MULTISCALE MODEL. SIMUL.
Vol. 14, No. 1, pp. 482–501

RANDOMIZED OVERSAMPLING FOR GENERALIZED MULTISCALE FINITE ELEMENT METHODS*

VICTOR M. CALO†, YALCHIN EFENDIEV‡, JUAN GALVIS§, AND GUANGLIAN LI¶

Abstract. In this paper, we develop efficient multiscale methods for flows in heterogeneous media. We use the generalized multiscale finite element (GMsFEM) framework. GMsFEM approximates the solution space locally using a few multiscale basis functions. This approximation selects an appropriate snapshot space and a local spectral decomposition, e.g., the use of oversampled regions, in order to achieve an efficient model reduction. However, the successful construction of snapshot spaces may be costly if too many local problems need to be solved in order to obtain these spaces. We use a moderate quantity of local solutions (or snapshot vectors) with random boundary conditions on oversampled regions with zero forcing to deliver an efficient methodology. Motivated by the randomized algorithm presented in [P. G. Martinsson, V. Rokhlin, and M. Tygert, *A Randomized Algorithm for the approximation of Matrices*, YALEU/DCS/TR-1361, Yale University, 2006], we consider a snapshot space which consists of harmonic extensions of random boundary conditions defined in a domain larger than the target region. Furthermore, we perform an eigenvalue decomposition in this small space. We study the application of randomized sampling for GMsFEM in conjunction with adaptivity, where local multiscale spaces are adaptively enriched. Convergence analysis is provided. We present representative numerical results to validate the method proposed.

Key words. generalized multiscale finite element method, oversampling, high contrast, randomized approximation, snapshot spaces construction

AMS subject classifications. 65N30, 65N15

DOI. 10.1137/140988826

NOTATION.

κ	permeability field
ω	coarse neighborhood
K	coarse block
N_c	number of coarse nodes
χ	partition of unity function
$k_{\text{nb}}^{\omega_i}$	number of bases selected on ω_i
$p_{\text{nb}}^{\omega_i}$	buffer number on ω_i
η_i	local error indicator on ω_i
$\lambda_{l_i}^{\omega}$	the l_ith smallest eigenvalue for some spectral problem on ω_i
V_{off}	offline space

*Received by the editors September 25, 2014; accepted for publication (in revised form) December 22, 2015; published electronically March 23, 2016. This publication also was made possible by a National Priorities Research Program grant NPRP grant 7-1482-1278 from the Qatar National Research Fund (a member of The Qatar Foundation).

http://www.siam.org/journals/mms/14-1/98882.html

†Center for Numerical Porous Media (NumPor), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia, and Applied Geology Department, Western Australian School of Mines, Faculty of Science and Engineering, Curtin University, Perth, WA, Australia (Victor.Calo@Curtin.edu.au).

‡Center for Numerical Porous Media (NumPor), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia, and Department of Mathematics & Institute for Scientific Computation (ISC), Texas A&M University, College Station, TX (efendiev@math.tamu.edu).

§Departamento de Matemáticas, Universidad Nacional de Colombia, Bogotá D.C., Colombia (jugal4621@gmail).

¶Institute for Numerical Simulation, University of Bonn, 53115 Bonn, Germany (lotusli0707@gmail.com).

1. Introduction. Model reduction is becoming increasingly important when dealing efficiently with problems characterized by multiple scales. Due to scale disparity, single-scale discretization techniques cannot provide useful results with acceptable computational cost in practice. In order to efficiently handle these multiscale problems, many model reduction techniques have been developed in the literature. These include approaches that are based on homogenization and numerical homogenization [10, 28, 24, 18], approaches that employ finite element or finite volume basis functions to approximate the fine-scale features of the solution space [1, 2, 4, 5, 6, 7, 17, 21, 23, 25, 26], and approaches that employ global model reduction techniques [20, 11, 9]. In this paper, we focus on approaches based on multiscale finite element methods which fall into the second category. We use a recently introduced framework known as the generalized multiscale finite element method (GMsFEM) and discuss how one can reduce the setup cost employing randomized singular value decomposition (SVD) concepts [27, 22].

To construct multiscale basis functions, we employ the GMsFEM framework where the multiscale basis functions are constructed via a local spectral decomposition of a snapshot space. This snapshot space typically consists of spatial fields that represent the solution space up to some desired accuracy. For example, one choice for the snapshot space is to use harmonic functions that can represent any boundary value in each coarse region [13]. These snapshots are constructed by solving local problems for all possible Dirichlet boundary conditions. The latter allows us to represent any fine-scale traces and incorporate the effects of many small-scale features into these snapshots and thus achieve low-dimensional coarse models. However, the computation of these snapshots is expensive. In this paper, we propose the use of random boundary conditions in constructing snapshot vectors. We show that by using only a few of these randomly generated snapshots, we can adequately approximate the dominant modes of the solution space. To avoid oscillations near the boundary, an oversampling technique is used [13]. More precisely, we solve local problems on domains that are larger than the target coarse blocks. Typically, they are larger by several layers of fine-grid blocks around the target coarse block. Furthermore, we perform a local spectral decomposition using the restriction of the randomly generated snapshots to the target coarse-grid domain.

The use of random boundary conditions (to generate the snapshot spaces) is motivated by the randomized SVD methodology [27, 22]. In general, randomized SVD algorithms allow computing dominant eigenvectors by considering a random linear combination of the columns (or rows) of a given matrix. The random linear combinations typically have a component in the dominant modes and thus by performing a spectral decomposition in the span of these random combinations, we can achieve an accurate approximation of dominant eigenvectors.

We take advantage of the idea of randomized linear combinations to considerably reduce the computational cost associated with the computation of snapshot vectors. In particular, we propose solving local problems with random boundary conditions and perform the local spectral decomposition in the space of these snapshots. The cost reduction is due to the fact that, in previous approaches, the snapshot spaces were constructed by solving local problems for every possible boundary condition in each coarse region. Using our new methodology, the number of snapshots to be generated is only slightly larger than the number of desired eigenvectors. Our experience suggests that for GMsFEM modeling, in general it suffices to include four additional random boundary conditions to the number of eigenvectors sought. For instance, in our numerical experiments, when three basis functions per coarse grid are needed, we

compute only seven snapshot vectors (i.e., only seven random boundary conditions are generated, irrespective of the number of degrees of freedom on the coarse block boundary). We discuss how the number of additional snapshots can depend on the eigenvalue structure for some special cases. This new methodology provides substantial computational savings in the offline stage as we compute many fewer snapshots. We apply randomized boundary conditions on the oversampled region to avoid oscillations near the boundaries. Indeed, if random boundary conditions are imposed on the target coarse grid (and no oversampling is used), the computed solution has oscillations near the boundaries which can cause large errors. Moreover, oversampling snapshots have several additional advantages [13] as they allow faster convergence for GMsFEM discretizations.

We compare the results obtained by using randomized snapshots to those obtained when all snapshot vectors are used. In the latter, we employ all possible boundary conditions on the oversampled region to construct the snapshot vectors. The local spectral decomposition is based on local eigenvalue problems, following previous studies [13]. Our numerical results show similar accuracy when using fewer random snapshots instead of using all possible snapshot vectors. Furthermore, we discuss approaches that can improve the results obtained by using randomized snapshots, although at a larger computational cost.

We analyze the proposed method using [27, Lemma 18] and the convergence of oversampling GMsFEM [13]. In a first step, we estimate the approximation error between the full snapshots and randomized snapshots in each coarse neighborhood in a certain norm. This approximation error is used within GMsFEM analysis to show the convergence of the solution solved in the randomized snapshot space. We also discuss adaptive strategies for randomized snapshots. In adaptive methods, additional multiscale basis functions are added based on error estimators. These estimators are proposed and investigated in [8]. Later in the paper, we discuss how additional multiscale basis functions can be computed by considering only a few extra random snapshots. In particular, in simulations we only compute four additional snapshot vectors in order to compute each additional multiscale basis function to be added as a refinement in the coarse domains that contain most error. The main objective of this paper is to show that the local snapshot spaces can be constructed inexpensively with an accuracy comparable to state-of-the-art alternatives.

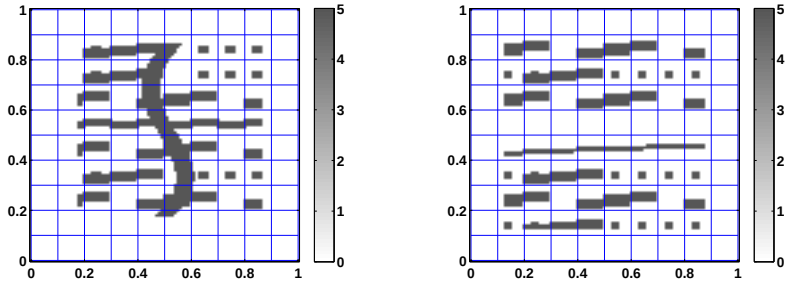
The paper is organized as follows. In section 2, we give an introductory description of GMsFEM. In section 3, we present the randomized snapshot algorithm. Section 4 is devoted to numerical results. In this section, we also discuss the use of adaptive strategies and how to compute additional multiscale basis functions. In section 5, we present the mathematical analysis of the method, and in section 6 we draw conclusions.

2. Preliminaries. We consider linear elliptic equations of the form

$$-\operatorname{div}\big(\kappa(x)\,\nabla u\big) = f \text{ in } D, \tag{1}$$

where u is prescribed on ∂D. We assume that the coefficient $\kappa(x)$ has multiple scales and high variations (see, e.g., Figure 1). Herein, we focus on two-dimensional cases but our methodology can be easily extended to problems in three dimensions, where the implied savings could be larger. This is due to the fact that in three-dimensional problems, the number of snapshots representing all Dirichlet boundary conditions is very large.

2.1. Fine and coarse grids. Let $\mathcal{T}^H$ be a conforming partition of the computational domain D into finite elements denoted by $\{K_j\}$ (triangles, quadrilaterals,



(a) $\kappa_1(x)$ (b) $\kappa_2(x)$

FIG. 1. *Permeability fields in* $\log_{10}$*-scale.*

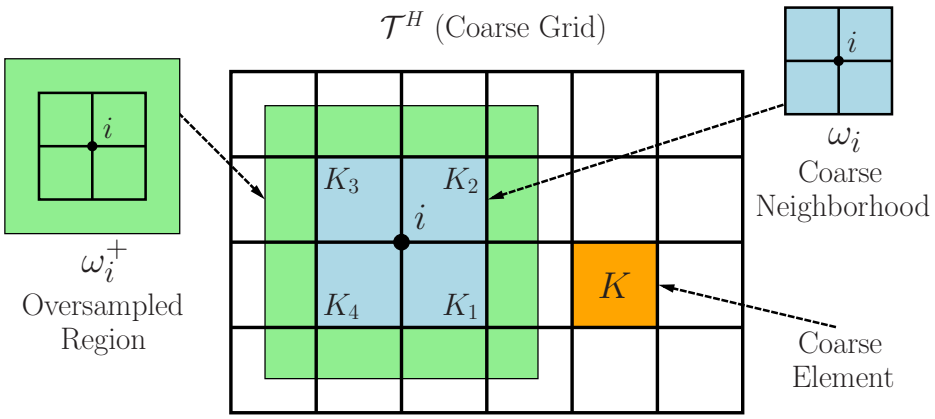


FIG. 2. *Illustration of a coarse neighborhood and an oversampled domain. Here,* K *is a coarse-grid block,* ω_i *is a coarse neighborhood of* x_i*, and* ω_i^+ *is an oversampled domain.*

tetrahedrals, etc.), called coarse grid. Assume that each coarse subregion is partitioned into a connected union of fine-grid blocks. Assume the fine grids match across coarse elements boundaries and denote by $\mathcal{T}^h$ the obtained (fine-grid) triangulation of D. We use $\{x_i\}_{i=1}^{N_c}$ (where N_c is the number of coarse nodes) to denote the vertices of the coarse mesh $\mathcal{T}^H$, and define the neighborhood of the node x_i by

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H; \quad x_i \in \overline{K}_j\}. \tag{2}$$

See Figure 2 for an illustration of neighborhoods and elements subordinated to the coarse discretization. We introduce notation for oversampled regions. We denote by ω_i^+ the oversampled region of $\omega_i \subset \omega_i^+$, defined by adding several fine- or coarse-grid layers around ω_i. Important features of the methodology are that the coarse grid is too coarse to effectively resolve all heterogeneities and scales presented in the coefficient κ, while the fine grid resolves all variations of κ but it leads to a huge linear system that is not practical to solve.

2.2. GMsFEM. Throughout this paper, we use the continuous Galerkin formulation and use ω_i as the support of the coarse basis functions. The regions ω_i^+ are used to construct the multiscale basis functions. For the purpose of this description, we formally denote the basis functions of the offline space V_{off} by $\phi_k^{\omega_i}$ where k denotes the basis function index in the domain ω_i. The solution is sought as $u_H(x) = \sum_{i,k} c_k^i \phi_k^{\omega_i}(x)$. Once the basis functions are identified, we solve

$$a(u_H, v) = (f, v) \quad \text{for all } v \in V_{\text{off}} \tag{3}$$

and

$$a(u, v) = \int_D \kappa(x) \nabla u \cdot \nabla v.$$

Now, we briefly describe GMsFEM. We consider oversampling for the GMsFEM (see [13, 12]) that uses harmonic snapshots. That is, snapshot vectors are obtained as harmonic extensions of a subset of all possible boundary conditions on the oversampled domain. We construct a snapshot space $V_{\text{snap}}^{\omega_i^+}$. The construction of the snapshot space involves solving local problems and we detail the standard process below [13, 12].

The snapshot space consists of harmonic extensions of fine-grid functions defined on the boundary of ω_i^+. More precisely, for each fine-scale function with support on the boundary of the oversampled coarse domain, $\delta_l^h(x)$, we solve a local problem. Let $\delta_l^h(x_k) = \delta_{lk}$ be one of these functions where for all $l, k \in J_h(\omega_i^+)$, where $J_h(\omega_i^+)$ are the indices of the fine-grid boundary nodes on $\partial\omega_i^+$ and δ_{lk} is Kronecker's delta with value 1 for $k = l$ and value 0 otherwise. Thus, the local problem we solve is

$$-\operatorname{div}(\kappa(x)\nabla\psi_{l,\omega_i}^{+,\text{snap}}) = 0 \ \text{ in } \omega_i^+ \tag{4}$$

subject to the boundary conditions, $\psi_{l,\omega_i}^{+,\text{snap}} = \delta_l^h(x)$ on $\partial\omega_i^+$. We form the snapshot matrices by placing the solutions of these local problems as the rows of the following matrix (throughout, for notational convenience, we do not distinguish between the fine-grid vectors and their continuous representations):

$$\Psi_{\omega_i}^{+,\text{snap}} = [\psi_{1,\omega_i}^{+,\text{snap}}; \ldots; \psi_{l,\omega_i}^{+,\text{snap}}; \ldots.].$$

We define the vectors $\psi_{l,\omega_i}^{\text{snap}}$ as the restrictions of the snapshot vectors $\psi_{1,\omega_i}^{+,\text{snap}}$ to degrees of freedom in ω_i by taking their values at the fine-grid nodes of ω_i. Considering these vectors, we form the snapshot matrix in ω_i

$$\Psi_{\omega_i}^{\text{snap}} = [\psi_{1,\omega_i}^{\text{snap}}; \ldots; \psi_{l,\omega_i}^{\text{snap}}; \ldots.]. \tag{5}$$

Next, we discuss the construction of a smaller offline space using an eigenvalue problem [12]. In order to construct an offline space V_{off}, we reduce the dimension of the snapshot space using an auxiliary spectral decomposition. We seek a subspace of the snapshot space where we approximate any element of the snapshot space in the appropriate norm defined via the following auxiliary bilinear forms. For each ω_i, we define

$$A^{\text{off}}\Theta_k^{\text{off}} = \lambda_k^{\text{off}} S^{\text{off}} \Theta_k^{\text{off}}, \tag{6}$$

where

$$A^{\text{off}} = [a_{mn}^{\text{off}}] = \int_{\omega_i} \kappa(x) \nabla\psi_{m,\omega_i}^{+,\text{snap}} \cdot \nabla\psi_{n,\omega_i}^{+,\text{snap}} = \Psi_{\omega_i}^{\text{snap}} A (\Psi_{\omega_i}^{\text{snap}})^T$$

and

$$S^{\text{off}} = [s^{\text{off}}_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x)\psi^{+,\text{snap}}_{m,\omega_i}\psi^{+,\text{snap}}_{n,\omega_i} = \Psi^{\text{snap}}_{\omega_i} S (\Psi^{\text{snap}}_{\omega_i})^T.$$

The coefficient $\widetilde{\kappa}(x)$ uses multiscale partition of unity functions (cf. [12]), which is described in (9). Here, A and S are fine-grid stiffness and mass matrices in the coarse region. To generate the offline space, we then choose the smallest M_{off} eigenvalues of (6) for each ω_i^+ and form the corresponding eigenvectors in the respective space of snapshots by setting $\psi^{+,\text{off}}_{k,\omega_i} = \sum_j \Theta^{\text{off}}_{kj}\psi^{+,\text{snap}}_{j,\omega_i}$ (for $k = 1, \ldots, M_{\text{off}}$), where Θ^{off}_{kj} are the components of the vector Θ^{off}_k. We then create the offline matrices

$$\Psi^{+,\text{off}}_{\omega_i} = \left[\psi^{+,\text{off}}_{1,\omega_i}, \ldots, \psi^{+,\text{off}}_{M_{\text{off}},w_i}\right] \quad \text{and} \quad \Psi^{\text{off}}_{\omega_i} = \left[\psi^{\text{off}}_{1,\omega_i}, \ldots, \psi^{\text{off}}_{M_{\text{off}},\omega_i}\right],$$

where $\psi^{\text{off}}_{k,\omega_i}$ is the restriction of $\psi^{+,\text{off}}_{k,\omega_i}$ to ω_i. To construct multiscale basis functions, we multiply the dominant eigenvectors by a partition of unity functions χ_i that are supported in ω_i, such that $\sum_i \chi_i = 1$. More precisely, the offline space is composed of the following basis functions:

$$\phi^{\omega_i}_k = \chi_i \psi^{\omega_i}_k. \tag{7}$$

Alternatively, we can choose the partition of unity functions to be multiscale finite element basis functions; see [16]. Let χ^0_i be the nodal bases of the standard finite element space W_H. For example, W_H consists of piecewise linear functions if $\mathcal{T}_H$ is a triangular partition or W_H consists of piecewise bi-linear functions if $\mathcal{T}_H$ is a rectangular partition. "Standard" multiscale finite element basis functions coincide with χ^0_i on the boundaries of the coarse partition and satisfy

$$\text{div}(\kappa\nabla\chi^{ms}_i) = 0 \ \text{ in } K \in \omega_i, \quad \chi^{ms}_i = \chi^0_i \ \text{ in } \partial K \quad \text{for all } \ K \in \omega_i, \tag{8}$$

where K is a coarse grid block within ω_i. In our numerical implementation, we take $\widetilde{\kappa} = \kappa$ for the computation of mass matrix. However, one can take a weighted permeability field (see detailed discussion in [12]) such as

$$\widetilde{\kappa} = \sum_i \kappa |\nabla \chi^{\text{ms}}_i|^2. \tag{9}$$

3. Randomized oversampling. As described above, a usual choice for the snapshot space consists of the harmonic extension of fine-grid functions defined on the boundary of ω_i^+. This type of snapshot is complete in the sense that it captures all the boundary information of the solution. However, the computational cost is expensive since $O(n^{\omega_i^+})$ number of local problems are required to solve in each local coarse neighborhood. Here, $n^{\omega_i^+}$ denotes the number of fine grids on the boundary of ω_i^+. A smaller yet accurate snapshot space is needed to build a more efficient multiscale method.

In the following, we generate inexpensive snapshots using random boundary conditions. That is, instead of solving (4) for each fine boundary node, we solve (4) with other boundary conditions and fewer combinations using random boundary conditions:

$$\psi^{+,\text{rsnap}}_{l,\omega_i} = r_l \text{ on } \partial\omega_i^+, \tag{10}$$

where r_l are independent identically distributed (i.i.d.) standard Gaussian random vectors on the fine-grid nodes of the boundary. Then, we can obtain the local random snapshot on the target domain ω_i by restricting the solution of this local problem, $\psi^{+,\text{rsnap}}_{l,\omega_i}$ to ω_i (which is denoted by $\psi^{\text{rsnap}}_{l,\omega_i}$). The space generated by $\psi^{\text{rsnap}}_{l,\omega_i}$ is a subspace of the space generated by all local snapshots $\Psi^{\text{snap}}_{\omega_i}$. Therefore, there exists a randomized matrix $\mathcal{R}$ with rows composed by the random boundary vectors r_l, such that

$$\Psi^{\text{rsnap}}_{\omega_i} = \mathcal{R}\Psi^{\text{snap}}_{\omega_i}. \tag{11}$$

Using these snapshots, we follow the procedure in the previous section to generate multiscale basis functions. Below, we summarize the algorithm. In our algorithm, we need to compute more snapshots to obtain an accurate offline space. We denote the dimension difference between those two spaces the buffer number, $p^{\omega_i}_{\text{bf}}$ for each ω_i and the dimension of local offline space by $k^{\omega_i}_{\text{nb}}$ for each ω_i. Later on, we use the same buffer number for all ω_i and simply use the notation p_{bf}.

Algorithm 1. Randomized GMsFEM algorithm

1: **Input**: Fine grid size h, coarse grid size H, oversampling size t, buffer number $p^{\omega_i}_{\text{bf}}$ for each ω_i, the number of local basis functions $k^{\omega_i}_{\text{nb}}$ for each ω_i;
2: Generate oversampling region for each coarse block: $\mathcal{T}^H$, $\mathcal{T}^h$, and ω_i^+;
3: Generate $k^{\omega_i}_{\text{nb}} + p^{\omega_i}_{\text{bf}}$ random vectors r_l and obtain randomized snapshots in ω_i^+ (equation (10));
Add a snapshot that represents the constant function on ω_i^+;
4: Obtain $k^{\omega_i}_{\text{nb}}$ offline bases by a spectral decomposition ((6) restricted to random snapshots);
5: Construct multiscale basis functions (equation (7)) and solve (equation (3)).
6: **output**: Coarse-scale solution u_H.

Remark 1. Computational savings of the randomized GMsFEM are derived from the fact that given an LU factorization of the local oversampled problem, forming the snapshot matrix in (5) corresponds to solving $n^{\omega_i^+}$ local problems, while forming the randomized snapshot space described in Algorithm 1 requires $k^{\omega_i}_{\text{nb}} + p^{\omega_i}_{\text{bf}}$ solutions of the same system. For all practical purposes $k^{\omega_i}_{\text{nb}} + p^{\omega_i}_{\text{bf}} << n^{\omega_i^+}$. Similarly, the eigenvalue system solved in (6) is significantly smaller and thus cheaper. These savings combined render the cost of the randomized system insignificant when compared with the full snapshot system, while the accuracy is comparable, as the numerical examples described in section 4 demonstrate.

Remark 2. The proposed algorithms that use randomized snapshots are limited to linear problems. In nonlinear problems, these techniques can be used if the problem is linearized. We expect a larger computational gain in some nonlinear problems (e.g., those studied in [14, 3]), where the snapshot spaces are constructed in each linearization step. In this case, we can avoid the costly computations and use fewer snapshot vectors.

4. Numerical results. In this section, we present representative numerical experiments that demonstrate the good performance of the randomized snapshots algorithm. We take the domain D as a square, set the forcing term $f = 0$, and use a linear boundary condition for the problem (1), that is, $u = x_1 + x_2$ on ∂D, where

x_i are the Cartesian components of each point. In our numerical simulations, we use a coarse-grid of 10×10 blocks, and each coarse-grid block is divided into 10×10 fine-grid blocks. Thus, the whole computational domain is partitioned by a 100×100 fine grid. We use a few multiscale basis functions per coarse block. This coarse basis set defines the problem size. We assume that the fine-scale solution is obtained by discretizing problem (1) by the classical conforming piecewise bilinear elements on the fine grid. To test the performance of our algorithm, we consider two permeability fields κ as depicted in Figure 1. The first permeability field (left figure) has more connected regions and they are more irregular compared to the second permeability field (right figure). We observe similar behavior for these two cases, and therefore we focus on the numerical results for the first permeability field (Figure 1(a)). In the following results, we are calculating the errors under the weighted L^2 norm and energy norm, which are defined as

$$\|u\|_{L^2_\kappa} = \left(\int_D \kappa u^2\right)^{\frac{1}{2}} \quad \text{and} \quad \|u\|_{H^1_\kappa} = \left(\int_D \kappa |\nabla u|^2\right)^{\frac{1}{2}}.$$

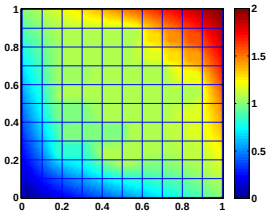
The proposed methods do not conserve mass locally, but one can use these ideas within mixed multiscale finite element methods to achieve local mass conservation.

In Table 1, a comparison between using all snapshots and using the randomized snapshots is shown. The first column shows the dimension of the offline space for each test. We choose 5, 10, 15, 20, and 25 basis functions per interior node (in addition to the constant eigenvectors) and use an oversampling layer that consists of three fine-grid blocks ($t = 3$). The offline space V_{off} is defined via a local spectral decomposition as specified in section 3. The snapshot ratio is calculated as the number of randomized snapshots divided by the number of the full snapshots. This ratio is displayed in the second column. Here, the total number of snapshots refers to the number of boundary nodes of the oversampled region. In our numerical results, an oversampled region has 26×26 fine-grid dimension and there are a total 104 snapshots if all boundary nodes are used. For example, when the dimension of the offline space is 931, we compute 14 snapshots instead of 104. This ratio gives the information on the computational savings of our algorithm compared to using all snapshots. The next two columns show the relative weighted L^2 error and relative energy error using the full snapshots. Further, the relative weighted L^2 error and relative energy error using the randomized snapshots are shown in the last two columns. From this table, we observe that the randomized algorithm converges in the sense that the relative error decreases as we increase the dimension of the coarse space. Comparing the fourth column with the last column, we conclude that the accuracy when using the randomized snapshots is similar to using all snapshot vectors. The latter has a larger dimension as the percentage of the snapshots computed shows. As a consequence, the proposed method is faster while having comparable accuracy. For example, when the dimension of the offline space is 931, the accuracy of the methods is comparable, while the randomized snapshot approach uses only 38.46% of the snapshots. Similar results are obtained when the fine mesh is refined to 200×200. In particular, with the offline space with dimension 931 and the snapshot ratio of 10%, we obtain similar $L^2_\kappa(D)$ and $H^1_\kappa(D)$ errors, which are 1.28% and 24.02%. We observe that for lower-dimensional offline spaces, we need a larger number p_{bf}. The behavior is similar when we use the permeability field in Figure 1(b). The results are displayed in Table 2. Here, p_{bf} refers to the buffer that is used to compute the eigenvectors. For example, $p_{\text{bf}} = 4$ means that we use $n + 4$ snapshots to compute n basis functions for each coarse block.

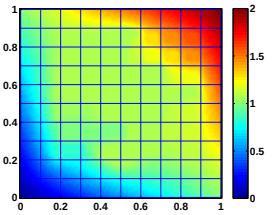
TABLE 1

Numerical results comparing the results between using all harmonic snapshots and the snapshots generated by random boundary conditions with $p_{bf} = 4$, κ as shown in Figure 1(a). *In parentheses, we show a higher value of the snapshot ratio.*

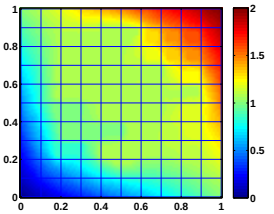
dim(V_{off})	Snapshot ratio (%)	All snapshots (%)		Few randomized snapshots (%)	
		$L^2_\kappa(D)$	$H^1_\kappa(D)$	$L^2_\kappa(D)$	$H^1_\kappa(D)$
526	8.65(15.38)	0.87	18.15	2.81(1.38)	44.95(26.04)
931	13.46(38.46)	0.64	14.85	1.04(0.72)	23.61(16.88)
1336	18.27	0.55	13.59	0.70	18.08
1741	23.08	0.50	12.69	0.64	15.91
2146	27.88	0.47	12.17	0.54	14.16



(a) Fine-scale solution



(b) Coarse-scale solution using the full snapshots



(c) Coarse-scale solution using the randomized snapshots

FIG. 3. *The fine-scale solution and coarse-scale solutions correspond to Figure* 1(a).

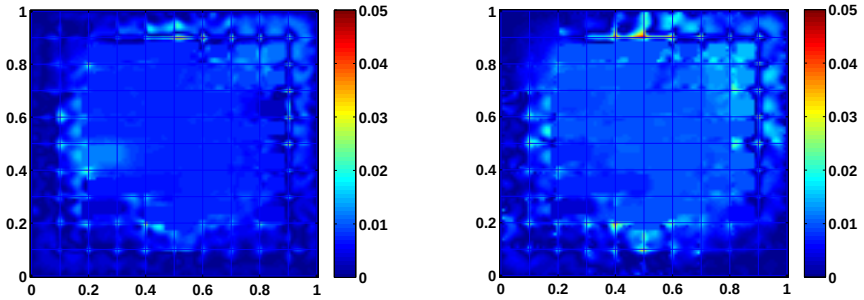
In Figure 3, the fine-scale solution, coarse-scale solution using all snapshots, and coarse-scale solution using randomized snapshots are shown. These solutions are obtained using the second test (when the dimension of the offline space is 931) in Table 1. These two coarse-scale solutions are a good approximation of the fine-scale solution. This is corroborated in Figure 4, where we plot the absolute error of the two solutions.

Next, we investigate the effect of the buffer number p_{bf} on the accuracy of the coarse solution. We test a series of simulations with different p_{bf} while keeping the coefficients and meshes fixed. The results are presented in Table 3, which shows that a larger buffer coefficient decreases the relative energy error. However, there is no need for very large buffer values. If we take $p_{\text{bf}} = 4$, we can get a coarse solution with error of 15.51% while obtaining a 14.49% error if using $p_{\text{bf}} = 20$ at the cost of solving 16 extra local problems for each inner coarse node. We also list the results using $p_{\text{bf}} = 2$ in Table 4. Comparing it with the results shown in Table 1 with $p_{\text{bf}} = 4$, one can observe that a larger p_{bf} can provide a more reliable solution.

TABLE 2

Numerical results comparing the results between using all harmonic snapshots and the snapshots generated by random boundary conditions with $p_{bf} = 4$, κ as shown in Figure 1(b).

$\dim(V_{\text{off}})$	Snapshot ratio (%)	All snapshots (%)		Using the randomized snapshots (%)	
		$L^2_\kappa(D)$	$H^1_\kappa(D)$	$L^2_\kappa(D)$	$H^1_\kappa(D)$
526	8.65(15.38)	0.71	20.98	1.33(0.80)	33.76(24.14)
931	13.46	0.51	17.33	0.66	21.67
1336	18.27	0.45	15.83	0.53	18.26
1741	23.08	0.40	14.66	0.48	17.13
2146	23.88	0.36	13.65	0.43	15.39



(a) Absolute error using the full snapshots (b) Absolute error using randomized snapshots

FIG. 4. *The absolute errors correspond to Figure* 1(a) *using full snapshots and random snapshots.*

TABLE 3

Numerical results for different p_{bf} and using 20 *local bases in each coarse neighborhood, κ as shown in Figure* 1(a).

p_{bf}	$\|u - u^{\text{off}}\|$ (%)	
	$L^2_\kappa(D)$	$H^1_\kappa(D)$
4	0.62	15.51
10	0.62	15.08
15	0.57	14.70
20	0.57	14.49

TABLE 4

Numerical results comparing the results between using all harmonic snapshots and the snapshots generated by random boundary conditions with $p_{bf} = 2$, κ as shown in Figure 1(a).

$\dim(V_{\text{off}})$	Snapshot ratio (%)	All snapshots (%)		Few randomized snapshots (%)	
		$L^2_\kappa(D)$	$H^1_\kappa(D)$	$L^2_\kappa(D)$	$H^1_\kappa(D)$
526	8.65	0.87	18.15	4.07	56.53
931	13.46	0.64	14.85	1.31	25.57
1336	18.27	0.55	13.59	0.70	18.13
1741	23.08	0.50	12.69	0.64	16.35
2146	27.88	0.47	12.17	0.54	14.16

Last, numerical tests are conducted to study the influence of oversampling effects on the accuracy of the randomized snapshots. The simulation results are shown in Table 5. From this table, we observe that the oversampling technique is needed to obtain an accurate solution. However, a larger oversampling domain is not necessary since

TABLE 5

Numerical results for different oversampling domain $\omega_i^+ = \omega_i + t$ and using 20 local bases in each coarse neighborhood, $p_{bf} = 4$, κ as shown in Figure 1(a).

t	$\|u - u^{\text{off}}\|$ (%)	
	$L^2_\kappa(D)$	$H^1_\kappa(D)$
0	1.52	23.26
2	0.61	15.63
4	0.62	15.56
7	0.59	15.24

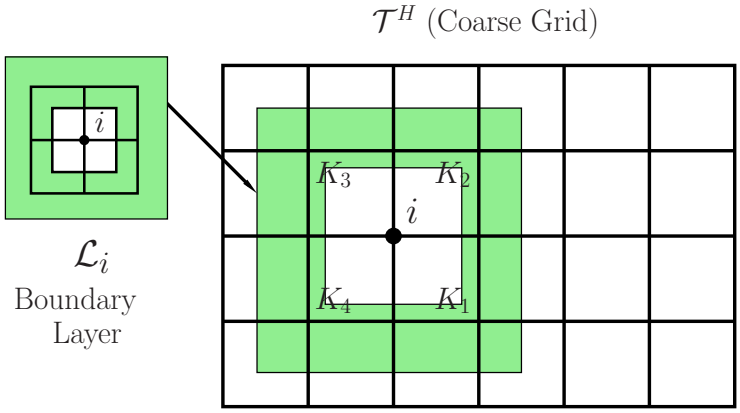


FIG. 5. *Illustration of a skin layer $\mathcal{L}_i$ that is used for computing boundary conditions for the snapshots in ω.*

it increases the computational cost of the solution, while no significant improvement in the solution accuracy is observed.

4.1. Comparison of results of different spectral problems. As we mentioned in the introduction, one can use solution-based boundary conditions to achieve higher accuracy compared to the random boundary conditions. In this section, we demonstrate this. The main idea behind this algorithm is to select boundary modes using a small spectral decomposition over the boundary layer $\mathcal{L}_i$ instead of the oversampling region ω_i^+ that surrounds the boundary in the spectral problem (6). More precisely, we consider a local spectral problem in the layer of a few fine-grid blocks in the region that contains the boundary of ω_i (see Figure 5). We choose a layer that has a thickness of five fine-grid elements (two interior to ω_i and three on the immediate neighborhood of ω_i). Furthermore, we select dominant eigenvectors (corresponding to smallest eigenvalues) by solving the local eigenvalue problem in the strip. The local eigenvalue problem uses local stiffness and mass matrices (as in [19, 15]). This approach provides correct fine-scale features and we expect higher accuracy compared to the randomized snapshots. The numerical results are shown in Table 6. Comparing the fourth column with the last column of Table 6, we observe that this new algorithm is more accurate compared to the previous one. Taking the fifth row as an example, for the same dimension of the offline space, the new algorithm gives 14.97% error, while the previous algorithm ends with 17.13%. In general, one can apply random-

TABLE 6

Numerical results comparing the results between the snapshots obtained from skin layer spectral problems and the snapshots generated by random boundaries with $p_{bf} = 4$, κ as shown in Figure 1(b).

$\dim(V_{\text{off}})$	Snapshot ratio (%)	Snapshots from skin layer (%)		Randomized snapshots (%)	
		$L^2_\kappa(D)$	$H^1_\kappa(D)$	$L^2_\kappa(D)$	$H^1_\kappa(D)$
526	8.65	1.03	26.51	1.33	33.76
931	13.46	0.63	18.64	0.66	21.67
1336	18.27	0.48	16.29	0.53	18.26
1741	23.08	0.42	14.97	0.48	17.13
2146	27.88	0.39	14.40	0.43	15.39

ized snapshot algorithms to reduce the computational cost associated with our new algorithm. That is, one can use randomized snapshots for the strip $\mathcal{L}_i$ to reduce the computational cost further.

4.2. A randomized multiscale adaptive algorithm. In this section, we discuss how to efficiently use randomized snapshots within adaptive algorithms. We use the error indicators developed in [8]. First, we briefly recall these error estimators. Letting $V_i = H^1_0(\omega_i)$, define a linear functional $R_i(v)$ on V_i by

$$R_i(v) = \int_{\omega_i} fv - \int_{\omega_i} \kappa \nabla u_{\text{ms}} \cdot \nabla v, \tag{12}$$

where the norm of R_i is defined as

$$\|R_i\|_{V_i^*} = \sup_{v \in V_i} \frac{|R_i(v)|}{\|v\|_{V_i}}. \tag{13}$$

Here $\|v\|_{V_i} = (\int_{\omega_i} \kappa(x)|\nabla v|^2\, dx)^{\frac{1}{2}}$. In [8] it is shown that

$$\|u - u_H\|^2_V \leq C_{\text{err}} \sum_{i=1}^{N} \|R_i\|^2_{V_i^*} (\lambda^{\omega_i}_{l_i+1})^{-1}, \tag{14}$$

where C_{err} is a uniform constant and $\lambda^{\omega_i}_{l_i+1}$ denotes the (l_i+1)th eigenvalue over coarse neighborhood ω_i that corresponds to the first eigenvector excluded from the construction of V_{off}. We define the error indicator in each coarse neighborhood as follows:

$$\eta_i^2 = \|R_i\|^2_{V_i^*} (\lambda^{\omega_i}_{l_i^m+1})^{-1} \qquad \text{for } H^{-1}\text{-based residual.}$$

The pivotal issue to solve is to generate additional linearly independent bases for a selected coarse neighborhood ω_i for the current iteration. Specifically, those extra bases are required to be linearly independent from the bases in the previous iteration. In what follows, we describe a possible solution to this issue using the residue of a series of random bases and their projection onto the offline space of the previous iteration.

Remark 3. Step 3 in Algorithm 2 guarantees that the added local bases are independent from the previous local bases in the M-norm as defined in the next section. In the randomized snapshots, we have added the constant local bases manually to guarantee that the multiscale bases are included. However, this constant bases should be excluded in step 3 since the constant is not in the spectral vectors and if it is added we can get linear dependency.

Algorithm 2. Local basis enrichment algorithm

1: **Input**: An index of the coarse nodes I selected by the error indicator for enrichment, the local offline space $\Psi_{\omega_i}^{\text{rsnap}}$, buffer number $c_{\text{bf}}^{\omega_i}$, an additional local basis number $c_{\text{nb}}^{\omega_i}$ for each $i \in I$.

2: Generate $c_{\text{nb}}^{\omega_i} + c_{\text{bf}}^{\omega_i}$ random vectors r_l and obtain randomized snapshots in ω_i^+ (equation (10)). Denote as $\phi_1, \ldots, \phi_{c_{\text{nb}}^{\omega_i}+c_{\text{bf}}^{\omega_i}}$;

3: A modification of the random bases obtained from step 2:

$$\tilde{\phi}_i = \phi_i - \sum_{j=1}^{N} \frac{\langle \phi_i, \psi_j \rangle_M}{\langle \psi_j, \psi_j \rangle_M} \phi_i,$$

where $\psi_1, \ldots, \psi_N$ denotes a series of bases of $\Psi_{\omega_i}^{\text{rsnap}}$ excluding the constant one;

4: Obtain $c_{\text{nb}}^{\omega_i}$ offline bases by a spectral decomposition (equation (6)), next, add in a snapshot that represent the constant function on ω_i^+, and denote the resulting vectors as $\Psi_{\omega_i}^{\text{enrich}}$;

5: $\Psi_{\omega_i}^{\text{rsnap}} \equiv \Psi_{\omega_i}^{\text{rsnap}} \cup \Psi_{\omega_i}^{\text{enrich}}$.

6: **output**: an enriched local offline space $\Psi_{\omega_i}^{\text{rsnap}}$ corresponds to each nodes in I.

The numerical results are displayed in Table 7. First, we take five bases per coarse block. Then, we apply the multiscale adaptive algorithm proposed in [8] and identify the coarse nodes index I requiring more bases. Set $c_{nb}^{\omega_i} = 2$ and $c_{bf}^{\omega_i} = 1$ and follow Algorithm 2; next, we generate $c_{nb}^{\omega_i} + c_{bf}^{\omega_i} = 3$ local random bases for those nodes and use Step 3 to get three new linearly independent bases. Afterwards, a local spectral decomposition is performed to select two important bases from those three bases. In the end, the corresponding multiscale basis functions are constructed and added to the coarse space.

Comparing Tables 1 and 7, we observe that the randomized adaptive algorithm is cheaper since many fewer basis functions are used to achieve comparable accuracy to that of the uniform increase of bases shown in Table 2. We use 2146 basis functions to attain an energy error of 14.16% in Table 2, while only 2061 are necessary to get a smaller error of 13.90% using the adaptive randomized algorithm. The main computational gain in the adaptivity is that we can select the number of basis functions adaptively in each region. In many applications, one may need a few multiscale basis functions in many coarse regions (e.g., coarse regions that contain isolated high-conductivity inclusions), and adaptive algorithms can detect this based on a residual information. Here, we do not discuss the computational cost of our adaptive algorithm in detail and refer to [8]. Our main emphasis is that one can use randomized snapshots within existing adaptive algorithms.

TABLE 7

Numerical results using adaptive algorithm with $p_{bf} = 4$, and 5 local bases per node at the beginning and with two more bases for selected nodes, κ as shown in Figure 1(a).

$\dim(V_{\text{off}})$	Using the usual snapshots (%)	
	$L^2_\kappa(D)$	$H^1_\kappa(D)$
526	4.11	50.23
916	0.99	21.65
1323	0.63	17.33
1717	0.53	15.10
2061	0.51	13.90

5. Analysis. In this section, we present an analysis of the method described above. Our main objective is to show that using randomized snapshots, we can obtain similar convergence results to using all snapshot vectors. In the analysis described below, we first estimate the error due to the approximation using randomized snapshots. In the first lemma, we compare an arbitrary snapshot obtained using all snapshot vectors and its approximation in the space of randomized snapshots. To avoid cumbersome notation, we denote the local snapshot matrix $\Psi_{\omega_i}^{\text{snap}}$ in (5) by Ψ and the local randomized snapshot matrix $\Psi_{\omega_i}^{\text{rsnap}}$ in (11) by Ψ^r.

The following lemma shows that the randomized snapshot Ψ^r with l random bases is a good approximation of the full snapshot Ψ composed of m bases, $m > l$. We use the notation $A \preceq B$ when $A \leq CB$ with C being independent of the size ratio between the coarse and fine meshes, and spatial scales. Throughout, $\|\cdot\|$ denotes the l^2 norm for vectors and the l^2-based spectral norm for matrices, while $\|z\|_A^2 = z^T A z$. We recall that throughout, for notational convenience, we do not distinguish between the fine-grid vectors and their continuous representations.

LEMMA 4. *Suppose $\Psi \in \mathbf{R}^{m\times n}$ of rank m and $\mathcal{R} \in \mathbf{R}^{l\times m}$ whose entries are i.i.d. Gaussian random variables. Define $\Psi^r = \mathcal{R}\Psi$; then, for any $\xi \in \mathbf{R}^m$, there exists $\xi^r \in \mathbf{R}^l$ such that*

(15)
$$\left\|\xi^T\Psi - (\xi^r)^T\Psi^r\right\|_{\widetilde{M}(\omega_i)}^2 = \int_{\omega_i} \widetilde{\kappa}|\xi^T\Psi - (\xi^r)^T\Psi^r|^2 \preceq \left(\frac{\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\| + 1}{\lambda_{k+1}}\right)^2 \left\|\xi^T\Psi\right\|_{A(\omega_i)}^2,$$

where $k < l < m < n$, and $\mathcal{S}$ and $\mathcal{H}$ are defined in (17) *and* (19).

Here, λ_{k+1} is the $(k+1)$th smallest diagonal value of Λ defined in (16) *and $A(\omega_i) = (\int_{\omega_i} (\kappa\nabla\phi_j)^T\nabla\phi_k)_{n\times n}$ with ϕ_j as the jth local fine-scale basis in the ω_i. Besides, $\left\|\xi^T\Psi\right\|_{\widetilde{M}(\omega_i)} = (\int_{\omega_i}(\widetilde{\kappa}\xi^T\Psi)^T\xi^T\Psi)^{\frac{1}{2}}$, $\left\|\xi^T\Psi\right\|_{A(\omega_i)} = (\int_{\omega_i}(\kappa\nabla(\xi^T\Psi))^T\nabla(\xi^T\Psi))^{\frac{1}{2}}$.*

Proof. Denote $\widetilde{M}(\omega_i) = (\int_{\omega_i} \widetilde{\kappa}\phi_j\phi_k)_{n\times n}$ with ϕ_j as the jth local fine-scale bases in ω_i. The matrix $\widetilde{M}(\omega_i)$ is symmetric positive definite. Besides, $A(\omega_i)$ is symmetric semipositive definite. Thus, there exists an $m \times m$ matrix U such that

$$U^T\Psi\widetilde{M}(\omega_i)\Psi^T U = \Lambda \text{ and } U^T\Psi A(\omega_i)\Psi^T U = I, \tag{16}$$

where I is an identity matrix and Λ denotes a diagonal matrix with nondecreasing diagonal values,

$$\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_m}.$$

Define $X = U^{-T}\Lambda^{\frac{1}{2}}$; then we obtain $XX^T = \Psi\widetilde{M}(\omega_i)\Psi^T$.

Suppose F is a matrix of dimension $m \times l$; take $\xi^r = F^T\xi$. Then

$$\begin{aligned}\int_{\omega_i} \widetilde{\kappa}|\xi^T\Psi - \xi^{r^T}\Psi^r|^2 &= (\xi^T\Psi - (\xi^r)^T\mathcal{R}\Psi)\widetilde{M}(\omega_i)(\xi^T\Psi - (\xi^r)^T\mathcal{R}\Psi)^T \\ &= \xi^T(I - F\mathcal{R})\Psi\widetilde{M}(\omega_i)\Psi^T(I - F\mathcal{R})^T\xi \\ &= \xi^T(X - F\mathcal{R}X)(X - F\mathcal{R}X)^T\xi = \left\|\xi^T(X - F\mathcal{R}X)\right\|^2.\end{aligned}$$

In the following, we construct a matrix F that minimizes $\|\xi^T(X - F\mathcal{R}X)\|$. Following [27, Lemma 18], we define

$$F = U^{-T}\begin{pmatrix}\mathcal{H}^{(-1)}\\ 0\end{pmatrix},$$

where $\mathcal{H}$ and $\mathcal{S}$ are matrices of dimension $l \times k$ and $l \times (m-k)$ defined as

$$\mathcal{R}U^{-T} = \begin{pmatrix}\mathcal{H} & \mathcal{S}\end{pmatrix}, \tag{17}$$

$$\mathcal{H}^{(-1)} = (\mathcal{H}^T\mathcal{H})^{-1}\mathcal{H}^T. \tag{18}$$

That is, $\mathcal{H}$ is of rank k and contains the first k columns of $\mathcal{R}U^{-T}$ and $\mathcal{H}^{(-1)}$ is the pseudoinverse of $\mathcal{H}$.

We obtain

$$\xi^T(X - F\mathcal{R}X) = -\xi^T U^{-T}\left(\begin{pmatrix}\mathcal{H}^{(-1)}\\ 0\end{pmatrix}\begin{pmatrix}\mathcal{H} & \mathcal{S}\end{pmatrix} - I\right)\Lambda^{\frac{1}{2}}.$$

Furthermore,

$$\|\xi^T(X - F\mathcal{R}X)\| \le \|\xi^T U^{-T}\|(\|\mathcal{H}^{(-1)}\mathcal{S}T\| + \|T\|),$$

where T is defined as

$$\Lambda^{\frac{1}{2}} = \begin{pmatrix}S & 0\\ 0 & T\end{pmatrix}. \tag{19}$$

Thus, the spectral norm of T is bounded, that is, $\|T\| \le \frac{1}{\lambda_{k+1}}$. Then, using standard properties of subordinated norms we have

$$\|\xi^T(X - F\mathcal{R}X)\| \le \|\xi^T U^{-T}\|(\|\mathcal{H}^{(-1)}\mathcal{S}T\| + \|T\|) \tag{20}$$

$$\le \|\xi^T U^{-T}\|(\|\mathcal{H}^{(-1)}\mathcal{S}\| + 1)\|T\| \tag{21}$$

$$\le \frac{\|\mathcal{H}^{(-1)}\mathcal{S}\| + 1}{\lambda_{k+1}}\|\xi^T\Psi\|_{A(\omega_i)}. \tag{22}$$

Here, to obtain the last step we have used the relation (16), which implies

$$\|\xi^T U^{-T}\| = (\xi^T U^{-T}\cdot(\xi^T U^{-T})^T)^{\frac{1}{2}} = (\xi^T U^{-T}U^{-1}\xi)^{\frac{1}{2}} \tag{23}$$

$$= (\xi^T\Psi A(\omega_i)\Psi^T\xi)^{\frac{1}{2}} = \|\xi^T\Psi\|_{A(\omega_i)}. \tag{24}$$

Hence,

$$\int_{\omega_i}\kappa|\nabla\chi|^2|\xi^T\Psi - \xi^{r^T}\Psi^r|^2 \le \left(\frac{\|\mathcal{H}^{(-1)}\mathcal{S}\| + 1}{\lambda_{k+1}}\right)^2\|\xi^T\Psi\|^2_{A(\omega_i)}.$$

The proof is complete. ∎

Remark 5 (estimate for $\|\mathcal{H}^{(-1)}\mathcal{S}\|$). U in Lemma 4 is orthonormal with respect to the $A(\omega_i)$-inner product. If U is an orthonormal matrix itself, then by [27, Lemma 18], $\|\mathcal{H}^{(-1)}\mathcal{S}\| \le \sqrt{l}\beta\frac{1}{\lambda^2_{k+1}}$ for some positive number β and given k. If U is not orthonormal, then by applying the Gram–Schmidt process to the first k columns of

U^{-T} (denoted as V_1) as well as the rest of the columns of it (denoted as V_2), we can obtain nonsingular triangular matrices D_1 and D_2, and S_1 and S_2 with $S_1^T S_1 = I$ and $S_2^T S_2 = I$, such that

$$U^{-T} = \begin{pmatrix} V_1 & V_2, \end{pmatrix},$$

$$V_1 = S_1 D_1, \text{ and } V_2 = S_2 D_2.$$

Then

$$\mathcal{H}^{(-1)}\mathcal{S} = (\mathcal{H}^T\mathcal{H})^{-1}\mathcal{H}^T\mathcal{S} = (V_1^T\mathcal{R}^T\mathcal{R}V_1)^{-1}V_1^T\mathcal{R}^T\mathcal{R}V_2,$$

and using the expressions for V_1 and V_2, we obtain

$$\mathcal{H}^{(-1)}\mathcal{S} = D_1^{-1}(S_1^T\mathcal{R}^T\mathcal{R}S_1)^{-1}(D_1^{-T}D_1^T)(S_1^T\mathcal{R}^T\mathcal{R}S_2)D_2.$$

Therefore, we have

$$\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\| \leq \left\|D_1^{-1}\right\|\left\|(S_1^T\mathcal{R}^T\mathcal{R}S_1)^{-1}S_1^T R^T\right\|\left\|\mathcal{R}S_2\right\|\left\|D_2\right\|.$$

Since the entries of RS_1 and RS_2 are i.i.d. Gaussian random variables of zero mean and unit variance, using [27, Lemma 14], we get the estimate

$$\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\| \leq \sqrt{2lm\beta^2\gamma^2+1}\left\|D_1^{-1}\right\|\left\|D_2\right\|$$

with probability not less than

$$1 - \frac{1}{\sqrt{2\pi(l-k+1)}}\left(\frac{e}{(l-k+1)\beta}\right)^{l-k+1} - \frac{1}{4(\gamma^2-1)\sqrt{\pi m\gamma^2}}\left(\frac{2\gamma^2}{e^{\gamma^2-1}}\right)^m,$$

where β and γ are positive real numbers, $\gamma > 1$.

Next, we note that the ith diagonal elements of D_1 and D_2 are the norms of ith columns of V_1 and V_2. Moreover, $\left\|D_1^{-1}\right\|\left\|D_2\right\|$ is the ratio of the largest diagonal element of D_2 and the smallest diagonal element of D_1. Since $U^{-T}U^{-1} = \Psi A\Psi^T$ and $U^{-T}\Lambda U^{-1} = \Psi\widetilde{M}\Psi^T$, the estimate of $\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\|$ depends on the norms of the columns of U^{-T} and, thus, depends on the contrast, in general. In the particular case, we assume that $\Psi A\Psi^T$ is a diagonal matrix with entries $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$. In this case, $U^{-T} = U^{-1} = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_n^{-1/2})$ and $D_1 = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_l^{-1/2})$, $D_2 = \text{diag}(\lambda_{l+1}^{-1/2}, \lambda_{l+2}^{-1/2}, \ldots, \lambda_n^{-1/2})$. Then, it is easy to verify that $\left\|D_1^{-1}\right\|\left\|D_2\right\| = \lambda_l^{1/2}/\lambda_{l+1}^{1/2}$ in this case. This estimate shows that the error can be sensitive on the choice of the eigenspace that is selected. In GMsFEM, we usually select the most important eigenvalues that are small (see [15]); thus, in general, a contrast-dependent situation can be avoided.

In Lemma 4, we have derived the approximation of the randomized snapshot space to the full snapshot space locally in each patch ω_i. Next, we present the convergence the GMsFEM using randomized snapshots. The snapshots are obtained by multiplying the local snapshots $\Psi_{\omega_i}^{\text{snap}}$ with the corresponding partition of unity function χ_i (as in (7)). To simplify notation we denote by Ψ the full global snapshots (snapshots for all ω_i's) and by Ψ^r the full randomized snapshots (snapshots for all ω_i's).

THEOREM 6. *Denote by Ψ the snapshot matrix and by Ψ^r the randomized snapshot matrix of dimension $m \times n$ and $l \times n$, respectively, and their ranks are m and l, respectively. $\mathcal{R}$ is a matrix with i.i.d. Gaussian random entries and $\Psi^r = \mathcal{R}\Psi$.*

Suppose u_H is solved using the offline space formed using the snapshot matrix Ψ^r, and u is the fine-scale solution of (1)*; then we have*

$$(25)\int_D \kappa|\nabla(u-u_H)|^2 \preceq \left(\frac{1}{\Lambda_*}+\left(\frac{1}{\Lambda_*}\right)^2(\|\mathcal{H}^{(-1)}\mathcal{S}\|+1)^2\right)\int_D \kappa|\nabla u|^2 + H^2\int_D f^2,$$

where Λ_* is defined in (36) *and $l<m<n$.*

Proof. Denote I^{ω_i} and $I_r^{\omega_i}$ as arbitrary interpolants from the fine scale to the space spanned by the rows of Ψ and Ψ^r on the coarse neighborhood ω_i, respectively. Later, we choose a proper interpolant that reduces the error. Taking into account that the GMsFEM solution, u_H, provides a minimal energy error, we have

$$\begin{aligned}(26)\quad \int_D \kappa|\nabla(u-u_H)|^2 &\preceq \int_D \kappa|\nabla\left(\sum_i \chi_i(u-I_r^{\omega_i}u)\right)|^2\\ &\preceq \sum_i\int_{\omega_i}\kappa|\nabla(\chi_i(u-I^{\omega_i}u))|^2+\int_{\omega_i}\kappa|\nabla(\chi_i(I_r^{\omega_i}u-I^{\omega_i}u))|^2.\end{aligned}$$

Next, we use the inequalities

$$(27)\qquad \int_{\omega_i}\kappa\chi_i^2|\nabla(u-I^{\omega_i}u)|^2 \preceq \int_{\omega_i}\widetilde{\kappa}|(u-I^{\omega_i}u)|^2+\left|\int_{\omega_i} f\chi_i^2(u-I^{\omega_i}u)\right|,$$

$$(28)\qquad \int_{\omega_i}\kappa\chi_i^2|\nabla(I_r^{\omega_i}u-I^{\omega_i}u)|^2 \preceq \int_{\omega_i}\widetilde{\kappa}|(I_r^{\omega_i}u-I^{\omega_i}u)|^2,$$

where $\widetilde{\kappa}$ is defined by (9). Here, we have used the inequality (29) in [15]. Using (27) and (28), and we obtain from (26)

$$\begin{aligned}\int_D \kappa|\nabla(u-u_H)|^2 \preceq \sum_i\int_{\omega_i}\widetilde{\kappa}|(u-I^{\omega_i}u)|^2+\sum_i\left|\int_{\omega_i} f\chi_i^2(u-I^{\omega_i}u)\right|\\ +\sum_i\int_{\omega_i}\widetilde{\kappa}|(I_r^{\omega_i}u-I^{\omega_i}u)|^2.\end{aligned}\qquad(29)$$

Selecting a proper interpolant I^{ω_i}, we have

$$(30)\qquad \int_{\omega_i}\widetilde{\kappa}|(u-I^{\omega_i}u)|^2 \preceq \frac{1}{\lambda_{k+1}^{\omega_i}}\int_{\omega_i}\kappa|\nabla(u-I^{\omega_i}u)|^2,$$

where $\lambda_{k+1}^{\omega_i}$ is the eigenvalue that corresponds to the first eigenvector that is not included in the coarse space. Similarly, we can show that

$$\begin{aligned}\left|\int_{\omega_i} f\chi_i^2(u-I^{\omega_i}u)\right| &\preceq \int_{\omega_i}\widetilde{\kappa}^{-1}f^2+\int_{\omega_i}\widetilde{\kappa}|(u-I^{\omega_i}u)|^2\\ &\preceq \int_{\omega_i}\widetilde{\kappa}^{-1}f^2+\frac{1}{\lambda_{k+1}^{\omega_i}}\int_{\omega_i}\kappa|\nabla(u-I^{\omega_i}u)|^2.\end{aligned}\qquad(31)$$

We note that $\int_{\omega_i}\widetilde{\kappa}^{-1}f^2 \preceq \kappa^{-1}\int_{\omega_i}f^2$ if $|\nabla\chi_i^{\mathrm{ms}}| = \mathcal{O}(H^{-1})$. Combining the above estimates, we have

$$\begin{aligned}\int_D \kappa|\nabla(u-u_H)|^2 \preceq \sum_i\frac{1}{\lambda_{k+1}^{\omega_i}}\int_{\omega_i}\kappa|\nabla(u-I^{\omega_i}u)|^2\\ +\sum_i\int_{\omega_i}\widetilde{\kappa}^{-1}f^2+\sum_i\int_{\omega_i}\widetilde{\kappa}|(I_r^{\omega_i}u-I^{\omega_i}u)|^2.\end{aligned}\qquad(32)$$

For a fixed vector $I^{\omega_i}u \in \Psi$, by Lemma 4, we can get a corresponding vector $\xi^r \in \Psi^r$, such that

$$\int_{\omega_i} \widetilde{\kappa}|\xi^r - I^{\omega_i}u|^2 \preceq \left(\frac{\left\|\mathcal{H}^{(-1)}(\omega_i)\mathcal{S}(\omega_i)\right\| + 1}{\lambda_{k+1}^{\omega_i}}\right)^2 \left\|I^{\omega_i}u\right\|^2_{A(\omega_i)}, \tag{33}$$

for some integer k. For simplicity, we assume that $\lambda_{k+1}^{\omega_i}$ is the same eigenvector as in the interpolant defined in (30) by selecting the smallest index.

We define $I_r^{\omega_i}u = \xi^r$. Thus using (32) and (33), we obtain

$$\begin{aligned}\int_D \kappa|\nabla(u-u_H)|^2 &\preceq \max_{\omega_i}\left(\frac{1}{\lambda_{k+1}^{\omega_i}}\right)\int_{\omega_i}\kappa|\nabla u|^2 + \sum_i \int_{\omega_i}\widetilde{\kappa}^{-1}f^2 \\ &\quad + \sum_i \left(\frac{\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\|+1}{\lambda_{k+1}^{\omega_i}}\right)^2 \left\|I^{\omega_i}u\right\|^2_{A(\omega_i)} \qquad (34)\\ &\preceq \left(\frac{1}{\Lambda_*} + \frac{1}{\Lambda_*^2}(\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\|+1)^2\right)\int \kappa|\nabla u|^2 + \sum_i\int_{\omega_i}\widetilde{\kappa}^{-1}f^2, \qquad (35)\end{aligned}$$

where

$$\Lambda_* = \min_{\omega_i}\lambda_{k+1}^{\omega_i}. \tag{36}$$

Here, we have used the boundedness property of the interpolant in the energy norm [19]. Assuming $|\nabla\chi_i| = \mathcal{O}(H^{-1})$, we get

$$\int_D \kappa|\nabla(u-u_H)|^2 \preceq \left(\frac{1}{\Lambda_*} + \frac{1}{\Lambda_*^2}(\left\|\mathcal{H}^{(-1)}\mathcal{S}\right\|+1)^2\right)\int \kappa|\nabla u|^2 + H^2\int_D f^2. \quad \square \tag{37}$$

Remark 7. One can improve the error due to GMsFEM discretization by changing the eigenvalue problem (see [13]) and the error will scale as $\frac{1}{\Lambda_*^q}$ for a large q that depends on the size of the oversampled region. In this case, the error due to GMsFEM discretization will scale as $(1/\Lambda_*)^n$ for some large n.

6. Conclusions. In this paper, we study the use of randomized boundary conditions to reduce the computational cost in multiscale finite element methods. Local multiscale finite element basis functions are constructed in each coarse patch by computing snapshot vectors and performing local spectral decompositions. The choice of snapshot vectors and the local spectral decomposition is important for achieving low-dimensional coarse spaces that can approximate the solution accurately on a coarse mesh. For example, the use of harmonic functions computed in oversampled regions improves the accuracy. However, the computation of harmonic functions for all possible boundary conditions in each local region is expensive. Therefore, we propose the use of randomized boundary conditions for computing the snapshot vectors. We show that with a few snapshot vectors, we can compute the basis functions that provide an accuracy that is similar to that obtained using all snapshot vectors. We analyze the method and validate our estimates with numerical evidence. Moreover, we discuss approaches that are more accurate compared to randomized snapshots; however, they are more expensive. Finally, we discuss how adaptive computations can be performed efficiently and robustly within the framework of randomized snapshots where multiscale basis functions are added locally in some regions based on an error indicator.

Acknowledgment. Yalchin Efendiev would like to thank the partial support from the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award DE-FG02-13ER26165 and the DoD Army ARO Project.

REFERENCES

[1] J. E. Aarnes and Y. Efendiev, *Mixed multiscale finite element for stochastic porous media flows*, SIAM J. Sci. Comput., 30 (2008), pp. 2319–2339.
[2] J. E. Aarnes, Y. Efendiev, and L. Jiang, *Analysis of multiscale finite element methods using global information for two-phase flow simulations*, Multiscale Model. Simul., 7 (2008), pp. 2177–2193.
[3] M. Alotaibi, V. M. Calo, Y. Efendiev, J. Galvis, and M. Ghommem, *Global–local nonlinear model reduction for flows in heterogeneous porous media*, Comput. Methods Appl. Mech. Engrg., 292 (2015), pp. 122–137.
[4] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, *A multiscale mortar mixed finite element method*, Multiscale Model. Simul., 6 (2007), pp. 319–346.
[5] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, *A multiscale mortar mixed finite element method*, Multiscale Model. Simul., 6 (2007), pp. 319–346.
[6] I. Babuška and R. Lipton, *Optimal local approximation spaces for generalized finite element methods with application to multiscale problems*, Multiscale Model. Simul., 9 (2011), pp. 373–406.
[7] I. Babuška, V. Nistor, and N. Tarfulea, *Generalized finite element method for second-order elliptic operators with Dirichlet boundary conditions*, J. Comput. Appl. Math., 218 (2008), pp. 175–183.
[8] E. T. Chung, Y. Efendiev, and G. Li, *An adaptive GMsFEM for high-contrast flow problems*, J. Comput. Phys., 273 (2014), pp. 54–76.
[9] M. Drohmann, B. Haasdonk, and M. Ohlberger, *Reduced bases approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937–A969.
[10] L. J. Durlofsky, *Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media*, Water Resour. Res., 27 (1991), pp. 699–708.
[11] Y. Efendiev, J. Galvis, and E. Gildin, *Local-global multiscale model reduction for flows in highly heterogeneous media*, J. Comput. Phys., 231 (2012), pp. 8100–8113.
[12] Y. Efendiev, J. Galvis, and T. Hou, *Generalized multiscale finite element methods*, J. Comput. Phys., 251 (2013), pp. 116–135.
[13] Y. Efendiev, J. Galvis, G. Li, and M. Presho, *Generalized multiscale finite element methods. Oversampling strategies*, Internat. J. Multiscale Comput. Engrg., 12(6), 2014.
[14] Y. Efendiev, J. Galvis, G. Li, and M. Presho, *Generalized multiscale finite element methods. Nonlinear elliptic equations*, Commun. Comput. Phys., 15 (2014), pp. 733–755.
[15] Y. Efendiev, J. Galvis, and X. H. Wu, *Multiscale finite element methods for high-contrast problems using local spectral bases functions*, J. Comput. Phys., 230 (2011), pp. 937–955.
[16] Y. Efendiev and T. Hou, *Multiscale Finite Element Methods: Theory and Applications*, Spring Science & Business Media, 2009.
[17] Y. Efendiev, T. Hou, and X. H. Wu, *Convergence of a nonconforming multiscale finite element method*, SIAM J. Numer. Anal., 37 (2000), pp. 888–910.
[18] R. Ewing, O. Iliev, R. D. Lazarov, I. Rybak, and J. Willems, *A simplified method for upscaling composite materials with high contrast of the conductivity*, SIAM J. Sci. Comput., 31 (2009), pp. 2568–2586.
[19] J. Galvis and Y. Efendiev, *Domain decomposition preconditioners for multiscale flows in high contrast media*, Multiscale Model. Simul., 8 (2010), pp. 1461–1483.
[20] M. Ghommem, M. Presho, V. M. Calo, and Y. Efendiev, *Mode decomposition methods for flows in high-contrast porous media. Global-local approach*, J. Comput. Phys., 253 (2013), pp. 226–238.
[21] H. Hajibeygi, D. Kavounis, and P. Jenny, *A hierarchical fracture model for the iterative multiscale finite volume method*, J. Comput. Phys., 230 (2011), pp. 8729–8743.
[22] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[23] T. Hou and X. H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comput. Phys., 134 (1997), pp. 169–189.

[24] O. P. Iliev, R. D. Lazarov, and J. Willems, *Fast numerical upscaling of heat equation for fibrous materials*, J. Comput. Vis. Sci., 13 (2010), pp. 275–285.
[25] P. Jenny, S. H. Lee, and H. Tchelepi, *Multi-scale finite volume method for elliptic problems in subsurface flow simulation*, J. Comput. Phys., 187 (2003), pp. 47–67.
[26] L. Jiang and I. D. Mishev, *Mixed multiscale finite volume methods for elliptic problems in two-phase flow simulations*, Commun. Comput. Phys., 11 (2012), pp. 19–47.
[27] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A Randomized Algorithm for the Approximation of Matrices*, YALEU/DCS/TR-1361, Yale University, 2006.
[28] X. H. Wu, Y. Efendiev, and T. Y. Hou, *Analysis of upscaling absolute permeability*, Discrete Contin. Dyn. Sys. Ser. B, 2 (2002), pp. 158–204.