3), we characterize g as a *Gaussian functional* with zero mean linear functional and covariance operator k, namely,

$$g(v, \boldsymbol{\mu}) \sim \mathcal{GF}(0, k((v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}'))), \qquad \forall (v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}') \in V \times \mathcal{D}\,. \tag{6}$$

We will require that the covariance operator $k : (V \times \mathcal{D}) \times (V \times \mathcal{D}) \to \mathbb{R}$ is symmetric positive-definite (SPD), namely,

$$k((v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}')) = k((v', \boldsymbol{\mu}'), (v, \boldsymbol{\mu})), \quad \text{and} \quad k((v, \boldsymbol{\mu}), (v, \boldsymbol{\mu})) > 0, \tag{7}$$

for all $(v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}') \in V \times \mathcal{D}$, and $k((0,0),(0,0)) = 0$, and that k is a bilinear form with respect to v and v'. In addition, we require that k satisfies the following condition

$$\sup_{v \in V} \frac{k((v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}'))}{\|v\|_V} < \infty, \qquad \forall v, v' \in V, \text{ and } \forall \boldsymbol{\mu}, \boldsymbol{\mu}' \in \mathcal{D}, \tag{8}$$

where $\|\cdot\|_V$ denotes an appropriate norm of the function space V. This condition guarantees that g is a bounded linear functional. We will discuss the selection of an appropriate covariance operator later.

The Gaussian functional (6) can be viewed as a generalization of Gaussian processes [24]. A Gaussian process is characterized by its mean function and covariance function, whereas a Gaussian functional is characterized by its mean functional and covariance operator. The main difference between Gaussian functional and Gaussian process lies in their input spaces. More precisely, the input space of the Gaussian process is a finite-dimensional parameter space. In contrast, the input space of the Gaussian functional is the tensor product of the function space V and the parameter space $\mathcal{D}$, $V \times \mathcal{D}$, which can be infinite-dimensional. This allows the GFR method to work with PDE models.

Under the prior (6), the model (5) becomes a stochastic parametrized model whose state $u(\boldsymbol{x}, \boldsymbol{\mu})$ is a Gaussian process with mean function u^{prior} and covariance function U^{prior}:

$$u(\boldsymbol{x},\boldsymbol{\mu}) \sim \mathcal{GP}(u^{\text{prior}}(\boldsymbol{x},\boldsymbol{\mu}), U^{\text{prior}}((\boldsymbol{x},\boldsymbol{\mu}),(\boldsymbol{x}',\boldsymbol{\mu}'))), \qquad \forall\, (\boldsymbol{x},\boldsymbol{\mu}),(\boldsymbol{x}',\boldsymbol{\mu}') \in \Omega\times\mathcal{D}\,. \tag{9}$$

The mean function u^{prior} and the covariance function U^{prior} are determined by inserting the prior (6) into the model (5). It thus follows that the output $s(\boldsymbol{\mu})$ is a random variable with normal distribution

$$s(\boldsymbol{\mu}) \sim \mathcal{N}(s^{\text{prior}}(\boldsymbol{\mu}), \vartheta^{\text{prior}}(\boldsymbol{\mu})), \qquad \forall\, \boldsymbol{\mu}\in\mathcal{D}\,. \tag{10}$$

It can be easily shown that the mean output $s^{\text{prior}}(\boldsymbol{\mu})$ is equal to $s^{\text{lo}}(\boldsymbol{\mu})$, where $s^{\text{lo}}(\boldsymbol{\mu})$ is the output of the deterministic LF model (3); and the variance $\vartheta^{\text{prior}}(\boldsymbol{\mu})$ is given by $\vartheta^{\text{prior}}(\boldsymbol{\mu}) = k((\phi(\boldsymbol{\mu}),\boldsymbol{\mu}),(\phi(\boldsymbol{\mu}),\boldsymbol{\mu}))$, where $\phi(\boldsymbol{\mu})$ is the solution of the adjoint problem (11) associated with the output functional c. It is important to point out that the normal distribution (10) is the *prior*. We next determine the *posterior distribution* of the stochastic output $s(\boldsymbol{\mu})$ using numerical observations from the HF model (1).

3.2. Posterior distribution of the output estimate

We begin with introducing the adjoint problem: for any given $\boldsymbol{\mu}\in\mathcal{D}$ the adjoint state $\phi(\boldsymbol{\mu})\in V$ satisfies

$$a(v,\phi(\boldsymbol{\mu}),\boldsymbol{\mu}) = -c(v,\boldsymbol{\mu}), \quad \forall v\in V\,. \tag{11}$$

It then follows from (5) and (11) that

$$s(\boldsymbol{\mu}) = c(u(\boldsymbol{\mu}),\boldsymbol{\mu}) = -a(u(\boldsymbol{\mu}),\phi(\boldsymbol{\mu}),\boldsymbol{\mu}) = g(\phi(\boldsymbol{\mu}),\boldsymbol{\mu}) - b(\phi(\boldsymbol{\mu}),\boldsymbol{\mu}), \tag{12}$$

and from (3) and (11) that

$$b(\phi(\boldsymbol{\mu}),\boldsymbol{\mu}) = a(u^{\text{lo}}(\boldsymbol{\mu}),\phi(\boldsymbol{\mu}),\boldsymbol{\mu}) = -c(u^{\text{lo}}(\boldsymbol{\mu}),\boldsymbol{\mu}) = -s^{\text{lo}}(\boldsymbol{\mu}). \tag{13}$$

Combining (12) and (13) we arrive at the following equation

$$s(\boldsymbol{\mu}) = s^{\text{lo}}(\boldsymbol{\mu}) + g(\phi(\boldsymbol{\mu}),\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu}\in\mathcal{D}. \tag{14}$$

This expression shows that the output $s(\boldsymbol{\mu})$ is a sum of the original output $s^{\text{lo}}(\boldsymbol{\mu})$ and the functional g of the adjoint state $\phi(\boldsymbol{\mu})$.

We assume that we are given a set of M training inputs $T_M = \{\boldsymbol{\mu}_m\in\mathcal{D}, 1\le m\le M\}$. We apply the equation (14) at the training inputs to obtain

$$s(\boldsymbol{\mu}_m) = s^{\text{lo}}(\boldsymbol{\mu}_m) + g(\phi(\boldsymbol{\mu}_m),\boldsymbol{\mu}_m), \qquad m=1,\ldots,M, \tag{15}$$

where $\phi(\boldsymbol{\mu}_m)$ is the solution of the adjoint problem (11) at the training input $\boldsymbol{\mu}_m$. By setting the output $s(\boldsymbol{\mu}_m)$ to match the HF output $s^{\text{hi}}(\boldsymbol{\mu}_m)$, we arrive at

$$s^{\text{hi}}(\boldsymbol{\mu}_m) - s^{\text{lo}}(\boldsymbol{\mu}_m) = g(\phi(\boldsymbol{\mu}_m),\boldsymbol{\mu}_m), \qquad m=1,\ldots,M. \tag{16}$$

The computational cost to generate the training data includes M solutions of the HF model (1) for the HF outputs $s^{\text{hi}}(\boldsymbol{\mu}_m)$, M solutions of the LF model (3) for the LF outputs $s^{\text{lo}}(\boldsymbol{\mu}_m)$, and M adjoint solutions $\phi(\boldsymbol{\mu}_m)$. For notational convenience, we rewrite (16) as

$$d^{\text{hi}}_{M\,m} - d^{\text{lo}}_{M\,m} = g(\psi_m,\boldsymbol{\mu}_m), \qquad m=1,\ldots,M, \tag{17}$$

where $d^{\text{hi}}_{M\,m} = s^{\text{hi}}(\boldsymbol{\mu}_m)$, $d^{\text{lo}}_{M\,m} = s^{\text{lo}}(\boldsymbol{\mu}_m)$, and $\psi_m = \phi(\boldsymbol{\mu}_m)$. Without loss of generality, we assume that the $\psi_m, m=1,\ldots,M$, are linearly independent. If they are not linearly independent, we remove the linearly dependent functions to obtain a new basis set in which all the functions are linearly independent.

According to the prior (6), the relationship (14), and the training data (17), the joint distribution of $(\boldsymbol{d}^{\text{hi}}_M - \boldsymbol{d}^{\text{lo}}_M)$ and $s(\boldsymbol{\mu})$ is given by

$$\begin{bmatrix} \boldsymbol{d}^{\text{hi}}_M - \boldsymbol{d}^{\text{lo}}_M \\ s(\boldsymbol{\mu}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ s^{\text{lo}}(\boldsymbol{\mu}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_M & \boldsymbol{q}_M(\boldsymbol{\mu}) \\ \boldsymbol{q}^{\text{T}}_M(\boldsymbol{\mu}) & \vartheta^{\text{prior}}(\boldsymbol{\mu}) \end{bmatrix} \right), \tag{18}$$

where $\boldsymbol{K}_M\in\mathbb{R}^{M\times M}$, $\boldsymbol{q}_M(\boldsymbol{\mu})\in\mathbb{R}^M$, and $\vartheta^{\text{prior}}(\boldsymbol{\mu})\in\mathbb{R}$ have entries

$$\begin{aligned} K_{M\,ij} &= k((\psi_i,\boldsymbol{\mu}_i),(\psi_j,\boldsymbol{\mu}_j)), && i=1,\ldots,M,\ j=1,\ldots,M, \\ q_{M\,i}(\boldsymbol{\mu}) &= k((\psi_i,\boldsymbol{\mu}_i),(\phi(\boldsymbol{\mu}),\boldsymbol{\mu})), && i=1,\ldots,M, \\ \vartheta^{\text{prior}}(\boldsymbol{\mu}) &= k((\phi(\boldsymbol{\mu}),\boldsymbol{\mu}),(\phi(\boldsymbol{\mu}),\boldsymbol{\mu})), \end{aligned} \tag{19}$$

respectively. We next apply the conditional distribution formula (see [24]) to the joint distribution (18) to obtain the *posterior distribution* of $s(\boldsymbol{\mu})$ as

Offline Phase: Given the training set $T_M = \{\boldsymbol{\mu}_m\}_{m=1}^M$, do the following steps once

1. Compute the vector $\boldsymbol{d}_M^{\text{hi}}$ by solving the HF model at the training set.
2. Compute the vector $\boldsymbol{d}_M^{\text{lo}}$ by solving the LF model at the training set.
3. Compute $\{\psi_m \equiv \phi(\boldsymbol{\mu}_m)\}_{m=1}^M$ by solving (11) at the training set.
4. Compute $\boldsymbol{\theta}$ by maximizing the log marginal likelihood (see Section 3.3).
5. Form $K_{M\,ij} = k((\psi_i, \boldsymbol{\mu}_i), (\psi_j, \boldsymbol{\mu}_j)), 1 \le i, j \le M$, and $\boldsymbol{\beta}_M = \boldsymbol{K}_M^{-1}(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})$.

Online Phase: Given new input $\boldsymbol{\mu} \in \mathcal{D}$, do

1. Compute the LF output $s^{\text{lo}}(\boldsymbol{\mu})$.
2. Compute the adjoint state $\phi(\boldsymbol{\mu})$ by solving the adjoint problem (11).
3. Compute the prior variance $\vartheta^{\text{prior}}(\boldsymbol{\mu}) = k((\phi(\boldsymbol{\mu}), \boldsymbol{\mu}), (\phi(\boldsymbol{\mu}), \boldsymbol{\mu}))$.
4. Compute the vector $q_{M\,m}(\boldsymbol{\mu}) = k((\psi_m, \boldsymbol{\mu}_m), (\phi(\boldsymbol{\mu}), \boldsymbol{\mu})), m = 1, \ldots, M$.
5. Evaluate $s_M^{\text{post}}(\boldsymbol{\mu})$ and $\vartheta_M^{\text{post}}(\boldsymbol{\mu})$ according to (21).

Fig. 1. Offline–Online algorithm for computing the posterior distribution of the output estimate.

$$s(\boldsymbol{\mu}) \sim \mathcal{N}(s_M^{\text{post}}(\boldsymbol{\mu}), \vartheta_M^{\text{post}}(\boldsymbol{\mu})), \tag{20}$$

where the posterior mean and variance are given by

$$\begin{aligned} s_M^{\text{post}}(\boldsymbol{\mu}) &= s^{\text{lo}}(\boldsymbol{\mu}) + \boldsymbol{q}_M^{\text{T}}(\boldsymbol{\mu})\boldsymbol{\beta}_M, \\ \vartheta_M^{\text{post}}(\boldsymbol{\mu}) &= \vartheta^{\text{prior}}(\boldsymbol{\mu}) - \boldsymbol{q}_M^{\text{T}}(\boldsymbol{\mu})\boldsymbol{K}_M^{-1}\boldsymbol{q}_M(\boldsymbol{\mu}). \end{aligned} \tag{21}$$

Here the vector $\boldsymbol{\beta}_M \in \mathbb{R}^M$ is given by

$$\boldsymbol{\beta}_M = \boldsymbol{K}_M^{-1}(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})\,. \tag{22}$$

Note that both the matrix $\boldsymbol{K}_M$ and vector $\boldsymbol{\beta}_M$ can be pre-computed and stored as they are independent of the input $\boldsymbol{\mu}$.

The algorithm outlining the steps to compute the posterior distribution of the output estimate is summarized in Fig. 1. Here the Offline Phase is performed only once, whereas the Online Phase can be repeated many times. Note that the HF model is invoked in the Offline Phase only. The computational cost of the Offline Phase is dominated by the cost of solving the HF model (1) at the training inputs. The computational cost of the Online Phase is dominated by the cost of solving the LF model (3) and the adjoint problem (11). The posterior mean $s^{\text{post}}(\boldsymbol{\mu})$ represents our prediction of the HF output $s^{\text{hi}}(\boldsymbol{\mu})$, while the posterior variance $\vartheta^{\text{post}}(\boldsymbol{\mu})$ is a probabilistic measure of the prediction error. Note that the posterior distribution depends critically on the training set T_M and the covariance operator k. The choice of the training set is an *experimental design problem* which will be discussed later in Section 4. We next discuss the choice of the covariance operator.

3.3. Covariance operators

We still need to specify the covariance operator k before computing the posterior distribution of the output estimate. The covariance operator k is a crucial ingredient in our approach because the posterior distribution of the output estimate strongly depends on k. We propose a class of covariance operators parametrized by a vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ of the form:

$$k((v, \boldsymbol{\mu}), (v', \boldsymbol{\mu}'); \boldsymbol{\theta}) = k_1(v, v'; \boldsymbol{\theta}_1) k_2(\boldsymbol{\mu}, \boldsymbol{\mu}'; \boldsymbol{\theta}_2). \tag{23}$$

Here $k_1(v, v'; \boldsymbol{\theta}_1)$ is a $\boldsymbol{\theta}_1$-dependent SPD bilinear operator of the form

$$k_1(v, v'; \boldsymbol{\theta}_1) = \theta_{11} \int_\Omega a_1(\boldsymbol{x}) v v' d\boldsymbol{x} + \theta_{21} \int_\Omega a_2(\boldsymbol{x}) \nabla v \cdot \nabla v' d\boldsymbol{x} + \theta_{31} \int_{\partial\Omega} a_3(\boldsymbol{x}) v v' d\boldsymbol{x}, \tag{24}$$

where the functions, $a_1(\boldsymbol{x})$, $a_2(\boldsymbol{x})$, and $a_3(\boldsymbol{x})$, can be carefully designed to give more weight to some particular area of the physical domain, depending on a particular application. Furthermore, the covariance function k_2 can be chosen as one of the many covariance functions given in [24] (Chapter 4).

To determine the hyperparameter vector $\boldsymbol{\theta}$, we note from (18) that the probability distribution of $(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})$ conditional on the hyperparameter vector $\boldsymbol{\theta}$ is Gaussian

$$(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_M(\boldsymbol{\theta})), \tag{25}$$

where the matrix $\boldsymbol{K}_M(\boldsymbol{\theta})$ as defined in (19) depends on $\boldsymbol{\theta}$ since the covariance operator k is parametrized by $\boldsymbol{\theta}$. It thus follows that the log marginal likelihood is given by

$$\log p((\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})|\boldsymbol{\theta}) = -\frac{1}{2}(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})^{\text{T}} \boldsymbol{K}_M(\boldsymbol{\theta})^{-1}(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}}) - \frac{1}{2}\log(\det(\boldsymbol{K}_M(\boldsymbol{\theta}))) - \frac{M}{2}\log(2\pi). \tag{26}$$

We follow the marginal likelihood maximization approach widely used in Gaussian processes [24] to determine $\boldsymbol{\theta}$ by solving the following optimization problem

$$\boldsymbol{\theta} = \arg \max_{\boldsymbol{\theta}' \in \mathbb{R}_+^Q} \log p((\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})|\boldsymbol{\theta}'). \tag{27}$$

Hence, the hyperparameters $\boldsymbol{\theta}$ are chosen as the maximizer of the log marginal likelihood.

Once a covariance operator is selected, we can compute the posterior distribution of the output estimate for any input by using the GFR method described earlier. The GFR method has a number of unique features. The method offers a framework for fully integrating mathematical models with observations to improve the prediction. The method characterizes model uncertainties by using infinite-dimensional functional spaces. In addition to the output estimate, the method can also give an estimate of the physical state. Furthermore, the method is nonparametric and nonintrusive, and capable of providing error bars for the prediction errors. All of these features distinguish the GFR method from other regression techniques including least squares regression [23], ridge regression [8,12,28], support vector machine regression [2,3,27], and Gaussian process regression [19,24,26]. These regression techniques do not fully incorporate mathematical models into their prediction models since they treat mathematical models as a computational black box. Mathematical models usually contain much richer information about the physical system because they capture the laws of physics and the conditions governing the behavior of the physical system. By exploiting the model form, the GFR method can provide accurate prediction beyond these regression techniques.

4. Model reduction and experimental design

In this section, we employ the reduced basis (RB) method [18,20,25] to reduce the computational cost of the output prediction in the real-time context (e.g., parameter-estimation, or control) and the many-query context (e.g., design optimization, or multi-model/scale simulation, experimental design). In these contexts, the number of input–output evaluations is often measured in the tens of thousands. As a result, the output estimate (20) may not be well suited for real-time and many-query applications because it requires the finite element approximation of the LF model (3), which is computationally expensive. It is thus desirable to replace the finite element approximation with the RB approximation when a large number of input–output evaluations is demanded. The RB method is particularly relevant for the experimental design problem in which numerous evaluations of the LF model and its adjoint are often required to search for good training inputs in the parameter space. Indeed, the greedy selection algorithm described in this section takes advantage of the RB method to make an efficient search of the parameter space.

4.1. Reduced basis method

We begin by assuming that we are given a set of N parameter points $\overline{T}_N = \{\bar{\boldsymbol{\mu}}_n, 1 \le n \le N\}$. We then introduce a primal RB space $V_N = \text{span}\{\zeta_m \equiv u^{\text{lo}}(\bar{\boldsymbol{\mu}}_n), 1 \le n \le N\}$ and an adjoint RB space $W_N = \text{span}\{\xi_n \equiv \phi(\bar{\boldsymbol{\mu}}_n), 1 \le n \le N\}$, where $u^{\text{lo}}(\bar{\boldsymbol{\mu}}_n)$ and $\phi(\bar{\boldsymbol{\mu}}_n)$ are the solution of the parametrized primal problem (3) and adjoint problem (11) at $\boldsymbol{\mu} = \bar{\boldsymbol{\mu}}_n$, respectively. (In practice, we orthogonalize the basis functions of the two RB spaces by using the Gram–Schmidt procedure to obtain orthogonal basis functions [13].)

Next, for any $\boldsymbol{\mu} \in \mathcal{D}$, we find the RB primal state $u_N^{\text{lo}}(\boldsymbol{\mu}) \in V_N$ such that

$$a(u_N^{\text{lo}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = b(v, \boldsymbol{\mu}), \quad \forall v \in V_N. \tag{28}$$

The RB primal system (28) is nothing but the Galerkin projection of the primal formation (3) onto the RB primal space V_N. Similarly, we find the RB adjoint state $\phi_N(\boldsymbol{\mu}) \in W_N$ from

$$a(v, \phi_N(\boldsymbol{\mu}), \boldsymbol{\mu}) = -c(v, \boldsymbol{\mu}), \quad \forall v \in W_N. \tag{29}$$

This RB system is the Galerkin projection of the adjoint formation (11) onto the RB adjoint space W_N. The RB output can then be evaluated as

$$s_N^{\text{lo}}(\boldsymbol{\mu}) = c(u_N^{\text{lo}}(\boldsymbol{\mu}), \boldsymbol{\mu}) - r^{\text{pr}}(\phi_N(\boldsymbol{\mu}), \boldsymbol{\mu}), \tag{30}$$

where r^{pr} is the RB primal residual

$$r^{\text{pr}}(v, \boldsymbol{\mu}) = b(v, \boldsymbol{\mu}) - a(u_N^{\text{lo}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}), \qquad \forall v \in V. \tag{31}$$

Note that the RB output (30) is enhanced with the primal residual at the RB adjoint state to ensure its rapid convergence to the finite element output $s^{\text{lo}}(\boldsymbol{\mu})$ [25].

In order to develop an efficient Offline–Online computational procedure, the RB method requires that the bilinear form a and linear functionals b, c can be expressed as the following affine decompositions

$$a(w,v,\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w,v), \quad b(v,\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu}) b^q(v), \quad c(v,\boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_c^q(\boldsymbol{\mu}) c^q(v) \tag{32}$$

where the bilinear forms a^q, linear functions b^q and c^q are independent of $\boldsymbol{\mu}$, while Θ_a^q, Θ_b^q and Θ_c^q are dependent on $\boldsymbol{\mu}$. If a, b, c do not admit the affine decompositions (32), we use the empirical interpolation method [1,6] or the "best points" interpolation method [15,16] to construct such affine decompositions.

We now express the RB approximations $u_N^{\mathrm{lo}}(\boldsymbol{\mu}) \in V_N$ and $\phi_N(\boldsymbol{\mu}) \in W_N$ as

$$u_N^{\mathrm{lo}}(\boldsymbol{\mu}) = \sum_{n=1}^{N} u_{N\,n}(\boldsymbol{\mu}) \zeta_n, \qquad \phi_N(\boldsymbol{\mu}) = \sum_{n=1}^{N} \phi_{N\,n}(\boldsymbol{\mu}) \xi_n. \tag{33}$$

Inserting these expressions into (28) and (29) we arrive at the following RB systems

$$\boldsymbol{A}_N^{\mathrm{pr}}(\boldsymbol{\mu}) \boldsymbol{u}_N(\boldsymbol{\mu}) = \boldsymbol{b}_N^{\mathrm{pr}}(\boldsymbol{\mu}), \qquad \boldsymbol{A}_N^{\mathrm{du}}(\boldsymbol{\mu}) \boldsymbol{\phi}_N(\boldsymbol{\mu}) = -\boldsymbol{c}_N^{\mathrm{du}}(\boldsymbol{\mu}), \tag{34}$$

where $\boldsymbol{A}_N^{\mathrm{pr}}(\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$, $\boldsymbol{b}_N^{\mathrm{pr}}(\boldsymbol{\mu}) \in \mathbb{R}^N$, $\boldsymbol{A}_N^{\mathrm{du}}(\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$, $\boldsymbol{c}_N^{\mathrm{du}}(\boldsymbol{\mu}) \in \mathbb{R}^N$ have entries

$$\begin{aligned} A_{N\,ij}^{\mathrm{pr}}(\boldsymbol{\mu}) &= a(\zeta_j, \zeta_i, \boldsymbol{\mu}), \quad b_{N\,i}^{\mathrm{pr}}(\boldsymbol{\mu}) = b(\zeta_i, \boldsymbol{\mu}), \quad i,j = 1,\ldots,N, \\ A_{N\,ij}^{\mathrm{du}}(\boldsymbol{\mu}) &= a(\xi_i, \xi_j, \boldsymbol{\mu}), \quad c_{N\,i}^{\mathrm{du}}(\boldsymbol{\mu}) = c(\xi_i, \boldsymbol{\mu}), \quad i,j = 1,\ldots,N. \end{aligned} \tag{35}$$

Using the affine decompositions (32), we can express these matrices and vectors as

$$\begin{aligned} \boldsymbol{A}_N^{\mathrm{pr}}(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \boldsymbol{A}_N^{\mathrm{pr},q}, \qquad \boldsymbol{b}_N^{\mathrm{pr}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu}) \boldsymbol{b}_N^{\mathrm{pr},q}, \\ \boldsymbol{A}_N^{\mathrm{du}}(\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \boldsymbol{A}_N^{\mathrm{pr},q}, \qquad \boldsymbol{c}_N^{\mathrm{du}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_c^q(\boldsymbol{\mu}) \boldsymbol{c}_N^{\mathrm{du},q}, \end{aligned} \tag{36}$$

where $\boldsymbol{A}_N^{\mathrm{pr},q} \in \mathbb{R}^{N\times N}$, $\boldsymbol{b}_N^{\mathrm{pr},q} \in \mathbb{R}^N$, $\boldsymbol{A}_N^{\mathrm{du},q} \in \mathbb{R}^{N\times N}$, $\boldsymbol{c}_N^{\mathrm{du},q} \in \mathbb{R}^N$ have entries

$$\begin{aligned} A_{N\,ij}^{\mathrm{pr},q} &= a^q(\zeta_j, \zeta_i), \quad q = 1,\ldots,Q_a, \qquad b_{N\,i}^{\mathrm{pr},q} = b^q(\zeta_i), \quad q = 1,\ldots,Q_b, \\ A_{N\,ij}^{\mathrm{du},q} &= a^q(\xi_i, \xi_j), \quad q = 1,\ldots,Q_a, \qquad c_{N\,i}^{\mathrm{du},q} = c^q(\xi_i), \quad q = 1,\ldots,Q_c, \end{aligned} \tag{37}$$

for $i,j = 1,\ldots,N$. Note that the matrices and vectors in (37) are independent of $\boldsymbol{\mu}$. Furthermore, we evaluate the RB output $s_N^{\mathrm{lo}}(\boldsymbol{\mu})$ in (28b) as

$$s_N^{\mathrm{lo}}(\boldsymbol{\mu}) = \boldsymbol{u}_N^{\mathrm{T}}(\boldsymbol{\mu}) \boldsymbol{c}_N^{\mathrm{pr}}(\boldsymbol{\mu}) - \boldsymbol{\phi}_N^{\mathrm{T}}(\boldsymbol{\mu}) \boldsymbol{b}_N^{\mathrm{du}}(\boldsymbol{\mu}) + \boldsymbol{\phi}_N^{\mathrm{T}}(\boldsymbol{\mu}) \boldsymbol{A}_N^{\mathrm{pr,du}}(\boldsymbol{\mu}) \boldsymbol{u}_N(\boldsymbol{\mu}), \tag{38}$$

where $\boldsymbol{c}_N^{\mathrm{pr}}(\boldsymbol{\mu})$, $\boldsymbol{b}_N^{\mathrm{du}}(\boldsymbol{\mu})$, and $\boldsymbol{A}_N^{\mathrm{pr,du}}(\boldsymbol{\mu})$ are calculated as

$$\boldsymbol{c}_N^{\mathrm{pr}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \Theta_c^q(\boldsymbol{\mu}) \boldsymbol{c}_N^{\mathrm{pr},q}, \quad \boldsymbol{b}_N^{\mathrm{du}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta_b^q(\boldsymbol{\mu}) \boldsymbol{b}_N^{\mathrm{du},q}, \quad \boldsymbol{A}_N^{\mathrm{pr,du}}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \boldsymbol{A}_N^{\mathrm{pr,du},q}. \tag{39}$$

Here the vectors $\boldsymbol{c}_N^{\mathrm{pr},q} \in \mathbb{R}^N$, $\boldsymbol{b}_N^{\mathrm{du},q} \in \mathbb{R}^N$, $\boldsymbol{A}_N^{\mathrm{pr,dy},q} \in \mathbb{R}^{N\times N}$ have entries $c_{N\,i}^{\mathrm{pr},q} = c^q(\zeta_i)$, $b_{N\,i}^{\mathrm{du},q} = b^q(\xi_i)$, $A_{N\,ij}^{\mathrm{pr,du},q} = a^q(\zeta_j, \xi_i)$ for $1 \le i,\ j \le N$, which are independent of $\boldsymbol{\mu}$.

The Offline–Online computational procedure can now be described as follows. In the Offline stage, we first get the primal and adjoint basis sets, $\{\zeta_n\}_{n=1}^N$ and $\{\xi_n\}_{n=1}^N$, by solving the primal and adjoint problems at the parameter inputs $\{\bar{\boldsymbol{\mu}}_n, 1 \le n \le N\}$, respectively; we then compute and store all the $\boldsymbol{\mu}$-independent matrices and vectors $\boldsymbol{A}_N^{\mathrm{pr},q}$, $\boldsymbol{A}_N^{\mathrm{du},q}$, $\boldsymbol{A}_N^{\mathrm{pr,du},q}$, $\boldsymbol{b}_N^{\mathrm{pr},q}$, $\boldsymbol{b}_N^{\mathrm{du},q}$, $\boldsymbol{c}_N^{\mathrm{du},q}$, $\boldsymbol{c}_N^{\mathrm{pr},q}$. The Offline stage is expensive and performed *only once*. The computational expense of the Offline stage is mainly due to the cost of getting the basis sets. In the Online stage, we first assemble the $\boldsymbol{\mu}$-dependent matrices and vectors in (36); we then solve the RB systems (34) for $\boldsymbol{u}_N(\boldsymbol{\mu})$ and $\boldsymbol{\phi}_N(\boldsymbol{\mu})$; and we finally evaluate (38) and (39) to obtain the RB output $s_N^{\mathrm{lo}}(\boldsymbol{\mu})$. The Online stage can be repeated for *many different values* of $\boldsymbol{\mu}$. The operation count of the Online stage for a given input $\boldsymbol{\mu}$ is $O(2N^3 + 3Q_aN^2)$.

Finally, we briefly discuss *a posteriori* error estimation in the RB method and refer to the paper [25] for details. The errors in the energy norm can be bounded by

$$\|u^{\mathrm{lo}}(\boldsymbol{\mu}) - u_N^{\mathrm{lo}}(\boldsymbol{\mu})\|_V \le \Delta_N^{\mathrm{pr}}(\boldsymbol{\mu}), \quad \|\phi(\boldsymbol{\mu}) - \phi_N(\boldsymbol{\mu})\|_V \le \Delta_N^{\mathrm{du}}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}, \tag{40}$$

where the error bounds are defined by

$$\Delta_N^{\mathrm{pr}}(\boldsymbol{\mu}) \equiv \frac{1}{\sqrt{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}} \sup_{v\in V} \frac{r^{\mathrm{pr}}(v,\boldsymbol{\mu})}{\|v\|_V}, \qquad \Delta_N^{\mathrm{du}}(\boldsymbol{\mu}) \equiv \frac{1}{\sqrt{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}} \sup_{v\in V} \frac{r^{\mathrm{du}}(v,\boldsymbol{\mu})}{\|v\|_V}. \tag{41}$$

Here $r^{\mathrm{du}}(v,\boldsymbol{\mu}) = -c(v,\boldsymbol{\mu}) - a(v,\phi_N(\boldsymbol{\mu})\boldsymbol{\mu}), \forall v \in V$, is the RB dual residual and $\alpha_{\mathrm{LB}}(\boldsymbol{\mu})$ is the lower bound of the stability constant $\alpha(\boldsymbol{\mu})$ of the bilinear form a. Typically, the lower bound $\alpha_{\mathrm{LB}}(\boldsymbol{\mu})$ is computed using the successive constraint method [10]. The output error bound takes the form

$$|s^{\mathrm{lo}}(\boldsymbol{\mu}) - s_N^{\mathrm{lo}}(\boldsymbol{\mu})| \le \Delta_N^s(\boldsymbol{\mu}) \equiv \Delta_N^{\mathrm{pr}}(\boldsymbol{\mu})\Delta_N^{\mathrm{du}}(\boldsymbol{\mu}), \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}. \tag{42}$$

The error bounds are computed using an Offline–Online procedure which is quite similar to that for the RB output. The Offline stage is expensive and done once; but the Online computational cost for computing these error bounds is $O(2Q_a^2N^2)$. The error bounds play an important role in ensuring efficiency and reliability of the RB method. Furthermore, these error bounds are used in the greedy sampling algorithm to obtain the parameter set $\overline{T}_N = \{\bar{\boldsymbol{\mu}}_n, 1 \le n \le N\}$.

4.2. Gaussian functional regression using the RB method

We now incorporate the RB approximation into the Gaussian functional regression described in Section 3 to provide efficient prediction of the HF output. Instead of the posterior distribution (20) we compute the following posterior distribution

$$s_N(\boldsymbol{\mu}) \sim \mathcal{N}(s_{M,N}^{\mathrm{post}}(\boldsymbol{\mu}), \vartheta_{M,N}^{\mathrm{post}}(\boldsymbol{\mu})), \tag{43}$$

where the posterior mean output and variance are given by

$$\begin{aligned} s_{M,N}^{\mathrm{post}}(\boldsymbol{\mu}) &= s_N^{\mathrm{lo}}(\boldsymbol{\mu}) + \boldsymbol{q}_{M,N}^{\mathrm{T}}(\boldsymbol{\mu})\boldsymbol{\beta}_M, \\ \vartheta_{M,N}^{\mathrm{post}}(\boldsymbol{\mu}) &= \vartheta_N^{\mathrm{prior}}(\boldsymbol{\mu}) - \boldsymbol{q}_{M,N}^{\mathrm{T}}(\boldsymbol{\mu})\boldsymbol{K}_M^{-1}\boldsymbol{q}_{M,N}(\boldsymbol{\mu}). \end{aligned} \tag{44}$$

Here the scalar $\vartheta_N^{\mathrm{prior}}(\boldsymbol{\mu})$ and the vector $\boldsymbol{q}_{M,N}(\boldsymbol{\mu}) \in \mathbb{R}^M$ are as follows

$$\begin{aligned} \vartheta_N^{\mathrm{prior}}(\boldsymbol{\mu}) &= k((\phi_N(\boldsymbol{\mu}),\boldsymbol{\mu}),(\phi_N(\boldsymbol{\mu}),\boldsymbol{\mu})), \\ q_{M,N\,m}(\boldsymbol{\mu}) &= k((\phi(\boldsymbol{\mu}_m),\boldsymbol{\mu}_m),(\phi_N(\boldsymbol{\mu}),\boldsymbol{\mu})), \quad m = 1,\ldots,M. \end{aligned} \tag{45}$$

We note from (23) and (33) that

$$\begin{aligned} \vartheta_N^{\mathrm{prior}}(\boldsymbol{\mu}) &= k_1(\phi_N(\boldsymbol{\mu}),\phi_N(\boldsymbol{\mu}))k_2(\boldsymbol{\mu},\boldsymbol{\mu}) \\ &= \sum_{n=1}^N \sum_{m=1}^N \phi_{N\,n}(\boldsymbol{\mu})k_1(\xi_m,\xi_n)\phi_{N\,m}(\boldsymbol{\mu})k_2(\boldsymbol{\mu},\boldsymbol{\mu}) \\ &= \left(\boldsymbol{\phi}_N^{\mathrm{T}}(\boldsymbol{\mu})\boldsymbol{H}_N\boldsymbol{\phi}_N(\boldsymbol{\mu})\right)k_2(\boldsymbol{\mu},\boldsymbol{\mu}), \end{aligned} \tag{46}$$

and

$$\begin{aligned} q_{M,N\,m}(\boldsymbol{\mu}) &= k_1(\phi(\boldsymbol{\mu}_m),\phi_N(\boldsymbol{\mu}))k_2(\boldsymbol{\mu}_m,\boldsymbol{\mu}) \\ &= \sum_{n=1}^N \phi_{N\,n}(\boldsymbol{\mu})k_1(\phi(\boldsymbol{\mu}_m),\xi_n)k_2(\boldsymbol{\mu}_m,\boldsymbol{\mu}) \\ &= \left(\sum_{n=1}^N \phi_{N\,n}(\boldsymbol{\mu})R_{M,N\,mn}\right)k_2(\boldsymbol{\mu}_m,\boldsymbol{\mu}), \quad m = 1,\ldots,M, \end{aligned} \tag{47}$$

where $\boldsymbol{H}_N \in \mathbb{R}^{N\times N}$ and $\boldsymbol{R}_{M,N} \in \mathbb{R}^{M\times N}$ have entries

$$\begin{aligned} H_{N\,ij} &= k_1(\xi_i,\xi_j), & 1 \le i,j \le N, \\ R_{M,N\,mn} &= k_1(\phi(\boldsymbol{\mu}_m),\xi_n), & 1 \le m \le M, 1 \le n \le N. \end{aligned} \tag{48}$$

Note that since $\boldsymbol{H}_N$ and $\boldsymbol{R}_{M,N}$ are independent of $\boldsymbol{\mu}$, they can be pre-computed and stored.

We develop an Offline–Online algorithm listed in Fig. 2 to efficiently compute the posterior distribution (43) for any input $\boldsymbol{\mu} \in \mathcal{D}$. The (Online) computational cost of evaluating this posterior distribution for any input $\boldsymbol{\mu} \in \mathcal{D}$ is $O(2N^3 + 3Q_aN^2 + M^2)$. The main advantage of using the RB method is that we only solve the RB systems (34) to obtain the posterior distribution (44) for any new input $\boldsymbol{\mu}$, whereas without using the RB method we have to solve the linear parametrized PDE and its adjoint. This could lead to a substantial saving in the computational time for real-time and many-query applications.

Offline Phase: Given $\{\boldsymbol{\mu}_m\}_{m=1}^M$ and $\{\bar{\boldsymbol{\mu}}_n\}_{n=1}^N$ do the following steps once

1. Compute the vector $\boldsymbol{d}_M^{\text{hi}}$ by solving the HF model (1) for $\boldsymbol{\mu} = \boldsymbol{\mu}_m, 1 \le m \le M$.
2. Compute the vector $\boldsymbol{d}_M^{\text{lo}}$ by solving the LF model (3) for $\boldsymbol{\mu} = \boldsymbol{\mu}_m, 1 \le m \le M$.
3. Compute $\{\psi_m \equiv \phi(\boldsymbol{\mu}_m)\}_{m=1}^M$ by solving (11) for $\boldsymbol{\mu} = \boldsymbol{\mu}_m, 1 \le m \le M$.
4. Compute $\boldsymbol{\theta}$ by maximizing the log marginal likelihood (see Section 3.3).
5. Form $K_{M\,ij} = k((\psi_i, \boldsymbol{\mu}_i), (\psi_j, \boldsymbol{\mu}_j)),\ 1 \le i, j \le M,$ and $\boldsymbol{\beta}_M = \boldsymbol{K}_M^{-1}(\boldsymbol{d}_M^{\text{hi}} - \boldsymbol{d}_M^{\text{lo}})$.
6. Perform the Offline Stage of the RB method and form $\boldsymbol{H}_N$ and $\boldsymbol{R}_{M,N}$ by (48).

Online Phase: Given new input $\boldsymbol{\mu} \in \mathcal{D}$, do

1. Solve the RB systems (34) for $\boldsymbol{u}_N(\boldsymbol{\mu})$ and $\boldsymbol{\phi}_N(\boldsymbol{\mu})$.
2. Compute the RB output $s_N^{\text{lo}}(\boldsymbol{\mu})$ by (38).
3. Compute $\vartheta_N^{\text{prior}}(\boldsymbol{\mu})$ by (46) and $\boldsymbol{q}_{M,N}(\boldsymbol{\mu})$ by (47).
4. Evaluate $s_{M,N}^{\text{post}}(\boldsymbol{\mu})$ and $\vartheta_{M,N}^{\text{post}}(\boldsymbol{\mu})$ according to (44).

Fig. 2. Offline–Online algorithm for computing the posterior distribution of the output estimate using the RB method.

4.3. Experimental design

Since numerical observations are computationally expensive as they require us to simulate the HF model, it is desired to use as few observations as possible. A principled way of reducing the number of observations is *experimental design* [26]. The notion of optimality in experimental design depends on a statistical model and an appropriate statistical criterion. Common criteria include A-optimal design which chooses the training inputs to minimize the average of the posterior variance or G-optimal design which chooses the training inputs to minimize the maximum value of the posterior variance. If the number of observations can be fixed in advance, we can simultaneously optimize the training inputs using one of these design criteria. The resulting strategy is called *optimal experimental design*. If the number of observations cannot be fixed in advance, we can evaluate observations as they are collected and optimize the next training input based on the previously selected observations. This strategy is called *sequential experimental design*.

In the present work, sequential experimental design is preferred for two main reasons. The first reason is that it is difficult to find a global optimum of the optimal experimental design problem because the optimization problem is typically nonlinear and nonconvex with numerous local optima. If the global optimum cannot be found and verified then it is not clear optimal experimental design will perform better than sequential experimental design. The second reason has to do with practical applications that do not favor using a fixed number of observations because it cannot be determined in advance. Specifically, using too few observations may risk inaccurate prediction, whereas using too many observations leads to high computational cost. In these applications, the judicious choice of a sequential design can bring about a considerable reduction in the observations necessary to reduce the uncertainty to a desired low level.

In our setting, the sequential design problem can be described as follows. We assume that we are given a large set of potential inputs $\Xi^{\text{greedy}} = \{\boldsymbol{\eta}_l \in \mathcal{D}, 1 \le l \le L\}$ with $L \gg M$, each of which corresponds to a potential numerical observation to be computed. Before actually performing any of these computations, we face the following decision questions. Which input among the L potential inputs should be selected first? Given the results of any previously selected inputs, which input should be chosen next? How many inputs are enough to provide accurate enough prediction? These decision questions can be summed up as follows: how do we choose a sequence of the training inputs $\{\boldsymbol{\mu}_m\}_{m=1}^M$ among the potential observation inputs $\{\boldsymbol{\eta}_l\}_{l=1}^L$?

We now describe our sequential experimental design strategy. We initialize the training set $T_M = \{\boldsymbol{\mu}_1\}$ with only $M = 1$ training input, where the first training input $\boldsymbol{\mu}_1$ is found in the greedy set Ξ^{greedy} such that

$$\boldsymbol{\mu}_1 = \arg \max_{\boldsymbol{\mu} \in \Xi^{\text{greedy}}} \vartheta_N^{\text{prior}}(\boldsymbol{\mu}). \tag{49}$$

We next pursue the Offline Phase of the algorithm listed in Fig. 2 with the initial training set T_M. Then, for $M = 2, \ldots, M_{\max}$, we find the next training input $\boldsymbol{\mu}_M$ as

$$\boldsymbol{\mu}_M = \arg \max_{\boldsymbol{\mu} \in \Xi^{\text{greedy}}} \vartheta_{M-1,N}^{\text{post}}(\boldsymbol{\mu}), \tag{50}$$

set $T_M = T_M \cup \{\boldsymbol{\mu}_M\}$, and perform the Offline Phase. Note that the greedy selection (50) can be viewed as a heuristic (more precisely, sub-optimal) solution to the optimization problem $\max_{\boldsymbol{\mu} \in \mathcal{D}} \vartheta_{M-1,N}^{\text{post}}(\boldsymbol{\mu})$. If the size of the greedy set Ξ^{greedy} is large enough, then our greedy selection can be as good as the optimal solution of the optimization problem.

This greedy selection algorithm is similar to the greedy sampling of the reduced basis method [25]. Roughly speaking, at iteration M, the greedy algorithm appends to the previously selected training inputs a new training input $\boldsymbol{\mu}_M$ that yields

the maximum variance over all the candidates in the greedy set Ξ^{greedy}. By choosing one particular input, we make the prediction *exact* at this input and reduce uncertainty in the region around it. Therefore, it makes sense to choose the point with the largest variance so as to minimize the uncertainty. The RB method plays an important role in our greedy selection algorithm. Without the RB method, we must evaluate the posterior variance $\vartheta_M^{\text{post}}(\boldsymbol{\mu})$ defined in (21) for all $\boldsymbol{\mu} \in \Xi^{\text{greedy}}$, which in turns requires the adjoint states $\phi(\boldsymbol{\mu}), \boldsymbol{\mu} \in \Xi^{\text{greedy}}$. The use of the RB method permits the size of the greedy set Ξ^{greedy} to be very large since $\vartheta_{M,N}^{\text{post}}(\boldsymbol{\mu})$ is much faster to compute than $\vartheta_M^{\text{post}}(\boldsymbol{\mu})$.

5. Numerical results

In this section, we present numerical examples to demonstrate our approach. In each example, the HF model is a parametrized nonlinear partial differential equation and the LF model is obtained through a linearization of the HF model around a particular (fixed) state. Therefore, the LF model may not predict well the input–output relationship of the HF model. We will illustrate how our approach can help improve the prediction. We will also compare the greedy selection to other choices of the training set.

5.1. Convection–diffusion models

In order to demonstrate the present method we consider that the HF state $u^{\text{hi}}(x, \mu)$ satisfies the following parametrized Burgers equation:

$$-\mu \frac{\partial^2 u^{\text{hi}}}{\partial x^2} + u^{\text{hi}} \frac{\partial u^{\text{hi}}}{\partial x} = f(x), \quad \text{in } \Omega \equiv (0, 1), \tag{51}$$

with Dirichlet boundary conditions $u^{\text{hi}}(0) = u^{\text{hi}}(1) = 0$ and the source term $f(x) = x^2(16\sin(\pi x) + 64\sin(4\pi x))$. Here we assume that the thermal coefficient (a single parameter input) μ resides in the parameter space $\mathcal{D} \equiv [\mu_{\min}, \mu_{\max}]$, where $\mu_{\min} = 0.1$ and $\mu_{\max} = 1$. The HF output $s^{\text{hi}}(\mu)$ is the average of the solution on the physical domain, namely,

$$s^{\text{hi}}(\mu) = c(u^{\text{hi}}(\mu)) \equiv \int_\Omega u^{\text{hi}}(\mu) dx. \tag{52}$$

Evaluating the HF output requires the solution of the nonlinear model (51).

For the LF model we consider the following linear convection–diffusion equation:

$$-\mu \frac{\partial^2 u^{\text{lo}}}{\partial x^2} + w(x) \frac{\partial u^{\text{lo}}}{\partial x} = f(x), \quad \text{in } \Omega \equiv (0, 1), \tag{53}$$

with Dirichlet boundary conditions $u^{\text{lo}}(0) = u^{\text{lo}}(1) = 0$. Here the convective velocity $w(x)$ is specified as

$$w(x) = u^{\text{hi}}(x, \mu_{\max}). \tag{54}$$

The weak formulation is to find $u^{\text{lo}}(\mu) \in V(\Omega)$ such that

$$a(u^{\text{lo}}(\mu), v, \mu) = b(v), \quad \forall v \in V(\Omega), \tag{55}$$

where

$$a(w, v, \mu) = \int_\Omega \left(\mu \frac{\partial w}{\partial x} \frac{\partial v}{\partial x} + \vartheta(x) \frac{\partial w}{\partial x} v \right) dx, \qquad b(v) = \int_\Omega f v dx, \quad \forall w, v \in V(\Omega). \tag{56}$$

Here the function space $V(\Omega)$ is then given by

$$V(\Omega) = \left\{ v \ : \ \int_\Omega \left(v^2 + \frac{\partial v}{\partial x} \frac{\partial v}{\partial x} \right) dx < \infty \text{ and } v(-1) = v(1) = 0 \right\}. \tag{57}$$

In practice, we replace the continuous space $V(\Omega)$ with a 100-element finite element space of local polynomials of degree $p = 3$. The LF output is calculated as $s^{\text{lo}}(\mu) = c(u^{\text{lo}}(\mu))$. We note that the linear model (53) is a linearization of the nonlinear model (51) at a fixed state $u^{\text{hi}}(x, \mu_{\max})$, which is numerically computed using the same finite element space. Fig. 3 shows the HF output $s^{\text{hi}}(\mu)$ and the LF output $s^{\text{lo}}(\mu)$ as a function of the input μ. We see that the LF output is considerably different from the HF output.

We can now apply the RB method described in Section 4 to the linear model (55) and its adjoint. We show in Table 1 the selected parameter $\bar{\mu}_N$ and the error measure ε_N for different values of N. Here the error measure is defined as

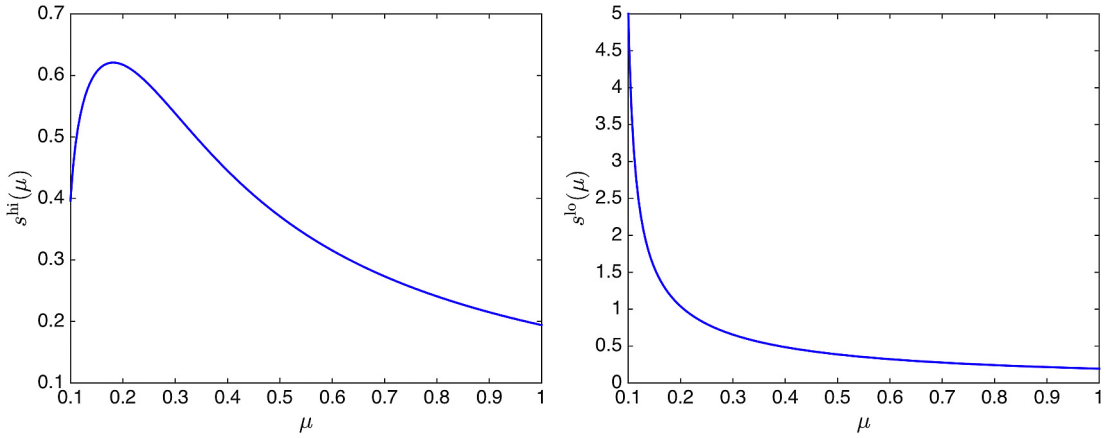


Fig. 3. The HF output $s^{\rm hi}(\mu)$ and the LF output $s^{\rm lo}(\mu)$ as a function of the input μ.

Table 1
The selected parameter $\bar{\mu}_N$ and the error measure $\varepsilon_N^{\max}$ for different values of N.

N	$\bar{\mu}_N$	ε_N
1	0.1000	4.94e−1
2	1.0000	9.98e−4
3	0.1506	1.13e−8
4	0.1128	2.75e−10
5	0.3498	4.46e−12

Table 2
The hyperparameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ as a function of M for different training sets.

M	Greedy selection	Equal-spaced nodes	Chebyshev nodes	Gauss–Labotto nodes
1	(0.155, 0.103, 1.990)	(0.155, 0.103, 1.990)	(0.155, 0.103, 1.990)	(0.155, 0.103, 1.990)
2	(0.001, 0.058, 1.990)	(0.001, 0.056, 0.149)	(0.001, 0.056, 0.149)	(0.001, 0.056, 0.149)
3	(0.001, 0.041, 0.660)	(0.001, 0.099, 1.990)	(0.001, 0.099, 1.990)	(0.001, 0.099, 1.990)
4	(0.001, 0.031, 0.282)	(0.001, 0.094, 1.990)	(0.001, 0.086, 1.550)	(0.001, 0.090, 1.705)
5	(0.001, 0.029, 0.285)	(0.001, 0.057, 0.658)	(0.001, 0.042, 0.522)	(0.001, 0.046, 0.558)
6	(0.001, 0.022, 0.169)	(0.001, 0.037, 0.405)	(0.001, 0.028, 0.323)	(0.001, 0.030, 0.343)
7	(0.001, 0.021, 0.180)	(0.001, 0.028, 0.303)	(0.001, 0.024, 0.242)	(0.001, 0.024, 0.255)
8	(0.030, 0.020, 0.185)	(0.004, 0.024, 0.251)	(0.001, 0.021, 0.197)	(0.006, 0.022, 0.209)
9	(0.200, 1.267, 0.391)	(0.001, 0.022, 0.217)	(0.001, 0.021, 0.173)	(0.040, 0.019, 0.185)

$$\varepsilon_N = \max_{\mu \in \Xi^{\rm test}} \frac{|s^{\rm lo}(\mu) - s_N^{\rm lo}(\mu)|}{|s^{\rm lo}(\mu)|}, \tag{58}$$

where $\Xi^{\rm test}$ is a set of $n^{\rm test} = 300$ parameters which are randomly chosen in the parameter space $\mathcal{D}$. As expected, most of the RB sample points are close to $\mu_{\min}$. We observe that the RB approximation converges very rapidly since the accuracy of 1.13×10^{-8} in the relative error is achieved with $N = 3$ only. Note that the (online) RB model with $N = 3$ is more than 1280 times faster to solve than the LF model. This translates to a significant saving in the computational time of the greedy selection algorithm that requires a large number of output evaluations over the greedy set $\Xi^{\rm greedy}$ of 1000 parameter points.

We will compare the performance of the greedy selection relative to other choices of the training set. To this end, we consider three other training sets. The first set $T_M^{\rm ES} = \{\mu_m^{\rm ES}\}_{m=1}^M$ is equally spaced in the parameter space; the second set $T_M^{\rm EC} = \{\mu_m^{\rm EC}\}_{m=1}^M$ consists of the Chebyshev nodes; and the third training set, $T_M^{\rm GL} = \{\mu_m^{\rm GL}\}_{m=1}^M$ consists of the Gauss–Labotto nodes. Furthermore, we take the covariance operator of the form

$$k((v,\mu),(v',\mu');\boldsymbol{\theta}) = \left(\theta_1 \int_\Omega v v' dx + \theta_2 \int_\Omega \frac{\partial v}{\partial x}\frac{\partial v'}{\partial x} dx\right) \exp\left(-\frac{0.5(\mu-\mu')^2}{\theta_3^2}\right), \tag{59}$$

where the hyperparameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ are determined by maximizing the log marginal likelihood (26). We present in Table 2 the hyperparameter values as a function of M.

We next present in Table 3 the average prediction error $\bar{\delta}_{M,N}^{\rm post}$ and the average standard deviation $\bar{\vartheta}_{M,N}^{\rm post}$ for different values of M with $N = 5$, where

Table 3
The average prediction error and the average standard deviation as a function of M for different training sets. The training set of the greedy selection is listed as well.

#	Greedy selection			Equal-spaced		Chebyshev		Gauss–Labotto	
M	μ_M	$\bar{\delta}^{\text{post}}_{M,N}$	$\bar{\vartheta}^{\text{post}}_{M,N}$	$\bar{\delta}^{\text{post}}_{M,N}$	$\bar{\vartheta}^{\text{post}}_{M,N}$	$\bar{\delta}^{\text{post}}_{M,N}$	$\bar{\vartheta}^{\text{post}}_{M,N}$	$\bar{\delta}^{\text{post}}_{M,N}$	$\bar{\vartheta}^{\text{post}}_{M,N}$
1	0.1000	1.28e−1	1.64e−1	1.28e−1	1.64e−1	1.28e−1	1.64e−1	1.28e−1	1.64e−1
2	0.1455	1.03e−1	1.52e−2	1.60e−2	1.88e−1	1.60e−2	1.88e−1	1.62e−2	1.88e−1
3	1.0000	7.41e−2	1.70e−2	8.29e−2	8.47e−3	8.29e−2	8.47e−3	8.29e−2	8.47e−3
4	0.4331	3.53e−2	1.81e−2	7.27e−2	2.04e−3	6.33e−2	1.96e−3	6.68e−2	1.97e−3
5	0.6591	2.74e−2	7.55e−3	5.21e−2	2.70e−3	3.57e−2	3.01e−3	4.00e−2	2.87e−3
6	0.2524	1.16e−2	9.67e−3	3.57e−2	3.14e−3	2.11e−2	3.18e−3	2.33e−2	2.98e−3
7	0.8402	1.05e−2	4.99e−3	2.45e−2	2.93e−3	1.35e−2	2.94e−3	1.42e−2	2.57e−3
8	0.5537	6.90e−3	2.10e−3	1.69e−2	2.42e−3	9.08e−3	2.59e−3	9.01e−3	2.10e−3
9	0.1117	1.95e−4	8.65e−5	1.19e−2	1.94e−3	6.12e−3	2.07e−3	5.63e−3	1.54e−3

$$\bar{\delta}^{\text{post}}_{M,N} = \frac{1}{n^{\text{test}}} \sum_{\mu \in \Xi^{\text{test}}} \left(s^{\text{hi}}(\mu) - s^{\text{post}}_{M,N}(\mu) \right), \qquad \bar{\vartheta}^{\text{post}}_{M,N} = \frac{1}{n^{\text{test}}} \sum_{\mu \in \Xi^{\text{test}}} \sqrt{\vartheta^{\text{post}}_{M,N}(\mu)} \tag{60}$$

We see from Table 3 that the greedy selection outperforms the other choices of the training set since it provides faster convergence for the output prediction. Furthermore, the greedy selection yields more better error estimators than the other training sets. The results demonstrate the importance of the training inputs to the outcome of our method. It is clear from Fig. 3 that the training inputs should be distributed closely to the left boundary of the parameter domain since the LF model deviates significantly from the HF model when μ is small. As listed in Table 3, the training set of the greedy selection has such property.

Finally, we show in Fig. 4 the HF output $s^{\text{hi}}(\mu)$, the mean prediction $s^{\text{post}}_{N,M}(\mu)$, and the 95% confidence region (shaded area) as a function of μ for the greedy selection and the Gauss–Labotto nodes. Here the 95% confidence region is an area bounded by the mean prediction plus and minus two times the standard deviation function. Note that both the prediction error and the standard deviation function are zero at the training points. We see that the prediction is more accurate when μ is close to $\mu_{\max}$ and becomes less accurate when μ is close to $\mu_{\min}$. Furthermore, the greedy selection outperforms the Gauss–Labotto nodes since it provides more accurate predictions for the same value of M. Note that the results of the Chebyshev nodes are similar to those of the Gauss–Labotto nodes, while the results of the equal-spaced nodes are even worse.

5.2. Diffusion–reaction models

We consider the following parametrized nonlinear diffusion–reaction equation:

$$-\mu_1 \nabla \cdot \left(\left(1 + \sin(2\pi x \mu_2) \sin(2\pi y \mu_2) + (u^{\text{hi}})^2 \right) \nabla u^{\text{hi}} \right) + u^{\text{hi}} = f, \quad \text{in } \Omega \equiv (0,1) \times (0,1), \tag{61}$$

with Dirichlet boundary conditions $u^{\text{hi}} = 0$ on the physical domain boundary and the source term $f(x,y) = 4\pi^2 \sin(1.5\pi x) \times \sin(1.5\pi y)$. Here we assume that the input $\boldsymbol{\mu} \equiv (\mu_1, \mu_2)$ resides in the parameter space $\mathcal{D} \equiv [\mu_{\min}, \mu_{\max}] \times [\mu_{\min}, \mu_{\max}]$, where $\mu_{\min} = 0.1$ and $\mu_{\max} = 1$. The HF output $s^{\text{hi}}(\boldsymbol{\mu})$ is the average of the solution on the physical domain, namely,

$$s^{\text{hi}}(\boldsymbol{\mu}) = c(u^{\text{hi}}(\boldsymbol{\mu})) \equiv \int_{\Omega} u^{\text{hi}}(\boldsymbol{\mu}) dx dy. \tag{62}$$

Evaluating the HF output requires the solution of the nonlinear model (61).

As an approximation to the nonlinear model (61) we consider the following linear diffusion–reaction equation:

$$-\mu_1 \nabla \cdot \left(\left(1 + \sin(2\pi x \mu_2) \sin(2\pi y \mu_2) + \tilde{u}^2 \right) \nabla u^{\text{hi}} \right) + u^{\text{hi}} = f, \quad \text{in } \Omega, \tag{63}$$

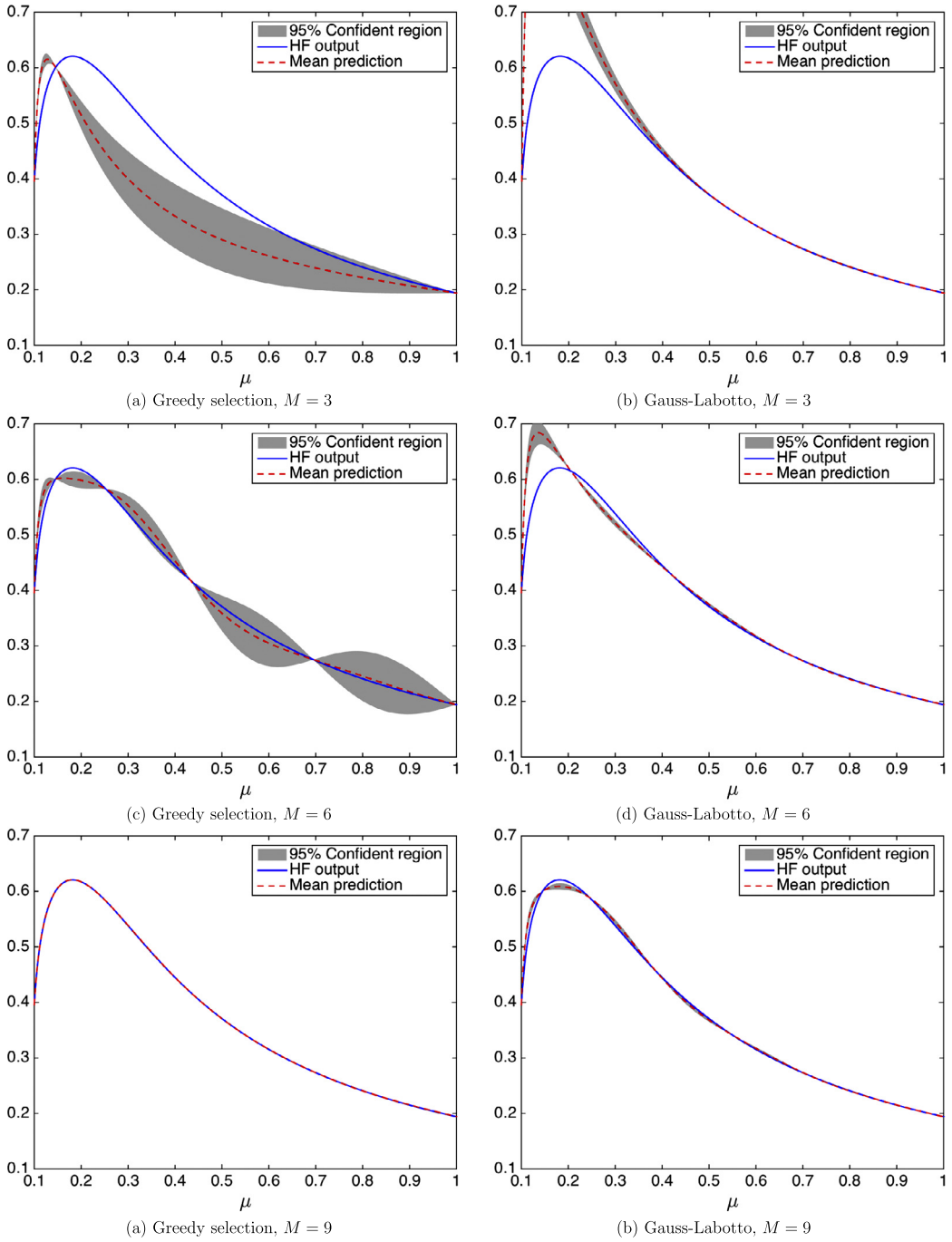
with Dirichlet boundary conditions $u^{\text{lo}} = 0$ on the entire boundary and $\tilde{u}$ being specified as

$$\tilde{u} = u^{\text{hi}}(\tilde{\boldsymbol{\mu}}), \tag{64}$$

for $\tilde{\boldsymbol{\mu}} = (0.5, 0.5)$. The weak formulation is to find $u^{\text{lo}}(\boldsymbol{\mu}) \in V(\Omega)$ such that

$$a(u^{\text{lo}}(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = b(v), \quad \forall v \in V(\Omega), \tag{65}$$

where



(a) Greedy selection, $M = 3$ (b) Gauss-Labotto, $M = 3$

(c) Greedy selection, $M = 6$ (d) Gauss-Labotto, $M = 6$

(a) Greedy selection, $M = 9$ (b) Gauss-Labotto, $M = 9$

Fig. 4. Panels show the HF output and the RB mean prediction for the greedy selection (left) and the Gauss–Labotto nodes (right). In these plots the shaded area represents the mean prediction plus and minus two times the standard deviation function (corresponding to the 95% confidence region).

$$
\begin{aligned}
a(w, v, \boldsymbol{\mu}) &= \mu_1 \int_{\Omega} \left(1 + \sin(2\pi x \mu_2)\sin(2\pi y \mu_2) + \tilde{u}^2\right) \nabla w \cdot \nabla v dx dy + \int_{\Omega} w v dx dy, \\
b(v) &= \int_{\Omega} f v dx dy,
\end{aligned}
\tag{66}
$$

for all $w, v \in V(\Omega)$. Here the function space $V(\Omega)$ is then given by

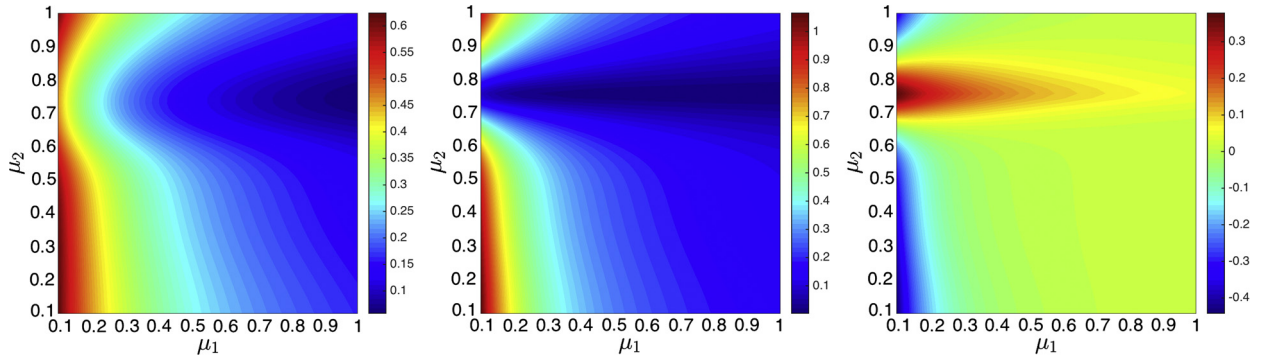


Fig. 5. Plots of the HF output $s^{\text{hi}}(\boldsymbol{\mu})$ (left), the LF output $s^{\text{lo}}(\boldsymbol{\mu})$ (center), and the difference between the HF output and the LF output (right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

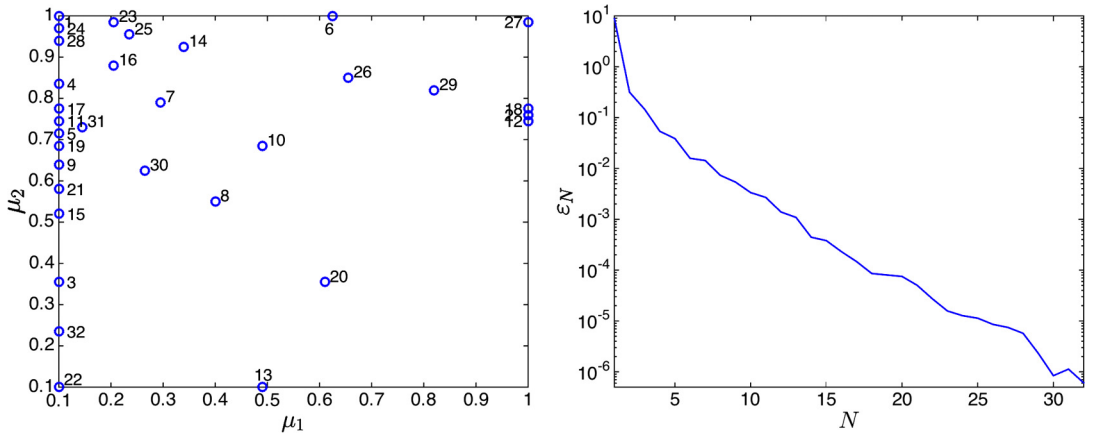


Fig. 6. Plots of the RB sample $\{\tilde{\boldsymbol{\mu}}_n\}_{n=1}^N$ (left) and the error measure ε_N (right) as a function of N. The numbers in the left figure indicates the order of the RB sample.

$$V(\Omega) = \left\{ v \;:\; \int_\Omega \left(v^2 + \frac{\partial v}{\partial x}\frac{\partial v}{\partial x} \right) dx < \infty \text{ and } v = 0 \text{ on } \partial\Omega \right\}. \tag{67}$$

In practice, we replace the continuous space $V(\Omega)$ with a 800-element finite element space of local polynomials of degree $p = 3$. We note that the linear model (65) is a fixed-point linearization of the nonlinear model (61) at the state $u^{\text{hi}}(\tilde{\boldsymbol{\mu}})$, which is numerically computed using the same finite element space. The LF output is calculated as $s^{\text{lo}}(\boldsymbol{\mu}) = c(u^{\text{lo}}(\boldsymbol{\mu}))$. Fig. 5 shows the HF output $s^{\text{hi}}(\boldsymbol{\mu})$ and the LF output $s^{\text{lo}}(\boldsymbol{\mu})$ as a function of the input $\boldsymbol{\mu}$. We see that the LF output is considerably different from the HF output especially when μ_1 is small and μ_2 is large.

We note that the bilinear form a of the linear model (65) does not admit the affine decomposition (32) due to the non-affine function $h(x, y, \mu_2) = \sin(2\pi x\mu_2)\sin(2\pi y\mu_2)$. We use the empirical interpolation method (EIM) [1,6] to construct an affine approximation $h_J(x, y, \mu_2) = \sum_{j=1}^J \gamma_j(\mu_2) w_j(x, y)$, where the coefficients γ_j, $1 \le j \le J$, are determined by requiring that $h_J(x, y, \mu_2)$ is equal to $h(x, y, \mu_2)$ at J interpolation points. The basis functions and the interpolation points are computed using the EIM procedure. We can now apply the RB method described in Section 4 to the linear model (65) in which the nonaffine function $h(x, y, \mu_2)$ is replaced with its affine approximation $h_J(x, y, \mu_2)$ for $J = 10$.

We show in Fig. 6 the RB sample $\{\tilde{\mu}_n\}_{n=1}^N$ and the error measure ε_N for different values of N. Here the RB sample is chosen from the greedy set Ξ^{greedy} of 101×101 parameter points and the error measure ε_N is similarly defined as in (58), where the test set Ξ^{test} is a uniform grid of 61×61 parameter points in the parameter space $\mathcal{D}$. We see that many parameter points are located near the top right conner of the parameter domain. This can be attributed to the fact that the solution $u(\boldsymbol{\mu})$ changes more rapidly as a function $\boldsymbol{\mu}$ when μ_1 is small and μ_2 is large. The RB approximation converges very rapidly as the maximum relative output error is less than 10^{-6} for $N = 32$. Note that the RB model with $N = 32$ is about 750 times faster than the LF model.

Next, we present in Fig. 7 the training set $\{\boldsymbol{\mu}_m\}_{m=1}^M$ as well as the average prediction error $\bar{\delta}_{M,N}^{\text{post}}$ and the average standard deviation $\bar{\vartheta}_{M,N}^{\text{post}}$ for different values of M with $N = 32$, where $\bar{\delta}_{M,N}^{\text{post}}$ and $\bar{\vartheta}_{M,N}^{\text{post}}$ are similarly defined as in (60) for the test set Ξ^{test} of size 61×61. We see that many training inputs are located near the upper left conner of the parameter domain. Indeed, the distribution of the training inputs is quite similar to that of the RB parameter points. However, there are more

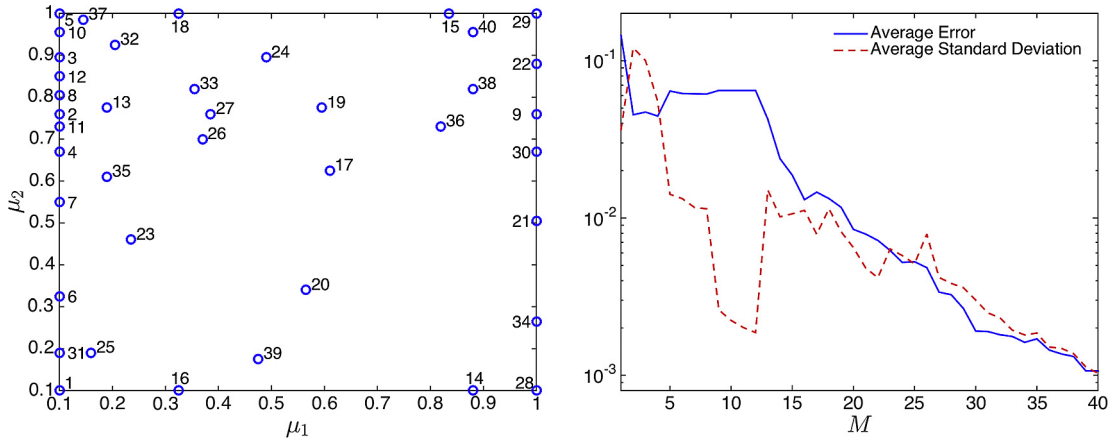
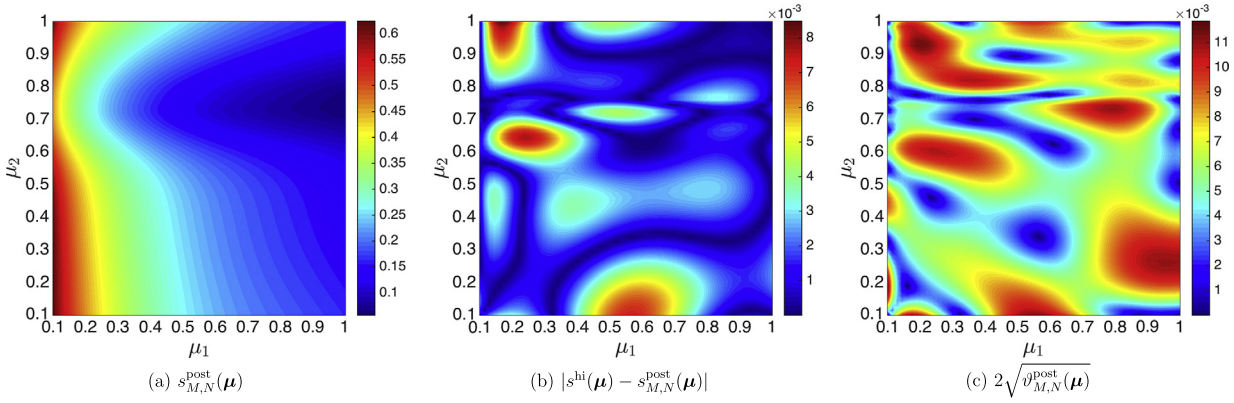


Fig. 7. Plots of the training set $\{\boldsymbol{\mu}_m\}_{m=1}^M$ (left), and the average prediction error $\bar{\delta}_{M,N}^{\text{post}}$ and the average standard deviation $\bar{\vartheta}_{M,N}^{\text{post}}$ as a function of M for $N = 32$. The numbers in the left figure indicates the order of the training set.



(a) $s_{M,N}^{\text{post}}(\boldsymbol{\mu})$ (b) $|s^{\text{hi}}(\boldsymbol{\mu}) - s_{M,N}^{\text{post}}(\boldsymbol{\mu})|$ (c) $2\sqrt{\vartheta_{M,N}^{\text{post}}(\boldsymbol{\mu})}$

Fig. 8. Plots of the mean prediction (a), the associated error (b), and the 95% confidence (c) for $M = 30$ and $N = 32$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Table 4
The average prediction error and the average standard deviation as a function of M for the equal-spaced training set, the Chebyshev training set, and the Gauss–Labotto training set.

#	Equal-spaced		Chebyshev		Gauss–Labotto	
M	$\bar{\delta}_{M,N}^{\text{post}}$	$\bar{\vartheta}_{M,N}^{\text{post}}$	$\bar{\delta}_{M,N}^{\text{post}}$	$\bar{\vartheta}_{M,N}^{\text{post}}$	$\bar{\delta}_{M,N}^{\text{post}}$	$\bar{\vartheta}_{M,N}^{\text{post}}$
4	0.0794	0.0233	0.0794	0.0233	0.0794	0.0233
9	0.0420	0.0121	0.0420	0.0121	0.0420	0.0121
16	0.0150	0.0244	0.0141	0.0230	0.0170	0.0231
25	0.0104	0.0159	0.0080	0.0203	0.0046	0.0194
36	59.772	0.0032	38.615	0.0032	31.243	0.0033
49	11.194	0.1651	21.192	0.1226	14.815	0.1009

training inputs on the right boundary than RB parameter points. As expected, both the average error and the standard deviation tend to decrease as M increases. In particular, the convergence of the mean prediction $s_{M,N}^{\text{post}}(\boldsymbol{\mu})$ to the HF output $s^{\text{hi}}(\boldsymbol{\mu})$ is quite fast as M increases. When M is large enough, the average standard deviation is very close to the average error. Hence, increasing the number of measurements reduces the uncertainty in our prediction of the HF output. We also display in Fig. 8 the mean prediction, the associated error, and the 95% confidence for $M = 30$ and $N = 32$. We see that the mean prediction for $M = 30$ resembles the HF output. Furthermore, the 95% confidence bounds the error quite well, implying that the HF output is likely to lie in the 95% confidence region.

Finally, we present in Table 4 the average prediction error $\bar{\delta}_{M,N}^{\text{post}}$ and the average standard deviation $\bar{\vartheta}_{M,N}^{\text{post}}$ for the other choices of the training set. It is interesting to note that these training sets yield the prediction error which initially decreases for $M < 36$ and grows rapidly for $M \geq 36$. The reason is that when M is large these training sets consist of many parameter points in the bottom right of the parameter domain, where the adjoint state $\phi(\boldsymbol{\mu})$ does not vary much. Consequently, the

resulting adjoint states $\{\phi(\boldsymbol{\mu}_m)\}_{m=1}^M$ are almost linearly dependent, which in turn makes the matrix $\boldsymbol{K}_M$ ill-conditioned. Our greedy selection algorithm is able to avoid this issue because it chooses the training inputs well.

6. Conclusions

In this paper, we have presented a new model assimilation approach that combines multi-fidelity models to make an efficient inference about the output of the high fidelity model. Our approach takes advantage of the mathematical structure of the LF model in two ways. We augment the LF model with a Gaussian functional and determine the posterior distribution of the Gaussian functional and the output estimate by utilizing the numerical observations from the HF model and the adjoint states from the LF model. We use the mathematical structure of the LF model to construct its RB approximation which is then incorporated into the GFR framework to enable fast prediction and experimental design. We demonstrate our approach on two numerical examples to highlight its features. Numerical results show that we can estimate the input–output relationship of the HF model quite well with a small number of HF runs even when the LF model is considerably different from the HF model. The results also show that the greedy selection outperforms both the Chebyshev nodes and the Gauss–Labotto nodes since it yields more accurate and reliable prediction.

We end the paper with a short discussion of several possible extensions and directions for further research. While we make full use of the LF model, we do not make the same use of the HF model. Indeed, the only information from the HF model being made used in our approach are the values of the output at the training inputs $\{s^{\text{hi}}(\boldsymbol{\mu}_m)\}_{m=1}^M$. Can we make use of the HF states at the training inputs $\{u^{\text{hi}}(\boldsymbol{\mu}_m)\}_{m=1}^M$? Of course, the state contains a lot more information than the output. Making use of the state may provide us valuable sources of information to improve the approach significantly. While we concentrate on single output prediction in this paper, we would like to develop an efficient procedure for estimating multiple outputs as well as the state of the HF model.

Herein we develop a greedy selection algorithm for choosing the training inputs in sequential fashion. The greedy method may not be very effective if the input is of high dimension because the size of the greedy sample will grow rapidly with the dimension of the parameter space and the RB method requires more basis functions to yield good approximation in high-dimensional input spaces. Future work should explore alternative sequential methods that are more effective than the proposed greedy method for high-dimensional input spaces. Moreover, we would like to develop efficient and robust algorithms for optimal experimental design. Both optimal and sequential experimental design can play a collaborative role in our approach: optimal design is carried out *only once* in the Offline Phase to provide the optimal training set, while sequential design may be performed *several times* in the Online Phase to choose the next training input when our prediction is not accurate enough.

Another important question we would like to tackle is that can we do better than the likelihood maximization algorithm in selecting an appropriate covariance operator? It is clear from the numerical results that the error estimates are sometimes not rigorous because the hyperparameters are not good. A better selection approach that makes full use of the model may produce a better covariance operator. Furthermore, we would like to explore other families of the covariance operators besides the family $k((\nu, \boldsymbol{\mu}), (\nu, \boldsymbol{\mu})) = k_1(\nu, \nu')k_2(\boldsymbol{\mu}, \boldsymbol{\mu})$.

Herein we consider only a single LF model and a single HF model. Extension of our approach to allowing for integration of several models with physical observations will be an interesting research. More significantly, we will need to address nonlinear LF models so as to broaden the application domain of our method to more complex mathematical models. Nonlinear LF models represent some significant challenges not only because the adjoint problems will depend on the state which we do not know, but also because the Gaussian property of the stochastic model is no longer preserved due to nonlinearity. Finally, we would like to apply our approach to important applications including multi-fidelity optimization, inverse problems, and optimal control.

Acknowledgements

We would like to thank Professor Robert M. Freund of the Sloan School of Management at M.I.T. for fruitful discussions. This work was supported by the Air Force Office of Scientific Research under the AFOSR Grant FA9550-12-1-0357 and AFOSR Grant FA9550-11-1-0141.

References

[1] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. 339 (9) (2004) 667–672.
[2] D. Basak, S. Pal, D.C. Patranabis, Support vector regression, Neural Inf. Process., Lett. Rev. 10 (2007) 203–224.
[3] R.G. Brereton, G.R. Lloyd, Support vector machines for classification and regression, Analyst 135 (2010) 230–267.
[4] A.I.J. Forrester, A. Sóbester, A.J. Keane, Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A, Math. Phys. Eng. Sci. 463 (2088) (December 2007) 3251–3269.
[5] S.E. Gano, J.E. Renaud, B. Sanders, Hybrid variable fidelity optimization by using a Kriging-based scaling function, AIAA J. 43 (11) (May 2012) 2422–2433.
[6] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, Math. Model. Numer. Anal. 41 (2) (2007) 575–605.

[7] Z.H. Han, R. Zimmermann, S. Görtz, A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data, in: 48th AIAA Aerospace Sciences Meeting, 2010, AIAA 2010-1125.
[8] A.E. Hoerl, R.W. Kennard, Ridge regression: applications to nonorthogonal problems, Technometrics 12 (1970) 69–82.
[9] D. Huang, T.T. Allen, W.I. Notz, R.A. Miller, Sequential kriging optimization using multiple-fidelity evaluations, Struct. Multidiscip. Optim. 32 (5) (May 2006) 369–382.
[10] D.B.P. Huynh, G. Rozza, S. Sen, 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, Anal. Numér. 345 (8) (2007) 473–478.
[11] M.C. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (1) (2000) 1–13.
[12] D.W. Marquardt, R.D. Snee, Ridge regression in practice, Am. Stat. 29 (3) (1975).
[13] N.C. Nguyen, Reduced-basis approximation and a posteriori error bounds for nonaffine and nonlinear partial differential equations: application to inverse analysis, PhD thesis, Singapore–MIT Alliance, National University of Singapore, 2005.
[14] N.C. Nguyen, H. Men, R.M. Freund, J. Peraire, Gaussian functional regression for state prediction using linear PDE models and observations, SIAM J. Sci. Comput. (2015), accepted for publication.
[15] N.C. Nguyen, A.T. Patera, J. Peraire, A 'best points' interpolation method for efficient approximation of parametrized functions, Int. J. Numer. Methods Eng. 73 (4) (January 2008) 521–543.
[16] N.C. Nguyen, J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, Int. J. Numer. Methods Eng. 76 (1) (October 2008) 27–55.
[17] N.C. Nguyen, J. Peraire, Gaussian functional regression for linear partial differential equations, Comput. Methods Appl. Mech. Eng. 287 (2015) 69–89.
[18] N.C. Nguyen, K. Veroy, A.T. Patera, Certified real-time solution of parametrized partial differential equations, in: S. Yip (Ed.), Handbook of Materials Modeling, Kluwer Academic Publishing, 2004, pp. 1523–1559.
[19] A. O'Hagan, J.F.C. Kingman, Curve fitting and optimal design for prediction, J. R. Stat. Soc., Ser. B, Methodol. 40 (1) (1978) 1–42.
[20] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: reduced-basis output bounds methods, J. Fluids Eng. 124 (1) (March 2002) 70–80.
[21] P.Z.G. Qian, C.F.J. Wu, Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments, Technometrics 50 (2) (2008) 192–204.
[22] Z. Qian, C.C. Seepersad, V.R. Joseph, J.K. Allen, C.F.J. Wu, Building surrogate models based on detailed and approximate simulations, J. Mech. Des. 128 (4) (July 2006) 668.
[23] C.R. Rao, H. Toutenburg, Shalabh, C. Heumann, Linear Models and Generalizations: Least Squares and Alternatives, Springer Publishing Company, Incorporated, December 2007.
[24] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.
[25] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (3) (May 2008) 229–275.
[26] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. 4 (4) (November 1989) 409–423.
[27] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222.
[28] R. Tibshirani, Regression selection and shrinkage via the Lasso, J. R. Stat. Soc. B 58 (1994) 267–288.