Journal of Computational and Applied Mathematics 301 (2016) 101–120

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# Reduced multiscale finite element basis methods for elliptic PDEs with parameterized inputs

Lijian Jiang [a,*], Qiuqi Li [b]

[a] Institute of Mathematics, Hunan University, Changsha 410082, China

[b] College of Mathematics and Econometrics, Hunan University, Changsha 410082, China

ARTICLE INFO

*Article history:*
Received 13 September 2015
Received in revised form 7 December 2015

*Keywords:*
Reduced basis methods
MsFEM
Cross-validation
Greedy algorithm
Proper orthogonal decomposition

ABSTRACT

In this paper, we present some reduced basis methods for elliptic PDEs with parameterized inputs. In the framework of Galerkin projection, dimension reduction techniques are used to construct a reduced order model. If the PDEs have multiscale structures, multiscale finite element method (MsFEM) is one of the efficient approaches to numerically solve the equations. When the inputs of the PDEs are parameterized by a few parameters, the MsFE basis functions usually depend on the parameters. This impacts on the computation efficiency. In order to get the multiscale basis functions independent of parameters, we can build multiscale basis functions based on a set of samples in the parameter space. This will result in a high dimensional MsFE space for approximation and bring great challenge for simulation. To treat this difficulty, we use some optimal strategies to identify a set of optimal reduced basis functions from the high dimensional MsFE space and obtain a reduced order multiscale model. We consider three optimal strategies for model reduction: cross-validation method, greedy algorithm and proper orthogonal decomposition. The dimension of the space spanned by the set of reduced basis functions is much smaller than the dimension of the original full order model. An offline–online computational decomposition is achieved in the reduced multiscale basis methods to significantly improve computation efficiency. Careful comparison is addressed for the reduced basis methods using different optimization strategies. A few numerical results are presented to illustrate the efficacy of the reduced basis methods.



## 1. Introduction

The inputs in many fundamental models have a wide range of length scales. The typical examples include the subsurface flow models in heterogeneous porous media and heat conduction models in composite materials. These model inputs often contain some uncertainties because of no enough knowledge about the physical properties and measurement noise. To get a computational model, the uncertainties in the model inputs are usually parameterized by a few random variables. Thus, these models can be described by parameterized PDEs (PPDEs). It is crucial to efficiently and accurately solving the associated PPDEs to predict the model's outputs and estimate the model's parameters. Both the multiple physical scales and the random parameters have great influence on the model. It may be computationally expensive to simulate these models in traditional methods. The interest in developing efficient multiscale methods and model reduction methods the PPDEs have been gotten much attention in recent years (Refs. [1–5]).

* Corresponding author.
*E-mail addresses:* ljjiang@hnu.edu.cn (L. Jiang), qiuqili@hnu.edu.cn (Q. Li).

http://dx.doi.org/10.1016/j.cam.2016.01.033
0377-0427/

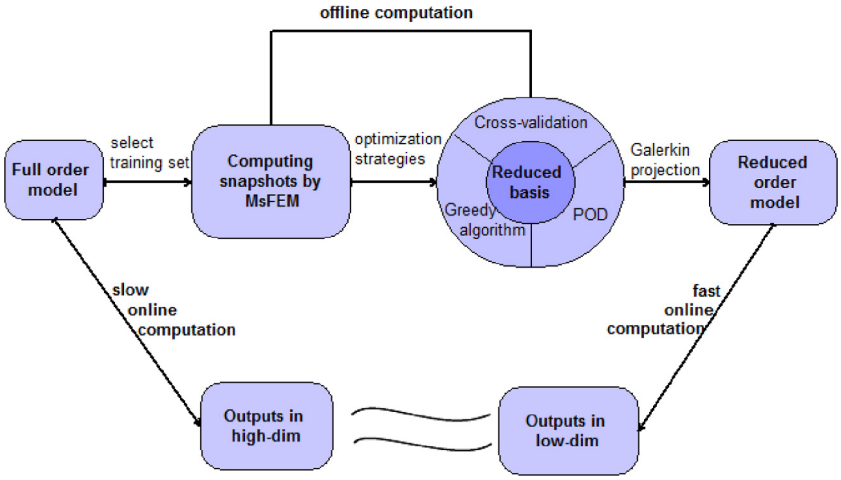


**Fig. 1.1.** Schema of reduced multiscale basis method.

There have many multiscale methods in the last decade (e.g., [6–9,2,10–13]). Multiscale Finite Element Method (MsFEM) [12] is one of the pioneer multiscale methods and many other multiscale share its similarity. The basic idea of MsFEM is to incorporate the small-scale information to multiscale basis functions and capture the impact of small-scale features on the coarse-scale through a variational formulation. One of the most important features for MsFEM is that the multiscale basis functions can be computed overhead and used repeatedly for the model with different source terms, boundary conditions and the coefficients with similar multiscale structures [10].

In this work, we focus on the multiscale basis functions in MsFEM and present reduced multiscale basis methods to solve parameterized elliptic PDEs with multiscale diffusion coefficients. Reduced basis (RB) method is one of model order reduction methods and has been used to solve PPDEs in a low-dimensional manifold [14–16,4,17,5]. The main idea of the RB method is to identify a small set of basis functions, which are constructed by a set of snapshots. The snapshots are the solutions of PPDEs corresponding to a set of parameter samples judiciously selected by some sampling strategies. The RB method depends on a projection onto the low-dimensional space, which is spanned by the small set of basis functions. It uses an offline–online computational decomposition to improve efficiency. In offline stage, snapshots are computed and reduced basis functions are generated. In online stage, a reduced model is solved for many instances of parameters, which are required to estimate the influence of the uncertainty.

For the RB method, we can use traditional finite element methods in a fine grid to get accurate snapshots. For multiscale models, the computation of the snapshots may be quite expensive because we have to use a very fine mesh to resolve all scales of the model for many snapshots. To this end, we use MsFEM on a coarse grid to compute snapshots and develop reduced multiscale basis methods. To get a set of optimal reduced basis functions from the snapshots, we use three strategies: cross-validation method, greedy algorithm and proper orthogonal decomposition (POD). In both cross-validation method and greedy algorithm, basis functions are generated by an incremental constructive manner. In cross-validation method, we search optimal snapshots or samples for basis construction such that an average error is minimal. The greedy algorithm shares the sophisticated concept from Kolmogorov $n$-width approximation [18]. It identifies the parameters for basis by worst approximation error of the best $n$-dimensional subspace. POD is to find a low rank approximation to the space spanned by snapshots. We use the three approaches to generate reduced basis. A reduced order model is constructed by projecting the full order model to the space of reduced basis. Fig. 1.1 describes the schema to get the reduced order model using the presented reduced basis method. A careful comparison is performed for the three approaches using FEM and MsFEM. We find that the cross validation method is a straightforward approach and leads to best approximation in the three approaches. In the RB methods, the offline–online computation decomposition is desirable when the model inputs (e.g., coefficients and sources) affinely depend on parameter variables. If the inputs are not affine with respect to the parameter variables, we utilize Empirical Interpolation Method (EIM) [19] to get an affine approximation for the inputs. In the reduced multiscale basis methods, the multiscale basis equations are solved in each coarse block independently. To improve the MsFEM approximation, we use some limited global information to construct MsFE basis functions, and this is particularly useful for the multiscale models in highly heterogeneous porous media [11,13].

The paper is structured as follows. In the next section, we present some preliminaries and notations for the paper. In Section 3, we present a general procedure for reduced basis methods and introduce a sampling strategy for RB methods: cross-validation method. Section 4 is devoted to presenting the reduced multiscale basis methods. In Section 5, we present three approaches used for constructing the optimal reduced basis: cross-validation, greedy algorithm and POD. The Empirical Interpolation Method is also presented in Section 5. In Section 6, some numerical examples are provided to illustrate the performance of the reduced multiscale basis methods. In the last section, we make some conclusions.

## 2. Preliminaries and notations

In this section, we present some preliminaries and notations for the rest of paper. Let $L^2(D)$ be the space of square integrable functions over $D$, and the usual Sobolev space $H_0^1(D) = \{v \in L^2(D), \nabla v \in [L^2(D)]^d, v|_{\partial D} = 0\}$. For a simplicity of notation, we denote $H_0^1(D)$ by $X$. It is well-known that the $L^2(D)$ is an inner product space with the inner product $(\cdot, \cdot)$, which induces the $L^2$-norm $\|\cdot\|$. Let $(\cdot, \cdot)_H$ be the inner product for $H_0^1(D)$ and induce the norm $\|\cdot\|_{1,D}^2 = (\cdot, \cdot)_H$.

In the paper, we consider parameterized elliptic PDEs. For simplicity of presentation, we assume that the parameterized elliptic PDEs can be formulated as the following weak form: find $u(\mu) \in X$ such that

$$a\big(u(\mu), v; \mu\big) = l(v; \mu), \quad \forall v \in X, \tag{2.1}$$

where $a(\cdot, \cdot; \mu) : X \times X \longrightarrow \mathbb{R}$ is a symmetric bilinear form for any parameter $\mu \in \Omega \subset \mathbb{R}^p$, and $l(\cdot; \mu)$ be a bounded linear functional over $X$ for any $\mu \in \Omega$. We define an energy inner product by

$$(w, v; \mu)_E := a(w, v; \mu), \quad \forall w, v \in X.$$

The energy norm by $\|w\|_E^2 = (w, w; \mu)_E$.

We consider the problem: given $\mu \in \Omega \subset \mathbb{R}^p$, evaluate the output $G(\mu)$ of the model (2.1), where $G(\mu) \in \mathbb{R}$ and is given by

$$G(\mu) = L\big(u(\mu)\big), \tag{2.2}$$

where $L$ is a bounded linear functional over $X$. For well-posedness of (2.1), we assume that $a(\cdot, \cdot; \mu)$ is continuous and coercive over $X$ for all $\mu \in \Omega$. Furthermore, we assume that both the parametric bilinear form $a(\cdot, \cdot; \mu)$ and the parametric linear form $l(\cdot; \mu)$ are affine with respect to $\mu$, i.e.,

$$\begin{cases} a(w, v; \mu) &= \displaystyle\sum_{i=1}^{m_a} k^i(\mu) a^i(w, v), \quad \forall w, v \in X, \ \forall \mu \in \Omega, \\ l(v; \mu) &= \displaystyle\sum_{i=1}^{m_l} f^i(\mu) l^i(v), \quad \forall v \in X, \ \forall \mu \in \Omega. \end{cases} \tag{2.3}$$

In the above, for $i = 1, \ldots, m_a$, each $k^i : \Omega \longrightarrow \mathbb{R}$ is a $\mu$-dependent function and each $a^i : X \times X \longrightarrow \mathbb{R}$ is a symmetric bilinear form independent of $\mu$. For $i = 1, \ldots, m_l$, each $f^i : \Omega \longrightarrow \mathbb{R}$ is a $\mu$-dependent function and each $l^i : X \longrightarrow \mathbb{R}$ is continuous functional independent of $\mu$. The affine assumption (2.3) is crucial to fulfill an offline–online computation decomposition for a many-query to model's outputs. When $a(\cdot, \cdot; \mu)$ and $l(\cdot; \mu)$ are not affine with regard to $\mu$, we can use an Empirical Interpolation Method (EIM) such that $a(\cdot, \cdot; \mu)$ and $l(\cdot; \mu)$ can be approximated by an affine representation. The EIM will be discussed in Section 5.4.

Let $\Xi_{train}$ be a training set, which is a collection of a finite number of samples in $\Omega$. Typically the training set is chosen by Monte Carlo methods. We require that the samples in $\Xi_{train}$ are sufficiently scattered in the domain $\Omega$. $|\Xi_{train}|$ denotes the cardinality of the set $\Xi_{train}$.

## 3. Reduced basis method

We follow the framework [20,17,5] to present a general reduced basis method in this section.

Let $X_h$ be a given finite element (FE) space in a fine grid and $\dim(X_h) = N_f$. Then the FE discretization of problem (2.2)–(2.1) is as follows: given any $\mu \in \Omega$, evaluate

$$G_h(\mu) = L\big(u_h(\mu)\big),$$

where $u_h(\mu) \in X_h$ satisfies

$$a\big(u_h(\mu), v; \mu\big) = l(v; \mu), \quad \forall v \in X_h. \tag{3.4}$$

We can now define an inner product and a norm for the space $X$, (refer to [5]) for a given $\bar{\mu}$ and a positive $\tau$ as follows,

$$(w, v)_X := (w, v; \bar{\mu})_E + \tau(w, v), \quad \forall w, v \in X,$$

and

$$\|w\|_X^2 := (w, w)_X, \quad \forall w \in X.$$

We can show the following norm equivalence between $\|\cdot\|_X$ and $\|\cdot\|_{1,D}$, i.e., there exist $\alpha_0 > 0$ and $\gamma_0 > 0$ such that

$$\alpha_0 \|w\|_{1,D} \le \|w\|_X \le \gamma_0 \|w\|_{1,D}.$$

Consequently, the bilinear form $a(\cdot,\cdot;\mu)$ is continuous and coercive in space $(X, \|\cdot\|_X)$ for all $\mu \in \Omega$, i.e., there exist $\gamma > 0$ and $\alpha > 0$ such that for any $\mu \in \Omega$,

$$a(w, v; \mu) \le \gamma(\mu)\|w\|_X\|v\|_X, \quad \forall w, v \in X, \ \forall \mu \in \Omega,$$
$$a(v, v; \mu) \ge \alpha(\mu)\|v\|_X^2, \quad \forall v \in X, \ \forall \mu \in \Omega. \tag{3.5}$$

Let $\{X_h^n\}_{n=1}^N$ be sequence of finite dimensional spaces satisfying

$$X_h^1 \subset X_h^2 \subset \cdots \subset X_h^N \subset X_h.$$

The RB method is devoted to approximating the solution $u(\mu)$ of the parameter dependent problem (2.1) by a few pre-computed solutions $u(\mu^n)$ of (3.4) for some selected parameter values $\mu^n$, $n = 1, \ldots, N$. Let

$$\mathcal{F} = \{u(\mu) \in X_h : \mu \in \Omega\}.$$

To assess the optimality of the approximation space $X_h^n$ with dimension $n$ ($n = 1, \ldots, N$), it is natural to compare this space with the best $n$-dimensional subspace spanned by elements of $\mathcal{F}$, which minimizes a projection error for $\mathcal{F}$ over all $n$-dimensional subspaces spanned by some elements of $\mathcal{F}$. The minimal error is given by the Kolmogorov width [14,18,21]

$$d_n(Y_n, \mathcal{F}) := \inf\{E(\mathcal{F}; Y_n) : Y_n \text{ is a } n\text{-dimensional subspace of } X\},$$

where $E(\mathcal{F}; Y_n)$ is the angle between $\mathcal{F}$ and $Y_n$. We construct a finite dimensional space, which is spanned by some elements of $\mathcal{F}$ with good approximation properties. The procedure is described as follows,

- $X_h^1 = \arg\min_{\substack{Y_1 \subset X \\ \dim Y_1 = 1}} d_n(Y_1, \mathcal{F})$,
- Assume that $X_h^{N-1}$ have been constructed, $X_h^N = \arg\min_{\substack{X_h^{N-1} \subset Y_N \subset X \\ \dim Y_N = N}} d_n(Y_N, \mathcal{F})$.

Thus we get a sequence of RB approximation spaces $\{X_h^n\}_{n=1}^N$ and a set of basis function $\{\varphi_n : 1 \le n \le N\}$. To obtain a set of $(\cdot,\cdot)_X$-orthonormal basis functions, we apply POD to the set $\{\varphi_n : 1 \le n \le N\}$ in the $(\cdot,\cdot)_X$ inner product. We denote the set of orthonormal basis functions by

$$\{\psi_i : 1 \le i \le N\},$$

which spans the same space as $span\{u(\mu^n) \in X_h : 1 \le n \le N\}$. If the support of each basis function $\psi_i$ ($i = 1, \ldots, N$) is the whole spatial domain $D$, then we call the RB method to be reduced global basis method.

### 3.1. Galerkin projection and offline–online computation

Now we consider the reduced order model and evaluate its output. Given $\mu \in \Omega$, we evaluate

$$G_h^N(\mu) = L(u_h^N(\mu)),$$

where $u_h^N(\mu) \in X_h^N \subset X_h$ satisfies

$$a(u_h^N(\mu), v; \mu) = l(v; \mu), \quad \forall v \in X_h^N. \tag{3.6}$$

Because $\{\psi_i\}_{i=1}^N$ is the set of basis functions for the reduced order model (3.6), the solution $u_h^N(\mu)$ can be represented by

$$u_h^N(\mu) = \sum_{i=1}^N u_i^N(\mu)\psi_i.$$

By plugging $v = \psi_j$ into (3.6), we have

$$\sum_{i=1}^N a(\psi_i, \psi_j; \mu)u_i^N(\mu) = l(\psi_j; \mu), \quad 1 \le i, j \le N. \tag{3.7}$$

Then we can evaluate output of the reduced order model by

$$G_h^N(\mu) = \sum_{i=1}^N u_i^N(\mu)L(\psi_i; \mu).$$

Eq. (3.7) implies a linear algebraic system with $N$ unknowns. The stiffness matrix and the load vector from Eq. (3.7) involve the computation of inner products with entities $\psi_i$, $1 \le i \le N$, each of which is represented by $N_f$ finite element

basis functions of $X_h$. This will lead to substantial computation for $u_h^N(\mu)$, and the marginal cost per input–output evaluation $\mu \longrightarrow G_h^N(\mu)$ is expensive. If the assumption (2.3) of affine decomposition holds, then Eq. (3.7) can be rewritten by

$$\sum_{i=1}^{N}\left(\sum_{q=1}^{m_a} k^q(\mu)a^q(\psi_i,\psi_j)u_i^N(\mu)\right)=\sum_{q=1}^{m_l} f^q(\mu)l^q(\psi_j),\quad 1\le j\le N. \tag{3.8}$$

This gives rise to the matrix form

$$\left(\sum_{q=1}^{m_a} k^q(\mu)\mathbf{A}_{Nh}^q\right)\mathbf{u}^N(\mu)=\sum_{q=1}^{m_f} f^q(\mu)\mathbf{F}_{Nh}^q, \tag{3.9}$$

where

$$(\mathbf{A}_{Nh}^q)_{ij}=a^q(\psi_i,\psi_j),\qquad (\mathbf{F}_{Nh}^q)_j=l^q(\psi_j),\quad 1\le i,j\le N.$$

Let $\{\xi_k\}_{k=1}^{N_f}$ be the FE basis of $X_h$. Because basis function $\psi_i$ belongs to the FE space $X_h$, it can be written as

$$\psi_i=\sum_{k=1}^{N_f} Z_{ik}\xi_k,\quad 1\le i\le N.$$

Let $(\mathcal{Z})_{ki}=Z_{ki}$, $1\le i\le N$. Thus $\mathcal{Z}\in\mathbb{R}^{N_f\times N}$, and we get

$$\mathbf{A}_{Nh}^q=\mathcal{Z}^T\mathcal{A}_{N_f}^q\mathcal{Z},\qquad \mathbf{F}_{Nh}^q=\mathcal{Z}^T\mathcal{F}_{N_f}^q,$$

where $(\mathcal{A}_{N_f}^q)_{ij}=a^q(\xi_j,\xi_i)$ and $(\mathcal{F}_{N_f}^q)_i=l(\xi_i)$. The matrixes $\mathbf{A}_{Nh}^q$ and the vectors $\mathbf{F}_{Nh}^q$ are independent of parameter $\mu$, and their computation is once and in offline phase. The online computation is to solve Eq. (3.9) for any $\mu\in\Omega$. Because the online computation only involves $N$ unknowns ($N\ll N_f$), this is efficient.

### 3.2. Cross-validation for sampling reduced basis

Greedy algorithm and POD have been utilized to find a set of optimal reduced basis functions [20,17,5]. In this section, we will introduce a cross-validation (CV) method to identify the samples from training set to construct reduced basis.

In the CV method, we use two of sample sets: training set $\Xi_{train}$ and validation set $\Xi_{validate}$. The set $\Xi_{train}$ is used to train the model, and the set $\Xi_{validate}$ is used to evaluate the performance of the reduced order model. We choose $|\Xi_{validate}|>|\Xi_{train}|$ and $\Xi_{validate}\supset\Xi_{train}$. $\varepsilon^*$ is a chosen tolerance for the stopping criterion of the sampling. The CV method is described in Algorithm 1 for the sampling reduced basis. The CV method is a straightforward approach. Our numerical experiments show that CV method gives the reduced surrogate model more accurate than reduced order model by greedy algorithm and POD.

If we use finite element methods in a fine grid to compute snapshots and use greedy algorithm, POD and CV method to identify optimal reduced basis, the support of the reduced basis functions is the whole spatial domain. Thus the resultant RB method is a reduced global basis method.

## 4. Reduced multiscale basis method

### 4.1. Multiscale FEM space

In the reduced basis method, we need to compute a set of snapshots $\{u(\mu^n)\}_{n=1}^N$ to construct reduced basis. These snapshots can be computed by traditional finite element methods in a fine grid. If the snapshots have strong multiscale features, then we have to use a very fine mesh to resolve the features in all scales. This computation may be very expensive. To overcome the difficulty, we can use multiscale finite element method (MsFEM) to compute the snapshots. In this section, we briefly present MsFEM and the reduced MsFEM.

Let $\mathcal{K}$ be a coarse partition of $D$ and the representative coarse element $K\in\mathcal{K}$. Let $\{x_i\}_{i=1}^{N_c}$ be the interior nodes of the coarse mesh $\mathcal{K}$. In each element $K\in\mathcal{K}$, we define a set of nodal basis $\{\phi_i^K\}_{i=1}^d$, where $d$ is the number of nodes of the coarse element. We usually use the leading order term of the differential operator to build multiscale basis equations. For simplicity, we still use the notation $a(\cdot,\cdot;\mu)$ to denote the bilinear functional associated with the leading order term of the elliptic PDE. Then the multiscale basis function $\phi_i^K$ with support $K$ solves the equation

$$\begin{cases} a(\phi_i^K(x,\mu),v;\mu) = l(v;\mu), & \forall v\in H_0^1(K),\\ \phi_i^K(x,\mu) = g_i(x,\bar{\mu}) & \text{on } \partial K, \end{cases} \tag{4.10}$$

where $\bar{\mu}$ is the mean of the random parameter $\mu$. In the multiscale basis equation, we use $g_i$ as limited global information [10, 11], which is obtained by linearly scaling the solution of (3.4) with parameter value $\bar{\mu}$. For the standard MsFEM introduced

**Input**: A train set $\Xi_{train} \subset \Omega$, a validating set $\Xi_{validate} \subset \Omega$, a tolerance $\varepsilon^*$

**Output**: $S_N$ *and* $X_h^N$

**Initialization**: for $\forall \mu \in \Xi_{train}$, $e^{1,\mu}(\mu_v) = u_h(\mu_v) - u_h^{1,\mu}(\mu_v)$,

-where $u_h^{1,\mu}(\mu_v)$ solves (3.6) $\forall v \in X_h^{1,\mu}$, and $X_h^{1,\mu} = span\{u_h(\mu)\}$;

$error_{mean}^1(\mu) = mean_{\mu_v \in \Xi_{validate}} \|e^{1,\mu}(\mu_v)\|_{L^2}$;

$\mu^1 = \arg\min_{\mu \in \Xi_{train}} error_{mean}^1(\mu)$;

$X_h^1 = span\{u_h(\mu^1)\},\ S_1 = \{\mu^1\}$;

$\varepsilon_1 = \max_{\mu \in \Xi_{train}} error_{mean}^1(\mu)$;

$\Xi_{train} = \Xi_{train} \setminus \mu^1$;

$N = 1$;

**while** $\varepsilon_N \leq \varepsilon^*$

  $N = N + 1$;

  for $\forall \mu \in \Xi_{train}$, $e^{N,\mu}(\mu_v) = u_h(\mu_v) - u_h^{N,\mu}(\mu_v)$;

  -where $u_h^{N,\mu}(\mu)$ solves (3.6) $\forall v \in X_h^{N,\mu}$, and $X_h^{N,\mu} = X_h^{N-1} \bigcup span\{u_h(\mu)\}$;

  $error_{mean}^N(\mu) = mean_{\mu_v \in \Xi_{validate}} \|e^{N,\mu}(\mu_v)\|_{L^2}$;

  $\mu^N = \arg\min_{\mu \in \Xi_{train}} error_{mean}^N(\mu)$;

  $X_h^N = X_h^{N-1} \bigcup span\{u_h(\mu^N)\},\ S_N = S_{N-1} \bigcup \{\mu^N\}$;

  $\varepsilon_N = \max_{\mu \in \Xi_{train}} error_{mean}^N(\mu)$;

  $\Xi_{train} = \Xi_{train} \setminus S_N$;

**end while**

Algorithm 1: The algorithm of cross-validation method for reduced global basis.

in [10], we take $l(v;\ \mu) = 0$ in the multiscale basis Eq. (4.10). We can straightforwardly extend the local basis functions $\{\phi_i^K\}$ to the whole domain $D$ and denote them by $\{\phi_i\}_{i=1}^{Nc}$, where '$i$' is a global subscript. Here the superscript $K$ will be suppressed for simplicity of notation.

The reduced multiscale basis method is devoted to approximating the solution $u(\mu)$ of the problem (2.1) by a set of pre-computed basis functions $\{\phi_i(\mu^n) : i = 1, \ldots, N_c,\ n = 1, \ldots, N\}$ for some selected parameter values $\{\mu^n\}_{n=1}^N$. Let $X_H^N$ be an $(N \times N_c)$-dimensional subspace of $X$. We define

$$\mathcal{M}(D) := \{\phi_i(\mu) \in X_h : 1 \leq i \leq N_c,\ \mu \in \Omega\}.$$

To assess approximation property, it is natural to compare the subspace $X_H^N$ with the best $N \times N_c$-dimensional subspace spanned by some elements of $\mathcal{M}(D)$, which minimizes the projection error for the $\mathcal{M}(D)$ over all $N \times N_c$-dimensional subspaces of X. This minimal error is given by the Kolmogorov width

$$d_{N\times N_c}(Y_{N\times N_c}, \mathcal{F}) := \inf\{E(\mathcal{F}; Y_{N\times N_c}) : Y_{N\times N_c} \text{ is an } N \times N_c\text{-dimensional subspace of } X\},$$

where $E(\mathcal{F}; Y_{N\times N_c})$ is the angle between $\mathcal{F}$ and $Y_{N\times N_c}$ under a metric. For example, $E(\mathcal{F}; Y_{N\times N_c}) := \sup_{u\in\mathcal{F}} \|u - P_{Y_{N\times N_c}} u\|_X$ for greedy algorithm. Here $P_{Y_{N\times N_c}}$ denotes a projection operator onto $Y_{N\times N_c}$. We construct a finite dimensional space, which is spanned by elements of $\mathcal{M}(D)$ with good approximation. The procedure is described as follows:

- $X_H^1 = \arg\min_{\substack{Y_{1\times N_c}\subset X \\ \dim Y_{1\times N_c} = N_c}} d_n(Y_{1\times N_c}, \mathcal{F})$,
- Assume that $X_H^{N-1}$ have been constructed, $X_H^N = \arg\min_{\substack{X_H^{N-1}\subset Y_{N\times N_c}\subset X \\ \dim Y_{N\times N_c} = N\times N_c}} d_n(Y_{N\times N_c}, \mathcal{F})$.

Thus we get a sequence of reduced multiscale approximation spaces:

$$X_H^1 \subset X_H^2 \subset, \ldots, \subset X_H^N \subset X_h,$$

and a set of basis function $\{\varphi_n^i : 1 \le i \le N_c, 1 \le n \le N\}$. To obtain a set of $(\cdot,\cdot)_X$-orthonormal basis functions, we apply POD to the set $\{\varphi_n^i : 1 \le i \le N_c, 1 \le n \le N\}$ in the $(\cdot,\cdot)_X$ inner product. We denote the set of orthonormal basis functions by:

$$\{\psi_i : 1 \le i \le N_c \times N\}, \tag{4.11}$$

which spans the same space as $span\{\phi_n^i : 1 \le i \le N_c, 1 \le n \le N\}$. We note that the support of each basis function $\psi_i$ in (4.11) is on a coarse block. So the reduced basis functions are local basis functions.

We can use the reduced multiscale basis method and Galerkin projection to get a reduced model. Let $u_H^N(\mu) \in X_H^N \subset X_h$ solve

$$a(u_H^N(\mu), v; \mu) = l(v; \mu), \quad \forall v \in X_H^N. \tag{4.12}$$

Given $\mu \in \Omega$, we evaluate

$$G_H^N(\mu) = L(u_H^N(\mu)).$$

Because $\{\psi_i\}_{i=1}^{N_c \times N}$ is the set of basis functions for the reduced order model (4.12), the solution $u_H^N(\mu)$ can be represented by

$$u_H^N(\mu) = \sum_{i=1}^{N_c \times N} u_i^N(\mu)\psi_i. \tag{4.13}$$

By plugging $v = \psi_j,\ 1 \le j \le N_c \times N$ into (4.12), we have

$$\sum_{i=1}^{N_c \times N} a(\psi_i, \psi_j; \mu)u_i^N(\mu) = l(\psi_j; \mu), \quad 1 \le i, j \le N_c \times N. \tag{4.14}$$

Then we can evaluate the output of the model by

$$G_H^N(\mu) = \sum_{i=1}^{N_c \times N} u_i^N(\mu)L(\psi_i; \mu).$$

### 4.2. Offline–online computation decomposition for reduced multiscale basis method

Eq. (4.14) implies a linear algebraic system with $N \times N_c$ unknowns. The stiffness matrix and the load vector from Eq. (4.14) involve the computation of inner products with entities $\psi_i,\ 1 \le i \le N \times N_c$, each of which is represented by $N_f$ finite element basis functions of $X_h$. This will lead to substantial computation for $u_h^N(\mu)$, and the input–output evaluation $\mu \longrightarrow G_H^N(\mu)$ is expensive. If the assumption (2.3) of affine decomposition holds, then Eq. (4.14) can be rewritten by

$$\sum_{i=1}^{N \times N_c} \left( \sum_{q=1}^{m_a} k^q(\mu)a^q(\psi_i, \psi_j)u_i^N(\mu) \right) = \sum_{q=1}^{m_l} f^q(\mu)l^q(\psi_j), \quad 1 \le j \le N \times N_c.$$

This gives rise to the matrix form

$$\left( \sum_{q=1}^{m_a} k^q(\mu)\mathbf{A}_N^q \right) \mathbf{u}_N(\mu) = \sum_{q=1}^{m_l} f^q(\mu)\mathbf{F}_N^q, \tag{4.15}$$

where

$$(\mathbf{A}_N^q)_{ij} = a^q(\psi_i, \psi_j), \qquad (\mathbf{F}_N^q)_j = l^q(\psi_j), \quad 1 \le i, j \le N \times N_c.$$

Because basis function $\psi_i$ belongs to the FE space $X_h$, it can be written as

$$\psi_i = \sum_{k=1}^{N_f} Z_{ik}\xi_k, \quad 1 \le i \le N \times N_c.$$

Let $(\mathcal{Z})_{ki} = Z_{ki},\ 1 \le N \le N$. Then $\mathcal{Z} \in \mathbb{R}^{N_f \times (N \times N_c)}$ and we can get

$$\mathbf{A}_N^q = \mathcal{Z}^T \mathcal{A}_{N_f}^q \mathcal{Z}, \qquad \mathbf{F}_N^q = \mathcal{Z}^T \mathcal{F}_{N_f}^q,$$

where $(\mathcal{A}_{N_f}^q)_{ij} = a^q(\xi_j, \xi_i)$, $(\mathcal{F}_{N_f}^q)_i = l(\xi_i)$. The matrixes $\mathcal{A}_{N_f}^q$ and the vectors $\mathcal{F}_{N_f}^q$ are independent of parameter $\mu$, and their computation is once and in offline phase. The online computation is to solve Eq. (4.15) for any $\mu \in \Omega$. The online computation only involves $N \times N_c$ unknowns, where $N \times N_c < N_f$.

We will present details of the three sampling strategies used for constructing the reduced multiscale basis: cross-validation, greedy algorithm and POD.

## 5. Strategies for constructing optimal reduced multiscale basis

### 5.1. Cross-validation method for sampling reduced multiscale basis

We can similarly extend the cross-validation method in Section 3.2 to the reduced multiscale basis method. The cross-validation is used to sample the parameters for construction of multiscale basis functions. We describe the details in Algorithm 2. Here $M_C$ is the number of parameter samples selected for basis.

**Input**: A training set $\Xi_{train} \subset \Omega$, a validating set $\Xi_{validate} \subset \Omega$, a tolerance $\varepsilon^*$

**Output**: $S_N$ *and* $X_H^{N,CV}$

**Initialization**: for $\forall \mu \in \Xi_{train}$, $e^{1,\mu}(\mu_v) = u_h(\mu_v) - u_H^{1,\mu}(\mu_v)$,

-where $u_H^{1,\mu}(\mu_v)$ solves (4.12) $\forall v \in X_H^{1,\mu}$, and $X_H^{1,\mu} = span\{\phi_i(\mu) : 1 \le i \le N_c\}$;

$error_{mean}^1(\mu) = mean_{\mu_v \in \Xi_{validate}} \|e^{1,\mu}(\mu_v)\|_{L^2}$;

$\mu^1 = \arg\min_{\mu \in \Xi_{train}} error_{mean}^1(\mu)$;

$X_H^{1,CV} = span\{\phi_i(\mu^1) : 1 \le i \le N_c\}$, $S_1 = \{\mu^1\}$;

$\varepsilon_1 = \max_{\mu \in \Xi_{train}} error_{mean}^1(\mu)$;

$\Xi_{train} = \Xi_{train} \setminus \mu^1$;

$N = 1$;

**while** $\varepsilon_N \le \varepsilon^*$

  $N = N + 1$;

  for $\forall \mu \in \Xi_{train}$, $e^{N,\mu}(\mu_v) = u_h(\mu_v) - u_H^{N,\mu}(\mu_v)$;

  -where $u_H^{N,\mu}(\mu)$ solves (4.12) $\forall v \in X_H^{N,\mu}$, and $X_H^{N,\mu} = X_H^{N-1} \bigcup span\{\phi_i(\mu) : 1 \le i \le N_c\}$;

  $error_{mean}^N(\mu) = mean_{\mu_v \in \Xi_{validate}} \|e^{N,\mu}(\mu_v)\|_{L^2}$;

  $\mu^N = \arg\min_{\mu \in \Xi_{train}} error_{mean}^N(\mu)$;

  $X_H^{N,CV} = X_H^{N-1} \bigcup span\{\phi_i(\mu^N) : 1 \le i \le N_c\}$;

  $S_N = S_{N-1} \bigcup \{\mu^N\}$;

  $\varepsilon_N = \max_{\mu \in \Xi_{train}} error_{mean}^N(\mu)$;

  $\Xi_{train} = \Xi_{train} \setminus S_N$;

**end while**

$M_C = N$;

Algorithm 2: Cross-validation for reduced multiscale basis.

Given $\Xi_{train}$, according to Algorithm 2, we get the set of parameter samples

$$S_{M_C} = \{\mu^1, \ldots, \mu^{M_C}\},$$

and the corresponding MsRB space

$$X_H^{M_C,CV} = span\{\phi_i(\mu^n) : 1 \le i \le N_c,\ 1 \le n \le M_C\}.$$

To get a set of $(\cdot,\cdot)_X$-orthonormal basis function, we apply POD to the set $\{\phi_i(\mu^n) \in X_h : 1 \le i \le N_c,\ 1 \le n \le M_C\}$ under the $(\cdot,\cdot)_X$ inner product. Then the set of orthogonal basis functions is denoted by

$$\{\psi_i : 1 \le i \le N_c \times M_C\}.$$

### 5.2. Greedy algorithm for sampling reduced multiscale basis

In order to find a few optimal parameters for reduced basis and to assure the fidelity of the reduced multiscale model to approximate the original model, we can use a greedy algorithm to obtain reduce multiscale basis functions. To this end, we need to make the posteriori error bounds for MsFEM. This will involve two basic ingredients of the error bounds: MsFEM residual error and stability information of the corresponding bilinear form [17,5].

First we consider the residual error for MsFEM. This is important for posteriori analysis. By Eq. (4.12), we get

$$a\big(u_h(\mu) - u_H^N(\mu) + u_H^N(\mu), v; \mu\big) = l(v; \mu), \quad \forall v \in X_h,$$

that is,

$$a\big(u_h(\mu) - u_H^N(\mu), v; \mu\big) = l(v; \mu) - a\big(u_H^N(\mu), v; \mu\big), \quad \forall v \in X_h.$$

Let the error $e(\mu) := u_h(\mu) - u_H^N(\mu)$, and $r(v; \mu) \in X_h^*$ (the dual space of $X_h$) be the residual

$$r(v; \mu) := l(v; \mu) - a\big(u_H^N(\mu), v; \mu\big), \quad \forall v \in X_h.$$

Then we get

$$a\big(e(\mu), v; \mu\big) = r(v; \mu), \quad \forall v \in X_h. \tag{5.16}$$

By Riesz representation theory, there exists a function $\hat{e}(\mu) \in X_h$ such that

$$\big(\hat{e}(\mu), v\big)_X = r(v; \mu), \quad \forall v \in X_h. \tag{5.17}$$

Then we can rewrite the error residual equation (5.16) as

$$a\big(e(\mu), v; \mu\big) = \big(\hat{e}(\mu), v\big)_X, \quad \forall v \in X_h.$$

Consequently, the dual norm of the residual $r(v; \mu)$ can be evaluated through the Riesz representation,

$$\|r(v; \mu)\|_{X_h^*} := \sup_{v \in X_h} \frac{r(v; \mu)}{\|v\|_X} = \|\hat{e}(\mu)\|_X. \tag{5.18}$$

The computation of the residual is crucial to perform the offline–online computation decomposition.

Secondly, we need a positive, parametric lower bound function $\alpha_{LB}^{N_f}(\mu)$ for $\alpha^{N_f}(\mu)$ in (3.5), where $\alpha^{N_f}(\mu)$ is the FE coercivity constant and can be defined by

$$\alpha^{N_f}(\mu) := \inf_{w \in X_h} \frac{a(w, w; \mu)}{\|w\|_X^2}.$$

Here the function $\alpha_{LB}^{N_f} : \Omega \to \mathbb{R}$, possesses the two properties: (i) $0 < \alpha_{LB}^{N_f}(\mu) \le \alpha^{N_f}(\mu)$ for any $\mu \in \Omega$; (ii) the online computation to evaluate $\alpha_{LB}^{N_f}(\mu)$ is independent of $N_f$. An efficient method for computing $\alpha_{LB}^{N_f}(\mu)$ is the Successive Constraint Method (Refs. [20,5]).

We define an error estimator [22,5] for the solution of Eq. (5.16) in the energy norm as

$$\Delta_N(\mu) := \frac{\|\hat{e}(\mu)\|_X}{(\alpha_{LB}^{N_f})^{1/2}}.$$

We define the error estimator by

$$\eta_N(\mu) := \Delta_N(\mu) / \|u_h(\mu) - u_H^N(\mu)\|_E.$$

It measures the quality of the posteriori error. Following the proof in the Refs. [17,5], we can show that for any $N = 1, \ldots, M_G$, the bound of the posteriori error satisfies

$$1 \le \eta_N(\mu) \le \sqrt{\frac{\gamma(\mu)}{\alpha_{LB}^{N_f}(\mu)}}, \quad \forall \mu \in \Omega,$$

where $M_G$ is the number of local basis functions at each coarse node. We note that the $\hat{e}(\mu)$ is related to $r(v; \mu)$ by Eq. (5.18). By (4.13) and (2.3), the residual can be expressed by

$$\begin{aligned} r(v; \mu) &= l(v; \mu) - a(u_H^N(\mu), v; \mu) \\ &= l(v; \mu) - \sum_{i=1}^{N \times N_c} u_i^N(\mu) a(\psi_i, v; \mu) \\ &= \sum_{q=1}^{m_l} f^q(\mu) l^q(v) - \sum_{i=1}^{N \times N_c} u_i^N(\mu) \sum_{p=1}^{m_a} k^p(\mu) a^p(\psi_i, v). \end{aligned} \tag{5.19}$$

By (5.19) and (5.17), we have

$$(\hat{e}(\mu), v)_X = \sum_{q=1}^{m_l} f^q(\mu) l^q(v) - \sum_{i=1}^{N\times N_c}\sum_{p=1}^{m_a} u_i^N(\mu) k^p(\mu) a^p(\psi_i, v).$$

This implies that

$$\hat{e}(\mu) = \sum_{q=1}^{m_l} f^q(\mu)\mathcal{C}_q + \sum_{i=1}^{N\times N_c}\sum_{p=1}^{m_a} u_i^N(\mu) k^p(\mu)\mathcal{L}_i^p, \tag{5.20}$$

where $\mathcal{C}_q$ is the Riesz representation of $l^q$, i.e., $(\mathcal{C}_q, v)_X = l^q(v)$ for any $v \in X_h$, $1 \le q \le m_l$. Similarly, $\mathcal{L}_i^p$ is the Riesz representation of $a^p(\psi_i, v)$, i.e., $(\mathcal{L}_i^p, v)_X = -a^p(\psi_i, v)$ for any $v \in X_h$, where $1 \le i \le N \times N_c$ and $1 \le p \le m_a$. We note that these $\mathcal{C}_q$ and $\mathcal{L}_i^p$ are the finite element solutions of elliptic equations. Eq. (5.20) gives rise to

$$\begin{aligned}\|\hat{e}(\mu)\|_X^2 = &\sum_{q=1}^{m_l}\sum_{q'=1}^{m_l} f^q(\mu) f^{q'}(\mu)(\mathcal{C}_q, \mathcal{C}_{q'})_X + \sum_{i=1}^{N\times N_c}\sum_{p=1}^{m_a} u_i^N(\mu) k^p(\mu)\\ &\times \left\{ 2\sum_{q=1}^{m_l} f^q(\mu)(\mathcal{C}_q, \mathcal{L}_i^p)_X + \sum_{i'=1}^{N\times N_c}\sum_{p'=1}^{m_a} u_{i'}^N(\mu) k^{p'}(\mu)(\mathcal{L}_i^p, \mathcal{L}_{i'}^{p'})_X \right\}.\end{aligned} \tag{5.21}$$

Combining (5.21) and (5.18) gives the calculation of the dual norm of the residual $\|r(v;\mu)\|_{X_h^*}$.

To efficiently compute $\|\hat{e}(\mu)\|_X$, we apply an offline–online procedure for the dual norm of the residual $r(v;\mu)$. In the offline stage we compute and store the parameter-independent quantities. In particular, we compute $\mathcal{C}_q$ and $\mathcal{L}_i^p$, where $1 \le i \le N\times N_c, 1 \le q \le m_l$ and $1 \le p \le m_a$. We store $(\mathcal{C}_q, \mathcal{C}_{q'})_X, (\mathcal{C}_q, \mathcal{L}_i^p)_X, (\mathcal{L}_i^p, \mathcal{L}_{i'}^{p'})$, where $1 \le i, i' \le N\times N_c,\ 1 \le q, q' \le m_l$, $1 \le p, p' \le m_a$. Thus the offline computation depends on $N \times N_c$, $m_l$, $m_a$ and $N_f$. In the online stage, for any $\mu$, we compute $u_i^N(\mu)$ $(1 \le i \le N \times N_c)$ and use (5.21) to compute $\|\hat{e}(\mu)\|_X$. The online computation is independent of the fine-scale degree of freedom $N_f$.

Next we describe the greedy algorithm for the reduced multiscale basis method. Let $\Xi_{train}$ be a given training set, and $\varepsilon^*$ a tolerance for the stopping criterion for the greedy algorithm. Then the greedy algorithm is described in Algorithm 3.

In Algorithm 3, $\{\psi_i : 1 \le i \le N^G\}$ are the first $N^G$ basis functions by performing POD for $\{\phi_i(\mu^n) \in X_h : 1 \le i \le N_c,\ 1 \le n \le N\}$. Here $N^G < N_c \times N$ and its value is determined by the tolerance $\varepsilon^{POD}$. Consequently, the reduced multiscale finite element space by the greedy algorithm is defined by

$$X_H^{M_G, Greedy} = \{\psi_i : 1 \le i \le N_c \times M_G\}.$$

### 5.3. POD method for construction of reduced multiscale basis

POD is used to construct a low rank approximation for a Hilbert space (Ref. [23]). In the case of matrix approximation, POD is the same as Singular Value Decomposition (SVD).

We firstly present a short review for POD. Let $\{y_1, y_2, \ldots, y_{n_t}\} \subset X$ be a given set of snapshots and $\mathcal{Y}_0 := span\{y_1, y_2, \ldots, y_{n_t}\}$ with $r = \dim(\mathcal{Y}_0)$. Suppose that $\{\Upsilon_i\}_{i=1}^r$ is a set of orthonormal basis for $\mathcal{Y}_0$. Then each snapshot $y_j$ can be written as

$$y_j = \sum_{i=1}^{r}(y_j, \Upsilon_i)_X \Upsilon_i, \quad j = 1, \ldots, n_t.$$

POD provides a way to find a set of orthonormal basis $\{\Upsilon_i\}_{i=1}^d$ such that $\tilde{y}_j := \sum_{i=1}^d (y_j, \Upsilon_i)_X \Upsilon_i$ and $\tilde{y}_j$ is the optimal approximation to $y_j$ for all $j = 1, \ldots, n_t$ in the sense of mean square error. The set $\{\Upsilon_i\}_{i=1}^d$ is called a POD basis with dimension $d$. The POD basis can be computed from a given set of snapshots.

Now we discuss using POD for reduced multiscale basis functions. For the POD method, the snapshots are the solutions of (4.10) for a few samples. Let $\Xi_s$ be the set of samples for snapshots and $|\Xi_s| = n_t$. We note that the set $\Xi_s$ plays the same role as the training set in the cross-validation method and the greedy algorithm described as before.

For each $1 \le i \le N_c$, we consider a set of snapshots: $\{\phi_i(\mu) : \mu \in \Xi_s\}$, each of which can be written as

$$\phi_i(\mu^n) = \sum_{k=1}^{N_f} y_{ki}^n \xi_k = \vec{\xi}\vec{y}_i^n, \quad 1 \le n \le n_t,$$

where $\vec{\xi} = [\xi_1, \ldots, \xi_{N_f}]_{1\times N_f}$ is the set of FE basis functions and $\vec{y}_i^n = [y_{1i}^n, \ldots, y_{N_f i}^n]' \in \mathbb{R}^{N_f}$. The matrix of the snapshot coefficients is defined by

$$Y_i = (y_{ki}^n) = [\vec{y}_i^1 \cdots \vec{y}_i^n \cdots \vec{y}_i^{n_t}] \in \mathbb{R}^{N_f \times n_t}, \quad 1 \le n \le n_t,\ 1 \le k \le N_f.$$

Based on the snapshots, the POD basis functions can be constructed as follows.

**Input**: A training set $\Xi_{train} \subset \Omega$, a tolerance $\varepsilon^*$

**Output**: $S_N$ *and* $X_H^{N,Greedy}$

**Initialization**: $\mu^1 = \arg\max_{\mu\in\Xi_{train}} \|u_h(\mu)\|_{1,D}$, $S_1 = \{\mu^1\}$;
compute $\{\phi_i(\mu^1)\}_{i=1}^{N_c}$;
$X_H^{1,Greedy} = span\{\phi_i(\mu^1) : 1 \le i \le N_c\}$;
$\Xi_{train} = \Xi_{train} \setminus \mu^1$;
$\varepsilon_1 = \max_{\mu\in\Xi_{train}} \Delta_1(\mu)$;
$N = 1$;
**while** $\varepsilon_N \le \varepsilon^*$
  $N = N + 1$;
  $\mu^N = \arg\max_{\mu\in\Xi_{train}} \Delta_{N-1}(\mu)$;
  $S_N = S_{N-1} \bigcup \{\mu^N\}$;
  compute $\{\phi_i(\mu^N)\}_{i=1}^{N_c}$;
  $X_H^{N,Greedy} = X_H^{N-1,Greedy} \bigcup span\{\phi_i(\mu^N) : 1 \le i \le N_c\} = span\{\psi_i : 1 \le i \le N^{greedy}\}$;
  $\Xi_{train} = \Xi_{train} \setminus S_N$;
  $\varepsilon_N = \max_{\mu\in\Xi_{train}} \Delta_N(\mu)$;
**end while**
$M_G = N$;

Algorithm 3: Greedy algorithm for reduced multiscale basis.

- Construct a matrix $\aleph$ using the inner product of the snapshots, i.e.,

$$\begin{aligned}\aleph &= ((\phi_i(\mu^m), \phi_i(\mu^n))_X)\\ &= \left(\sum_{k,k'=1}^{N_f} y_{ki}^m y_{k'i}^n (\xi_k, \xi_{k'})_X\right)\\ &= ([y_{1i}^m, \ldots, y_{N_f i}^m] M_h [y_{1i}^n, \ldots, y_{N_f i}^n]')\\ &= Y_i' M_h Y_i \in \mathbb{R}^{n_t \times n_t},\end{aligned}$$

  where $M_h = [(\xi_k, \xi_{k'})_X] \in \mathbb{R}^{N_f \times N_f}$ is the Gram matrix of FE basis functions.
- Compute $M_P$ eigenvectors of $\aleph$ corresponding the first $M_P$ largest eigenvalues, i.e.,

$$\aleph v_j = v_j \lambda_j, \quad j = 1, \ldots, M_P,$$

  where $v_j \in \mathbb{R}^{n_t}$ and $\lambda_1 \ge \lambda_2, \ldots, \lambda_{M_P}$.
- Given any $1 \le i \le N_c$, the POD basis functions $\varphi_i^j$ are given by

$$\varphi_i^j(x) = \frac{1}{\sqrt{\lambda_j}} \sum_{n=1}^{n_t} (v_j)_n \phi_i(\mu^n), \quad j = 1, \ldots, M_P.$$

- Write each POD basis function $\varphi_i^j$ in terms of FE basis $\vec{\xi} = [\xi_1, \ldots, \xi_{N_f}]$. Let $\aleph V = V\Lambda$ be the eigenvalue decomposition of $\aleph$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_{n_t})$ and $\lambda_1 \ge \cdots \ge \lambda_{n_t} \ge 0$. Define $V_i^{M_P} = [v_1 \cdots v_{M_P}] \in \mathbb{R}^{n_t \times M_P}$ to be the first $M_P$ columns of $V$ and $\Lambda_i^{M_P} = diag(\lambda_1, \lambda_2, \ldots, \lambda_{M_P}) \in \mathbb{R}^{M_P \times M_P}$. Then for given $1 \le i \le N_c$, the set of POD basis functions $\Phi_i(x) := [\varphi_i^1, \ldots, \varphi_i^{M_P}]$ can be written by

$$\Phi_i(x) = \vec{\xi} Y_i V_i^{M_P} (\Lambda_i^{M_P})^{-\frac{1}{2}}.$$

Given $M_P$ and $\Xi_s$, we use POD and get the reduced multiscale basis functions as follows

$$\{\varphi_i^n : 1 \le i \le N_c,\ 1 \le n \le M_P\}.$$

They are mutually orthonormal under the inner product $(\cdot,\cdot)_X$. Then the multiscale finite element space is defined by

$$X_H^{M_P,POD} := span\{\varphi_i^n : 1 \le i \le N_c,\ 1 \le n \le M_P\}.$$

### 5.4. Empirical interpolation method

To achieve the offline–online computation decomposition, we need to use the affine assumption (2.3) for the parametric bilinear form $a(\cdot,\cdot;\mu)$ and the parametric linear form $l(\cdot;\mu)$. If they are not affine with respect to the parameter $\mu$, then we can use Empirical Interpolation Method (EIM) to get an affine decomposition.

Now we use the function $l : D \times \bar{\Omega} \longrightarrow \mathbb{R}$ as an example to present EIM. When $l$ is not an affine form, we use EIM approximation $l_M \approx l$ such that

$$l_M(x,\mu) = \sum_{m=1}^{M} \varphi_m(\mu) q_m(x),$$

where $\varphi_m : \bar{\Omega} \longrightarrow \mathbb{R}$, and $q_m : D \longrightarrow \mathbb{R}$. We define a parametrical manifold

$$\mathcal{M}_l = \{l(x,\mu) : \mu \in \Omega\}.$$

We want to choose suitable functions $\varphi_m(\mu)$ and $q_m(x)$ to approximate any element in $\mathcal{M}_l$. The EIM approximation space $W_M$ has the property

$$W_M = span\{l(x,\mu_1),\ldots,l(x,\mu_M)\} = span\{q_1(x),\ldots,q_M(x)\}.$$

Then for any $\mu \in \Omega$, the EIM approximation $l_M(x,\mu)$ of $l(x,\mu)$ is given by

$$l_M(x,\mu) = \sum_{m=1}^{M} \varphi_m(\mu) q_m(x) \in W_M. \tag{5.22}$$

Let $\{x_1,\ldots,x_M\}$ be a set of interpolation nodes in $D$. Then for any given $\mu \in \Omega$, the coefficients $\varphi_1(\mu),\ldots,\varphi_M(\mu)$ in (5.22) are computed by solving the following algebraic system

$$l_M(x_n,\mu) := \sum_{m=1}^{M} \varphi_m(\mu) q_m(x_n) = l(x_n,\mu), \quad 1 \le n \le M.$$

By the procedure, it is crucial to choose suitable parameter values $\mu_1,\ldots,\mu_M \in \Omega$ and interpolation nodes $x_1,\ldots,x_M \in D$. We can use a greedy algorithm to find them.

Let the EIM approximation error $e_M(x,\mu) = l(x,\mu) - l_M(x,\mu)$. Following the Ref. [19], the EIM algorithm is described in Algorithm 4. We note that the posteriori error estimation of EIM can be found in [24].

## 6. Numerical results

In this section, we present a few numerical examples to illustrate the applicability of the proposed reduced multiscale basis methods for solving parameterized elliptic partial differential equations. In Section 6.1, we consider three examples to illustrate performance of the different reduced basis methods for elliptic PDEs with one-dimensional parameters. In Section 6.2, we consider an example with reduced multiscale basis methods for high dimensional parameters. We compare the results of the presented reduced multiscale finite element basis methods and discuss advantages and disadvantages.

Let $k(x,\mu) : D \times \Omega \longrightarrow \mathbb{R}$ be a diffusion coefficient function. We consider the following model elliptic equation for numerical computation,

$$\begin{cases} -\mathrm{div}\big(k(x,\mu)\nabla u(x,\mu)\big) = f(x,\mu) & \text{in } D \times \Omega, \\ u(x,\mu) = 0 & \text{on } \partial D. \end{cases} \tag{6.23}$$

### 6.1. Numerical results for one-dimensional parameters

In this subsection, we will consider three numerical examples. Before presenting the individual examples, we describe the computational domain, that is, spatial domain $D = (1,2)^2$, and parameter $\Omega = (1,3)$.

**Input**: A training set $\Xi_{train} \subset \Omega$, a tolerance $\varepsilon^*$ and a functional $l(x, \mu)$

**Output**: $S_M$, $X_M$ *and* $W_M$

**Initialization**: *pick* $\mu_1 \in \Xi_{train}$;

$x_1 = \arg\sup_{x\in D} |l(x, \mu_1)|, \quad q_1 = \frac{l(\cdot,\mu_1)}{l(x_1,\mu_1)}, \quad W_1 = span\{q_1\}$;

$X_1 = \{x_1\}, \quad S_1 = \mu_1, \quad W_1 = span\{q_1\}$;

$\Xi_{train} = \Xi_{train} \setminus \mu^1$;

**for** $M = 1 : M_{max} - 1$

$\mu_{M+1} = \arg\max_{\mu\in\Xi_{train}} \|e_M(x, \mu)\|_{L^\infty(D)}$;

$x_{M+1} = \arg\sup_{x\in D} |e_M(x, \mu_{M+1})|$;

$q_{M+1} = \frac{e_M(\cdot;\mu_{M+1})}{e_M(x_{M+1};\mu_{M+1})}$;

$X_{M+1} = X_M \bigcup \{x_{M+1}\}, \quad S_{M+1} = S_M \bigcup \mu_{M+1}$;

$W_{M+1} = W_M \bigcup span\{q_{M+1}\}$;

$\Xi_{train} = \Xi_{train} \setminus S_N$;

$\varepsilon_{M+1} = \|e_M(x, \mu_{M+1})\|_{L^\infty(D)}$;

**if** $\varepsilon_{M+1} < \varepsilon^*$

$M_{max} = M$;

**end if**

**end for**

Algorithm 4: Empirical interpolation method.

*6.1.1. Numerical example I: reduced global basis model*

In this example, we consider the elliptic equation (6.23), where the coefficient function $k(x, \mu)$ and source function $f(x, \mu)$ are defined by

$$k(x, \mu) = (\sin(50\mu) + 1)x_2 + \exp(\mu/1000)x_1x_2, \qquad f(x, \mu) = x_1 \sin^2(x_2 + \mu).$$

Here $x := (x_1, x_2) \in D$. The function $k(x, \mu)$ is affine with respect to the parameter $\mu$, and $f(x, \mu)$ is not affine with respect to the parameter $\mu$. We apply EIM for $f(x, \mu)$ to achieve offline–online computation decomposition. For the discretization of the spatial domain, we use $50 \times 50$ uniform grid, where the reference solution is computed. Thus the number of degree of freedom $N_f = 2401$ for FEM. We choose 101 parameter values for offline computation, i.e., the number of snapshots $n_{train} = 101$. We use the bilinear finite element method in fine mesh to compute the snapshots. Then we use cross-validation method, greedy algorithm and POD to find the reduced global basis functions. The resultant model is a reduced global basis model.

To compare the approximation accuracy of the three reduced global basis methods, we randomly choose 50 samples from $\Omega$ and compute the average of relative $L^2$ error and average of relative $H^1$ error for the three methods. In Fig. 6.2, we depict the average of relative $L^2$ error (left figure) and relative $H^1$ error (right figure) versus the number of global basis functions for the three methods. From this numerical result, we see that the approximation errors decays fast when the number of basis functions increases at beginning. We also find that cross-validation reduced global basis method and POD method give better approximation than greedy method when the number of basis functions is relatively small.

*6.1.2. Numerical results comparison by reduced global basis model and reduced multiscale basis model*

In this example, we compare the numerical results by using reduced global basis method and reduced multiscale basis (local) method. To this end, we consider the elliptic equation (6.23) with the following oscillating coefficient and source term,

$$k(x, \mu) = \left(1 + 0.8 \sin\big(12\pi(x_1 - x_2 + \mu)\big)\right), \qquad f(x, \mu) = x_1 \sin^2(x_2 + \mu).$$

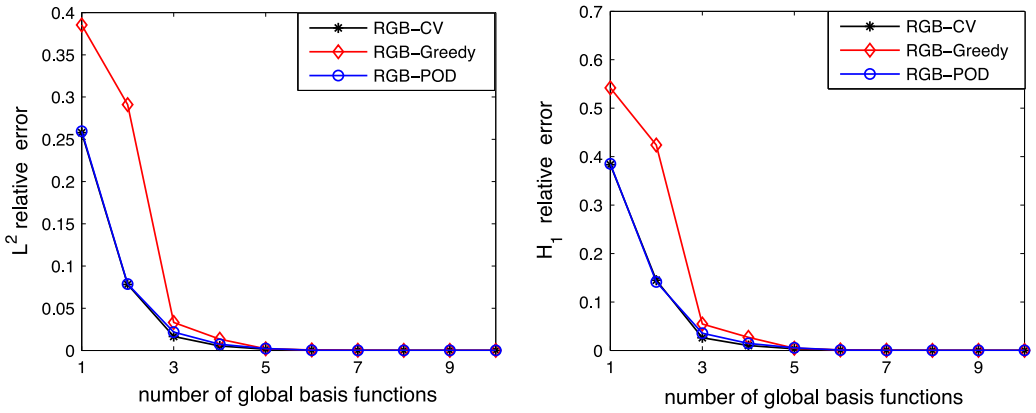


**Fig. 6.2.** The average of relative errors versus number of global basis functions for cross-validation global RB model (RGB-CV), greedy global RB model (RGB-Greedy) and POD-RB model (RGB–POD).

**Table 1**
Average relative $L^2$ error for RGB-CV, RGB-Greedy, RMsB-CV and RMsB-Greedy, $40 \times 40$ fine grid, $8 \times 8$ coarse grid, $N_s$ is the number of parameter samples selected for basis.

| $N_s$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| RGB-CV | 0.4731 | 0.3248 | 0.2086 | 0.1700 | 0.1336 | 0.1088 | 0.0747 | 0.0644 |
| RGB-Greedy | 0.5663 | 0.3610 | 0.2988 | 0.2779 | 0.2680 | 0.1255 | 0.1167 | 0.0879 |
| RMsB-CV | 0.3819 | 0.1853 | 0.1191 | 0.0786 | 0.0594 | 0.0527 | 0.0436 | 0.0358 |
| RMsB-Greedy | 0.4442 | 0.2212 | 0.1237 | 0.0953 | 0.0737 | 0.0549 | 0.0464 | 0.0414 |

**Table 2**
Average relative $H^1$ error in $L^2$ for RGB-CV, RGB-Greedy, RMsB-CV and RMsB-Greedy, $40 \times 40$ fine grid, $8 \times 8$ coarse grid, $N_s$ is the number of parameter samples selected for basis.

| $N_s$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| RGB-CV | 0.5665 | 0.4502 | 0.2967 | 0.2537 | 0.1951 | 0.1680 | 0.1265 | 0.1100 |
| RGB-Greedy | 0.6286 | 0.4644 | 0.4159 | 0.3872 | 0.3793 | 0.1921 | 0.1767 | 0.1429 |
| RMsB-CV | 0.4204 | 0.2608 | 0.1905 | 0.1473 | 0.1307 | 0.1238 | 0.1165 | 0.1069 |
| RMsB-Greedy | 0.4690 | 0.2803 | 0.1916 | 0.1654 | 0.1448 | 0.1293 | 0.1220 | 0.1166 |

Here $\mu \sim U(1, 3)$, i.e., uniform distribution in $(1, 3)$. The coefficient function $k(x, \mu)$ is oscillating with respect to the spatial variable $x$ and the random parameter $\mu$. This may give the implication that multiscale method may achieve better approximation than the global basis method. Both functions $k(x, \mu)$ and $f(x, \mu)$ are not affine with respect to the parameter $\mu$, we utilize EIM to get affine approximations for $k(x, \mu)$ and $f(x, \mu)$. We make $40 \times 40$ fine grid for computing reference solution. So the number of degrees of freedom $N_f = 1521$ for fine-scale FEM. The MsFEM computation is perform on the coarse grid of $8 \times 8$. For offline computation, 51 parameter values are selected for snapshots, i.e., $n_{train} = 51$. For the numerical example, we consider the four different model reduction methods: reduced global basis method using CV (RGB-CV), reduced global basis method using greedy algorithm (RGB-Greedy), reduced multiscale basis method using CV (RMsB-CV), reduced multiscale basis method using greedy algorithm (RMsB-Greedy).

To assess the approximation, we randomly choose 40 samples from the parameter space, which are used to compute the average relative errors. In Tables 1 and 2, we list the number of parameter samples for reduced basis versus the relative errors in $L^2$ and $H^1$, respectively, for the four methods: RGB-CV, RGB-Greedy, RMsB-CV and RMsB-Greedy. Fig. 6.3 illustrates the average relative $L^2$ error (left) and $H^1$ error (right) with respect to the sample number in $S_N$ for the four different model reduction methods. From the figure, we have three observations: (1) multiscale model reduction approach gives better accuracy than global model reduction approach in this example; (2) CV approach gives a little bit better approximation than greedy algorithm approach in both global basis reduction and multiscale basis reduction; (3) the error of RGB-CV decays more gradually than the error of RGB-Greedy as reduced basis functions enrich.

By the numerical example and our other numerical tests, we find that enriched local basis and global–local approach may achieve better accuracy than pure global basis method when the diffusion coefficients and source terms of the parameterized PDEs are highly oscillating with respect to both spatial variable and parameter variable, which are non-separable.

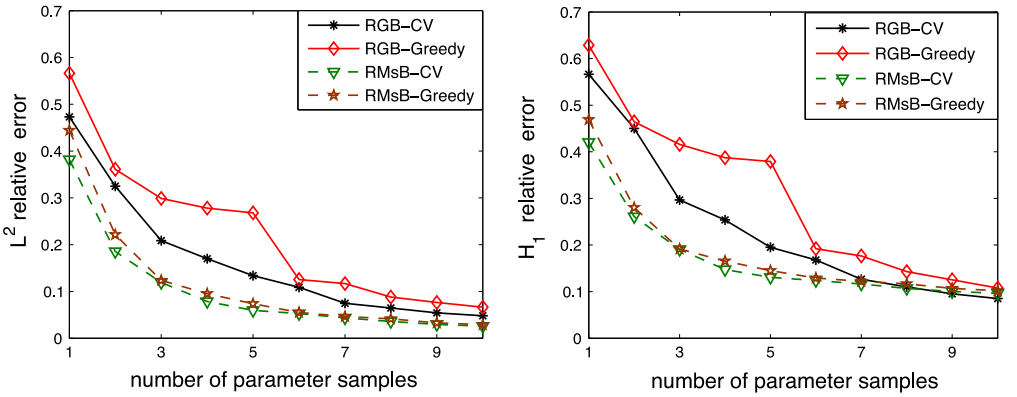


**Fig. 6.3.** The average relative errors versus number of samples for $S_N$ for the four model reduction methods: RGB-CV, RGB-Greedy, RMsB-CV and RMsB-Greedy. $40 \times 40$ fine grid and $8 \times 8$ coarse grid.

**Table 3**
The average relative $L^2$ error for RMsB-CV, RMsB-Greedy and RMsB-POD.

| ♯ of local basis functions per node | RMsB-CV | | RMsB-Greedy | | RMsB-POD | |
|---|---|---|---|---|---|---|
| | $5 \times 5$ | $10 \times 10$ | $5 \times 5$ | $10 \times 10$ | $5 \times 5$ | $10 \times 10$ |
| 2 | 0.0432 | 0.0132 | 0.0467 | 0.0254 | 0.0473 | 0.1270 |
| 4 | 0.0185 | 0.0058 | 0.0235 | 0.0125 | 0.0394 | 0.0114 |
| 6 | 0.0135 | 0.0037 | 0.0179 | 0.0044 | 0.0199 | 0.0070 |
| 8 | 0.0120 | 0.0025 | 0.0132 | 0.0038 | 0.0153 | 0.0043 |

### 6.1.3. *Numerical result with reduced multiscale basis model*

In this numerical example, we consider the elliptic equation (6.23), whose diffusion coefficient $k(x, \mu)$ and source term $f(x, \mu)$ are defined, respectively, by

$$k(x, \mu) = \sin^2(50\mu)x_1^2 x_2 + \frac{1}{\Big(3 + 2.8\sin\big(25\pi(x_1 - x_2)\big)\Big)(\mu^2 + 1)}, \qquad f(x, \mu) = \frac{1}{\sqrt{\mu x_1 + x_2^2}}.$$

Here the diffusion coefficient $k(x, \mu)$ is oscillating with respect to $x$, and is affine with respect to $\mu$. Because $f(x, \mu)$ is not affine about the parameter $\mu$, we use EIM to get affine approximation for $f(x, \mu)$. For the partition of spatial domain, we use $80 \times 80$ fine grid for reference solution. Hence, the number of degree of freedom $N_f = 6241$ for FEM in fine-scale. In offline computation phase, we choose 41 parameter samples for snapshots, i.e., $n_{train} = 41$, and use MsFEM to compute the snapshots. Then we apply cross-validation method, greedy algorithm and POD to reduce the original model, and get the three reduced order model: RMsB-CV, RMsB-Greedy and RMsB-POD.

To evaluate the approximation for the model reduction methods, we randomly choose 40 parameter samples and compute the average relative $L^2$ and $H^1$ errors. To discuss the effect of coarse grids, we consider two different coarse grids in the example: $5 \times 5$ and $10 \times 10$. The relative $L^2$ error is listed in Table 3 and the relative $H^1$ error is listed in Table 4. From the two tables, we find that (1) the approximation is improved as the coarse grid is refined for the three multiscale model reduction methods; (2) as the number of local basis functions increases, the errors become smaller; (3) RMsB-CV usually gives better approximation than the other two model reduction approaches. To visualize the individual errors of the 40 samples, we plot the relative errors for the three multiscale model reduction methods in Fig. 6.4 ($L^2$ error) and Fig. 6.5 ($H^1$ error). The two figures show that the error of RMsB-CV is not sensitive to the parameter samples compared with RMsB-Greedy and RMsB-POD. Fig. 6.6 shows the solution profile of the example for the four methods: FEM on fine grid (reference solution), RMsB-CV, RMsB-Greedy and RMsB-POD, where MsFEM is performed on $5 \times 5$ coarse grid with 5 multiscale basis functions on each coarse node. The figure shows all the solution profiles looks the same and the multiscale model reduction methods provide accurate approximation to the original full order model. Fig. 6.7 depicts the average relative errors versus the number of local basis functions for the three reduced multiscale method. From the figure, we see RMsB-CV achieves the best approximation.

## 6.2. *Numerical results for reduced multiscale basis model with a high-dimensional parameter*

In this section, we consider the elliptic equation (6.23), whose source term $f(x) = 150$, and $x \in (0, 1)^2$. The diffusion coefficient $k(x, \mu)$ is a random field, which is characterized by a two point exponential covariance function cov[$k$], i.e.,

**Table 4**
The average relative $H^1$ error for RMsB-CV, RMsB-Greedy and RMsB-POD.

| ♯ of local basis functions per node | RMsB-CV | | RMsB-Greedy | | RMsB-POD | |
|---|---|---|---|---|---|---|
| | $5 \times 5$ | $10 \times 10$ | $5 \times 5$ | $10 \times 10$ | $5 \times 5$ | $10 \times 10$ |
| 2 | 0.1855 | 0.0845 | 0.1629 | 0.0814 | 0.1836 | 0.1615 |
| 4 | 0.0671 | 0.0292 | 0.0777 | 0.0388 | 0.1704 | 0.0814 |
| 6 | 0.0501 | 0.0217 | 0.0640 | 0.0255 | 0.0794 | 0.0458 |
| 8 | 0.0444 | 0.0125 | 0.0464 | 0.0222 | 0.0594 | 0.0260 |

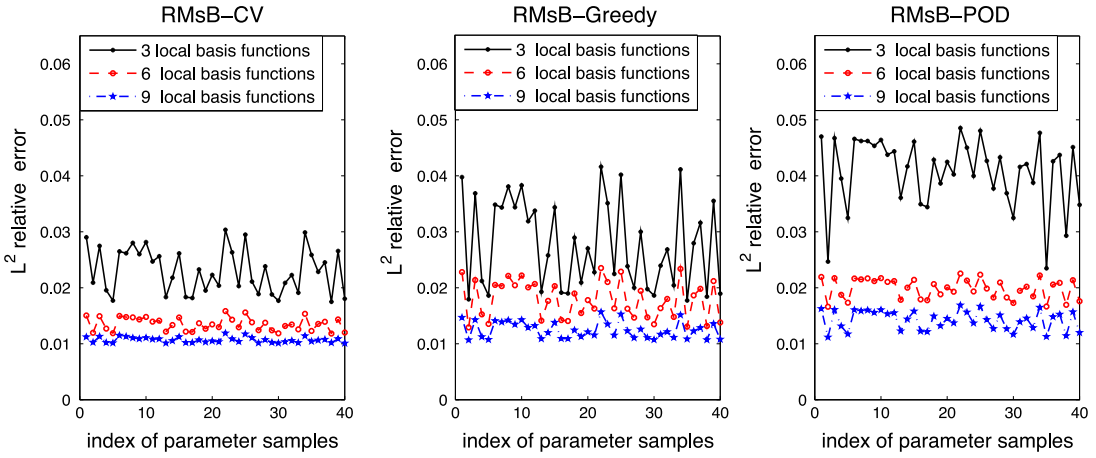


**Fig. 6.4.** The relative $L^2$ error by RMsB-CV, RMsB-Greedy and RMsB-POD, $80 \times 80$ fine grid, and $5 \times 5$ coarse grid.

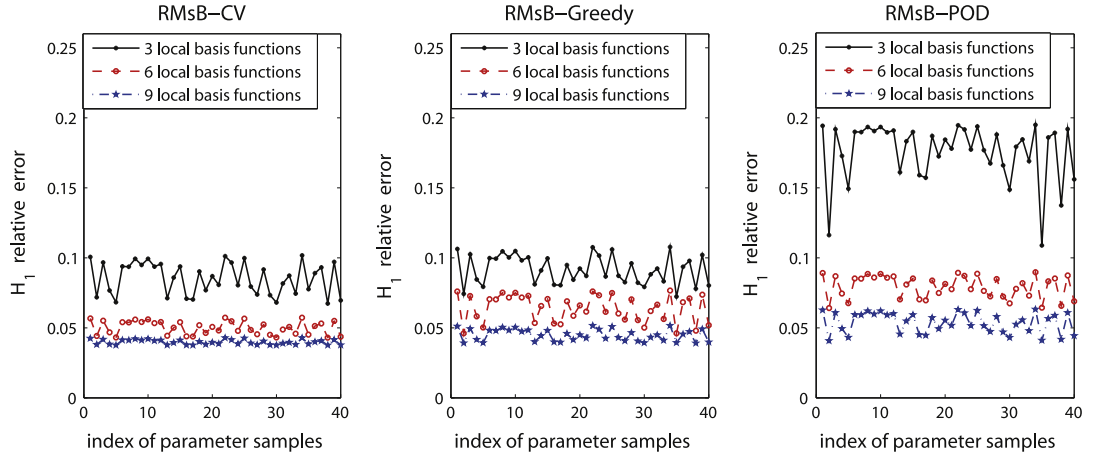


**Fig. 6.5.** The relative $H^1$ error by RMsB-CV, RMsB-Greedy and RMsB-POD, $80 \times 80$ fine grid, $5 \times 5$ coarse grid.

$$\mathrm{cov}[k](x_1, y_1; x_2, y_2) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|^2}{2l_x^2} - \frac{|y_1 - y_2|^2}{2l_y^2}\right),$$

where $(x_i, y_i)$ $(i = 1, 2)$ is the spatial coordinate in 2D. Here the variance $\sigma^2 = 1$, correlation length $l_x = l_y = 0.2$. The random coefficient $k(x, \mu)$ is obtained by truncated by a Karhunen–Loève expansion, i.e.,

$$k(x, \mu) := E[k] + \sum_{i=1}^{8} \sqrt{\gamma_i} b_i(x) \mu_i. \tag{6.24}$$

Here $E[k] = 8$ and the random vector $\mu := (\mu_1, \mu_2, \ldots, \mu_8) \in \mathbb{R}^8$. Each $\mu_i$ $(i = 1, \ldots, 8)$ is uniformly distributed in the interval $(0, 1)$.

We use $60 \times 60$ fine grid to compute the reference solution to the example. For MsFEM, we use $6 \times 6$ coarse grid. 51 parameter samples are selected for snapshots to get multiscale basis functions, i.e., $n_{train} = 51$. We apply RMsB-CV model, RMsB-Greedy model and RMsB-POD model to simulate the numerical example.

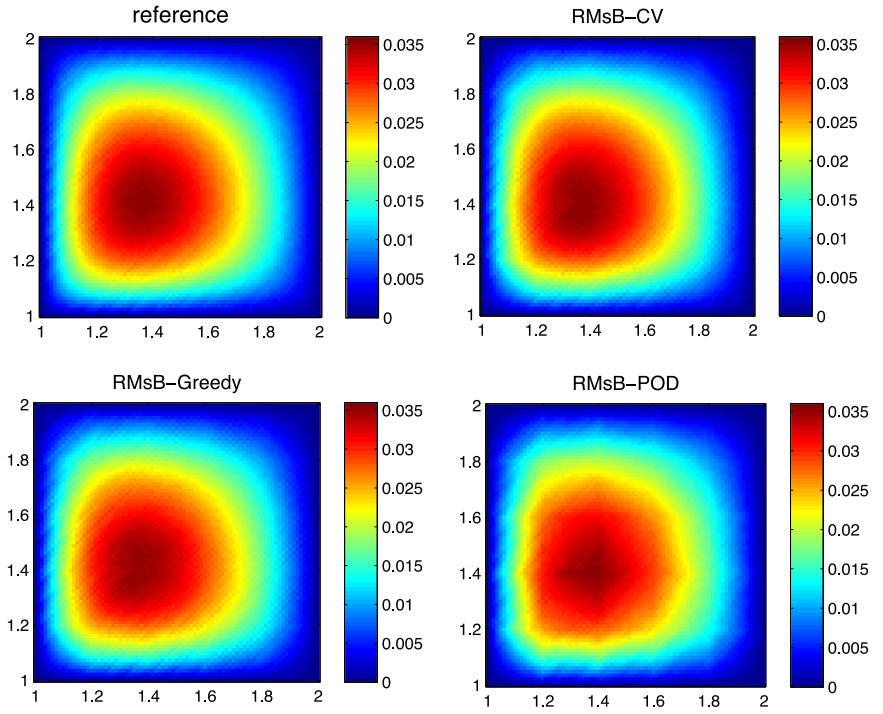


**Fig. 6.6.** Numerical solutions for the four methods (from left to right): FEM on fine grid, RMsB-CV, RMsB-Greedy and RMsB-POD, 80 × 80 fine grid, 5 × 5 coarse grid.

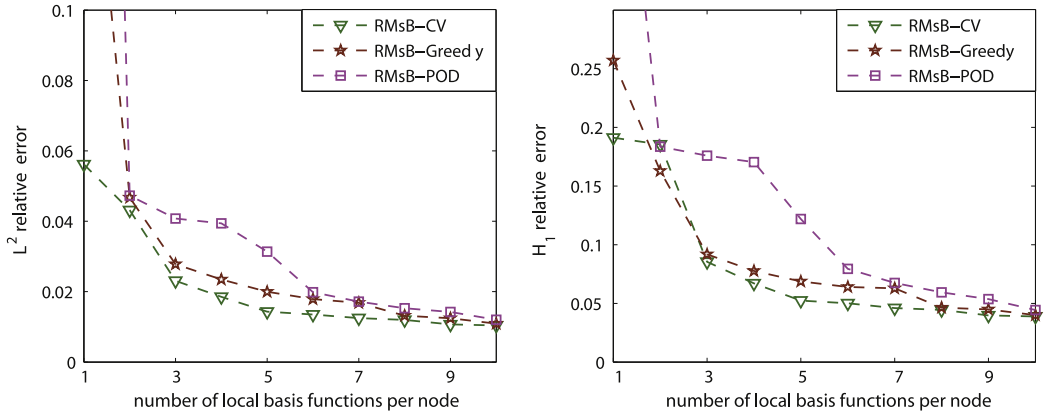


**Fig. 6.7.** The average relative error versus number of local basis functions for RMsB-CV, RMsB-Greedy and RMsB-POD, 5 × 5 coarse grid.

To compare the approximation, we randomly choose 50 samples and compute the average relative $L^2$ error and $H^1$ error for the three reduced multiscale basis methods. Fig. 6.8 depicts the average relative errors versus the number of local basis functions. The figure shows: (1) cross-validation gives the best approximation results; (2) as the number of local multiscale basis functions increase, cross-validation method can more steadily enhance accuracy than other two approaches.

To visualize the statistics properties, we plot the mean profile of the solutions in Fig. 6.9 and the variance profile of the solutions in Fig. 6.10, where we use 2 local multiscale basis functions at each node. From the two figures, we find that RMsB-CV gives better approximation for mean and variance of reference solution than RMsB-Greedy and RMsB-POD when we use a small number of nodal multiscale basis functions, and RMsB-Greedy shares the similar approximation to RMsB-POD.

## 7. Conclusions

In the paper, we have presented reduced basis methods for solving elliptic equations with parameterized inputs. If we solve the equations directly for a many-query situation, the computation efficiency will be bad. In order to significantly improve computation efficiency, we build a reduced order model by using reduced basis methods. To this end, we choose

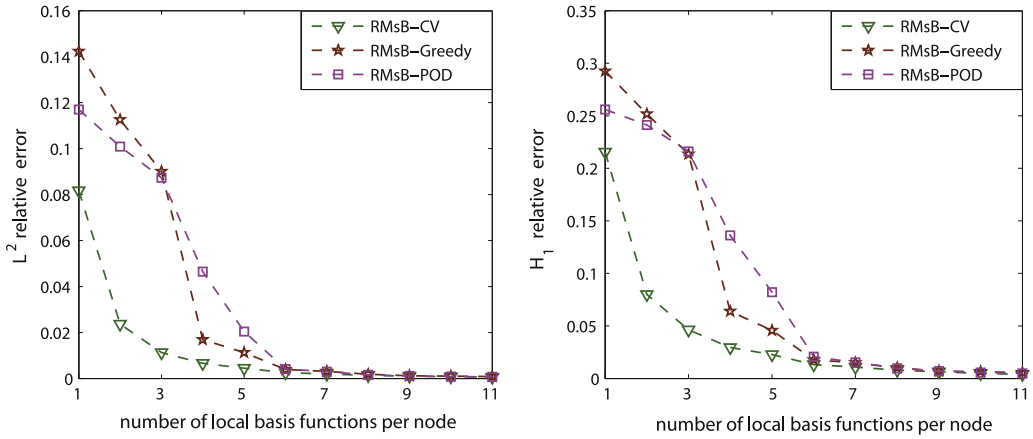


**Fig. 6.8.** The average relative errors for RMsB-CV, RMsB-Greedy and RMsB-POD, $6 \times 6$ coarse grid.

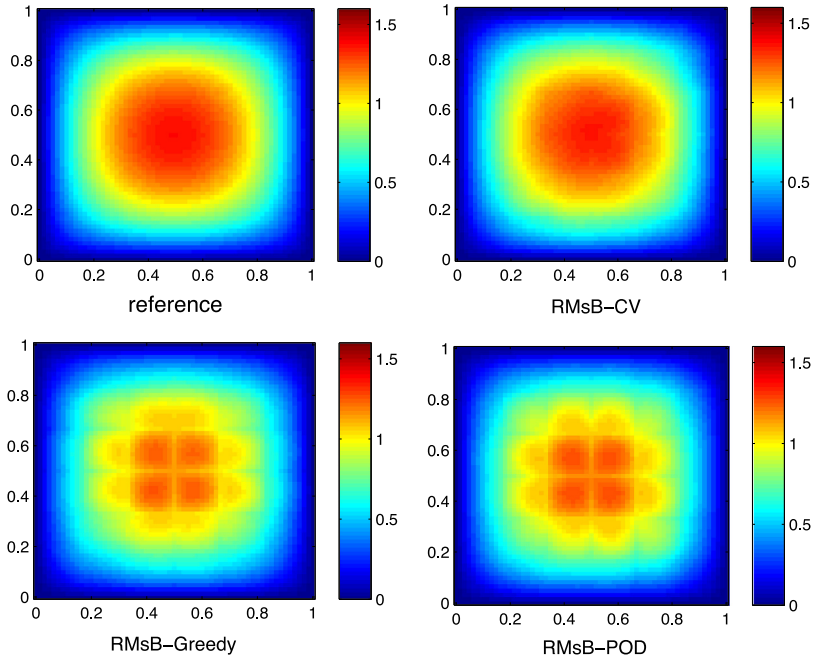


**Fig. 6.9.** The mean of solutions, $60 \times 60$ fine grid, $6 \times 6$ coarse grid, 2 local multiscale basis functions on each node.

a few snapshots sufficiently scattered in the manifold of the solutions. Then we use some optimization strategies to obtain reduced basis functions and get a reduced order model. Three optimization strategies have been presented in the paper: cross-validation method, greedy algorithm and POD. For the reduced basis methods, the whole computation admits offline–online decomposition. Although the offline computation may be expensive, the online computation is very efficient. This is very desirable for predicting the model's outputs for various parameter values and stochastic influence. We have carefully compared the performance of the three strategies for reduced basis, and found that cross-validation method may be more effective and stable to get a reduced model than greedy algorithm and POD when using a small number optimal parameters to construct reduced basis functions. If the elliptic equations have multiscale structures in inputs, we can use MsFEM to compute the snapshots in a coarse grid and get reduced multiscale basis model through the optimization strategies. Because the number of snapshots is usually large, using MsFEM can substantially improve the computation of snapshots. If the inputs (coefficients and source terms) of the elliptic equations are oscillating with respect to both spatial variable and parameter variable which are non-separable, we have found that the reduced multiscale basis methods may provide more accurate surrogate reduced order model than the reduced global basis methods. A few numerical results showed that the reduced multiscale basis method using cross-validation is effective to reduce basis functions for multiscale models.

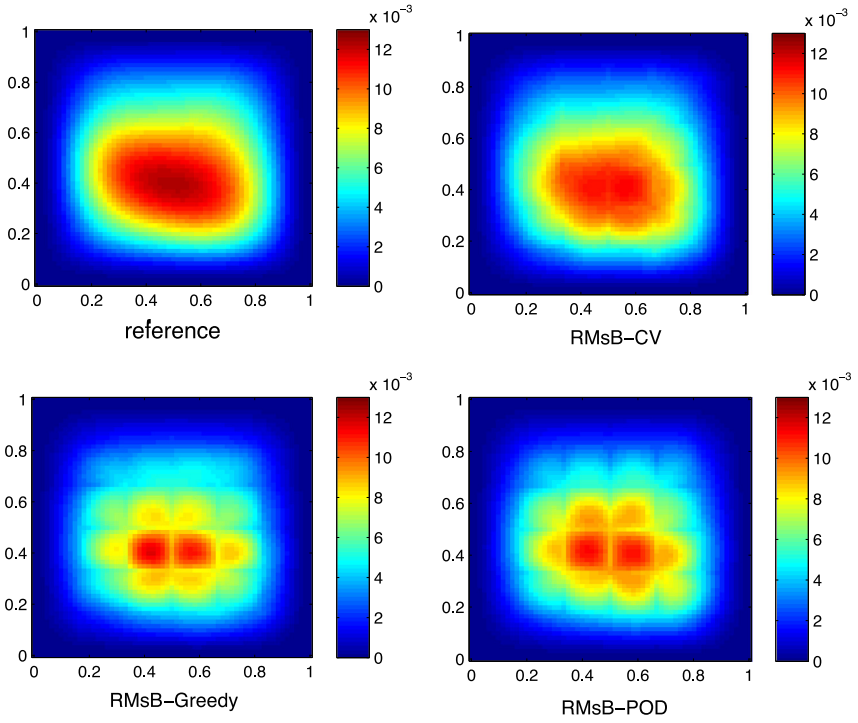


**Fig. 6.10.** The variance of solutions, 60 × 60 fine grid, 6 × 6 coarse grid, 2 multiscale basis functions on each node.

## Acknowledgment

We acknowledge the support of Chinese NSF 11471107.

## References

[1] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (2011) 395–404.
[2] P. Dostert, Y. Efendiev, T. Hou, Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification, Comput. Methods Appl. Mech. Engrg. 197 (2008) 3445–3455.
[3] L. Jiang, M. Presho, A resourceful splitting technique with applications to deterministic and stochastic multiscale finite element methods, Multiscale Model. Simul. 10 (2012) 954–985.
[4] N.C. Nguyen, A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales, J. Comput. Phys. 227 (2008) 9807–9822.
[5] 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 (2008) 229–275.
[6] G. Allaire, R. Brizzi, A multiscale finite element method for numerical homogennization, Multiscale Model. Simul. 4 (2005) 790–812.
[7] E.T. Chung, Y. Efendiev, Reduced-contrast approximations for high-contrast multiscale flow problems, Multiscale Model. Simul. 8 (2010) 1128–1153.
[8] T. Arbogast, G. Pencheva, M.F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, Multiscale Model. Simul. 6 (2007) 319–346.
[9] W.E. Weinan, B. Engquist, The heterogeneous multi-scale methods, Commun. Math. Sci. 1 (2003) 87–133.
[10] Y. Efendiev, T. Hou, Multiscale Finite Element Methods: Theory and Applications, Springer, 2009.
[11] Y. Efendiev, V. Ginting, T. Hou, R. Ewing, Accurate multiscale finite element methods for two-phase flow simulations, J. Comput. Phys. 220 (2006) 155–174.
[12] T.Y. Hou, X.H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput. Phys. 134 (1997) 169–189.
[13] L. Jiang, Y. Efendiev, V. Ginting, Multiscale methods for parabolic equations with continuum spatial scales, Discrete Contin. Dyn. Syst. Ser. B 8 (2007) 833–859.
[14] A. Buffa, Y. Maday, A.T. Patera, C. Prud'homme, G. Turinici, A priori convergence of the greedy algorithm for the parametrized reduced basis method, ESAIM Math. Model. Numer. Anal. 46 (2012) 595–603.
[15] Y. Chen, J.S. Hesthaven, Y. Maday, J. Rodriguez, Certified reduced basis methods and output bounds for the harmonic Maxwell's equations, SIAM J. Sci. Comput. 32 (2010) 970–996.
[16] M.A. Grepl, M. Karcher, Reduced basis a posteriori error bounds for parametrized linear quadratic elliptic optimal control problems, C. R. Math. 349 (2011) 873–877.
[17] A. Quarteroni, G. Rozza, A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, J. Math. Ind. 1 (2011) 1–49.
[18] G.G. Lorentz, M. von Golitschek, Y. Makovoz, Constructive Approximation: Advanced Problems, Springer-Verlag, New York, 1996.
[19] Barrault, Y. Maday, N.C. Nguyen, A.T. Patern, An "empirical interpolation" method: Application to efficient reduced-basis discretization of partial differentail equations, C. R. Math. Acad. Sci. Paris 339 (2004) 667–672.
[20] 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. Math. 345 (2007) 473–478.
[21] J.M. Melenk, On *n*-widths for elliptic problems, J. Math. Anal. Appl. 247 (2000) 272–289.

[22] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: reduced-basis output bounds methods, J. Fluids Eng. 124 (2002) 70–80.
[23] W.H. Schilders, Henk A. van der Vorst, J. Rommes, Model Order Reduction: Theory, Research Aspects and Applications, Springer, 2008.
[24] J. Eftang, M. Grepl, A. Patera, A posteriori error bounds for the empirical interpolation method, C. R. Math. 348 (2010) 575–579.