Proceedings of the Institute of Mathematics and Mechanics,
National Academy of Sciences of Azerbaijan
Volume 42, Number 2, 2016, Pages 212–229

A MULTISCALE DISCONTINUOUS GALERKIN METHOD IN PERFORATED DOMAINS

ERIC T. CHUNG, YALCHIN EFENDIEV, MARIA VASILYEVA, AND YATING WANG

Abstract. In this paper, we develop and investigate a multiscale model reduction technique within the framework of Interior Penalty Discontinuous Galerkin methods for problems in perforated domains. Previous research for developing multiscale methods for perforated domains is limited to continuous Galerkin formulations, which have some limitations. Discontinuous Galerkin approaches provide some advantages as they avoid partition of unity functions, allow more flexibility in constructing of basis functions and can be easily parallelized. We will present numerical examples for various 2D and some 3D examples to demonstrate the efficiency and accuracy of the proposed schemes.

1. Introduction

Many problems in perforated domains exhibit multiscale nature due to variations in the perforation sizes and their geometries. These include problems in porous media [15, 3], processes in membranes, and filters [13]. Due to scale disparity, some types of model reduction techniques are needed. Typical approaches construct reduced-order models on a coarse grid, where the coarse-grid sizes are much larger than the perforation sizes. Some successful approaches include homogenization methods where the macroscopic equations are formulated on a coarse grid. In these approaches, the macroscale equations are formulated in the domain without perforations. Due to perforations, the macroscopic equations can be of the form different from a microscopic equations.

Some numerical approaches for multiscale model reduction include numerical homogenization [2], multiscale finite element methods [15, 10, 1], Localized Orthogonal Decomposition [3], and generalized multiscale finite element methods (GMsFEM) [9]. *Due to page limitation, we can not give an exhaustive list of references here.* In these approaches, the multiscale basis functions are constructed in each coarse block to represent the local heterogeneities in each coarse region. In some of the approaches, only a limited number of basis functions are constructed. The GMsFEM is a systematic approach to identify multiscale basis functions via local snapshots and local spectral problems. The local snapshots are constructed by solving local problems and contain the information about local heterogeneities. By performing local spectral decomposition, the method identifies multiscale basis functions.

Some recent approaches concerning the development of multiscale methods in perforated domains are done within continuous Galerkin framework (see [15] for CrouzeixRaviart based multiscale methods). In these approaches, either local multiscale basis functions are multiplied by partition of unity functions or multiscale basis functions are constructed via conforming boundary conditions. When multiplying by partition of unity functions, there are several difficulties. First, one needs to identify appropriate partition of unity functions that can honor local boundary conditions on the perforations. Secondly, the partition of unity functions modify the local heterogeneities. To avoid this, we propose using a discontinuous Galerkin formulation. In terms of multiscale basis construction, we mainly follow our previous developments. The main novelty of the paper is developing a discontinuous Galerkin framework for problems in perforated domains.

Developing discontinuous Galerkin methods in conjunction with the GMsFEM is not new [15, 11, 5, 8]. In a number of papers, we have considered problems without perforations. This paper is the first work where the Interior Penalty Discontinuous Galerkin (IPDG) framework for the GMsFEM in perforated domains is developed. The main advantages of the IPDG framework are: (1) it avoids partition of unity functions; (2) it uses non-overlapping partitions; (3) it can use unstructured coarse and fine meshes; and (4) it provides block diagonal mass matrices.

In this paper, we study developing the discontinuous Galerkin GMsFEM for multiscale problems in perforated domains. We consider both elliptic flow equations and elasticity equations. We present the construction of local snapshot spaces and local spectral decomposition. The multiscale basis functions are constructed independently for each coarse-grid blocks and coupled via IPDG coupling.

We present some numerical examples. Our examples are performed for both structured and unstructured coarse grids. We study both 2D and 3D examples. The numerical results show that one can achieve good accuracy with a few basis functions. We present detailed numerical studies, which demonstrate our approach can be used for solving problem in perforated domains.

The paper is organized as follows. In Section 2, we introduce the problem and the fine-scale approximation using a Discontinuous Galerkin method. In Section 3, we present the framework of discontinuous Galerkin GMsFEM (GMsDGM) and constructing coarse scale function spaces. The randomized snapshots using an oversampling technique are discussed in Section 4. Next, in Section 5, we present some numerical examples when we use various perforated domains and high contrast medium. The three dimensional realization is shown in Section 6. Finally, we conclude our work in Section 7.

2. Problem formulation and fine-scale approximation

We assume $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ to be a bounded domain covered by inactive/active cells or particles $\mathcal{B}^\epsilon$ (see Figure 1). In this work, we consider problems in perforated domains (see left of Figure 4) or high-contrast domains (see right of Figure 4). The perforated domain is denoted by $\Omega^\epsilon = \Omega \backslash \mathcal{B}^\epsilon$. For the high-contrast domain, we set $\Omega^\epsilon = \Omega$, and define different permeability coefficients in

two subdomains ($\Omega\backslash\mathcal{B}^\epsilon$ and $\mathcal{B}^\epsilon$). we do not assume periodicity or scale separation in this paper.

The following problem is considered

$$\mathcal{L}(u) = f \text{ in } \Omega^\epsilon, \quad u = g_D \text{ on } \Gamma_D, \quad \mathcal{B}(u) = g_N \text{ on } \Gamma_N, \tag{2.1}$$

where $\mathcal{L}$ denotes a linear differential operator, $\mathcal{B}$ is a normal derivative operator and $\Gamma_D \cup \Gamma_N = \partial\Omega^\epsilon$.

- For the Laplace operator, we have

$$\mathcal{L}(u) = -\nabla \cdot (\kappa \nabla u), \quad \mathcal{B}(u) = \kappa \frac{\partial u}{\partial n} \tag{2.2}$$

  where n is the outward unit normal on $\partial\Omega^\epsilon$.
- For the elasticity operator, we assume the medium is isotropic. Let $u \in (H^1(\Omega^\epsilon))^d$ be the displacement field. The strain $\varepsilon(u) \in (L^2(\Omega^\epsilon))^{d\times d}$, and stress tensors $\sigma(u) \in (L^2(\Omega^\epsilon))^{d\times d}$ are defined as

$$\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T), \quad \sigma(u) = 2\mu\varepsilon(u) + \lambda \nabla \cdot u\, \mathcal{I},$$

  where $\lambda > 0$ and $\mu > 0$ are the Lamé coefficients. Then we have the elasticity operater

$$\mathcal{L}(u) = -\nabla \cdot \sigma, \quad \mathcal{B}(u) = \sigma\, n. \tag{2.3}$$

For the fine-scale discretization, we let $\mathcal{T}_h$ be a finite element partition of the domain, and Γ_h be the set of all interior faces. Let e be a face shared by two neighboring fine elements T_1 and T_2, then the jump and average of a vector function u are given by

$$\{u\} = \frac{u|_{T_1} + u|_{T_2}}{2}, \quad [u] = u|_{T_1} - u|_{T_2}. \tag{2.4}$$

The unit normal vector n is defined on e so that it points from T_1 to T_2.

We will use the standard IPDG scheme on the fine grid. Let V_h be the finite element space on the fine grid mesh $\mathcal{T}_h$. The weak formulation of the problem (2.1) is to find $u_h \in V_h$ such that

$$a_{DG}(u_h, v) = (f, v) \tag{2.5}$$

for any $v \in V_h$. The bilinear form a_{DG} is defined as follows:

- For the Laplace operator, we have

$$\begin{aligned} a_{DG}(u, v) = &\sum_{T\in\mathcal{T}^h} \int_K \kappa \nabla u \cdot \nabla v dx - \sum_{e\in\Gamma_h} \int_e \{\kappa \nabla u \cdot n_e\}[v] ds \\ &- \sum_{e\in\Gamma_h} \int_e \{\kappa \nabla v \cdot n_e\}[u] ds + \gamma_{\text{fine}} \sum_{e\in\Gamma_h} \int_e \kappa [u][v] ds, \end{aligned} \tag{2.6}$$

  where $V_h = \{v \in L^2(\Omega^\epsilon) \text{ such that } v|_T \in P_1(T),\ \forall T \in \mathcal{T}_h\}$.

- For the elasticity operator, we have

$$a_{DG}(u,v) = \sum_{K\in\mathcal{T}^h}\int_K (2\mu\varepsilon(u):\varepsilon(v) + \lambda\nabla\cdot u\nabla\cdot v)dx - \sum_{e\in\Gamma_h}\int_e \{\tau(u)\}[v]ds - \sum_{e\in\Gamma_h}\int_e \{\tau(v)\}[u]ds + \gamma_{\text{fine}}\sum_{e\in\Gamma_h}\int_e (\lambda+2\mu)[u][v]ds, \tag{2.7}$$

where $V_h = \{v \in (L^2(\Omega^\epsilon))^2 \text{ such that } v|_T \in (P_1(T))^2,\ \forall T \in \mathcal{T}_h\}$, and $\tau(\cdot) = \sigma(\cdot)\cdot n$.

We can write either of these equation in the matrix form

$$A_{\text{fine}}U_{\text{fine}} = F_{\text{fine}}, \tag{2.8}$$

where A_{fine} is the fine-scale stiffness matrix, F_{fine} is the fine-scale right-hand side and U_{fine} is the fine-scale solution.

For the coarse-scale discretization, we will apply the Generalized Multiscale Finite Element Method (GMsFEM) framework using DG coupling (GMsDGM). The GMsFEM is a generalization of the classical multiscale finite element method by systematically enriching the coarse spaces and taking into account small scale information. The GMsDGM is based on the interior penalty discontinuous Galerkin method. In GMsDGM, we design the snapshot space for each non-overlapping coarse local domain. Then some spectral problems are performed to obtain a reduced order space, which is the offline space. The coarse scale solution of the problems are found in the offline space. In next section, we will intruduce the necessary concepts, notations and consider multiscale space construction.

3. Coarse-scale discretization

Let $\mathcal{T}_H$ be the coarse scale partition of the domain Ω^ϵ which consists of local subdomains K. In the coarse grid mesh, the characteristic mesh size $H >> h$, where h is the size of fine scale mesh and the fine grid is sufficiently fine to resolve the fine-scale features. In Figure 1, we present the local subdomain in structured, quasi-unstructured and unstructured coarse mesh.

In GMsDGM framework we have two main components: the construction of the multiscale basis functions in non-overlapping local domains (offline computaion), and the global coarse-grid level DG coupling. The offline computation mentioned above usually contains two steps: (1) the construction of a shapshot space that will be used to compute an offline space; and (2) the construction of an offline space by performing a dimension reduction in the snapshot space.

For the offline computation, we first design local snapshot space $V_{\text{snap}}(K)$, which is constructed for each local domain $K \in \mathcal{T}_H$. The snapshot space contains a large library of local basis functions, which can be used to obtain a reliable approximation of the fine space. We decompose the local snapshot spaces $V_{\text{snap}}(K)$ into two subspaces:

$$V_{\text{snap}}(K) = V^b_{\text{snap}}(K) + V^i_{\text{snap}}(K),$$

where the first local snapshot space $V^b_{\text{snap}}(K)$ in the coarse grid block K is defined as the span of all harmonic extensions, and the second local snapshot space

$V^i_{\text{snap}}(K)$ is defined as $V^i_{\text{snap}}(K) = V^0_h(K)$, which is the restrictions of V_h on K with zero trace on ∂K. For dimension reduction in the above snapshot spaces, we use solutions of appropriate spectral problems in each snaphot space. Furthermore, eigenfunctions corresponding to the dominant eigenvalues of the spectral problem are used to construct the offline basis functions. The resulting reduced order space is called the local offline space $V_H(K)$ for the coarse-grid block K. We can write

$$V_H(K) = V^b_H(K) + V^i_H(K),$$

where $V^b_H(K)$ and $V^b_H(K)$ contain boundary and interior basis functions, respectively. The global multiscale space V_H is then defined as the sum of all these $V_H(K)$, $K \in \mathcal{T}_H$.

Now, let V_H be the coarse-scale offline space, which consists of functions that are defined on each coarse grid block $K \in \mathcal{T}_H$,

$$V_H = \text{span}\{\phi^{\text{ms}}_{r,i}, \quad r = 1, \cdots, M_r, \quad i = 1, \cdots, N_c\},$$

where M_r is the number of the multiscale basis functions in each K_r and N_c is the number of the local domains. Note that $\phi_{r,i}$ is continuous in coarse element K, but is not necessarily continuous along the coarse edges. We construct the coarse scale system in this offline space using discontinuous Galerkin coupling, where we need to penalize the jump of the solution on the coarse edges. In general, we want to seek an approximation $u_H = \sum_{r,i} c_{r,i}\phi_{r,i}$ in V_H such that

$$a_{DG}(u_H, v) = (f, v), \quad \forall v \in V_H.$$

We note that coarse-scale system can be formed by projecting the fine-scale matrices onto the coarse grid. The projection matrix can be assembled using the multiscale basis functions

$$R = [R_1, R_2, ..., R_{N_c}], \quad R_r = [\phi_{r,1}, \phi_{r,2}, ..., \phi_{r,M_r}],$$

where $r = 1, \cdots, M_r$ and R_r is the local projection matrix in a coarse element K. Using the projection matrix R, we can write the coarse-scale system as follows

$$A_H u_H = F_H, \text{with} \quad A_H = R^T A_{\text{fine}} R \text{ and } F_H = R^T F_{\text{fine}},$$

where A_{fine} and F_{fine} are defined in (2.8). After calculating the coarse-scale solution u_H, we can recover the solution on the fine grid by $u_{\text{ms}} = Ru_H$.

Now we present the details of constructing the boundary basis and interior basis. To construct the local snapshot space $V^b_{\text{snap}}(K)$ for each fine-grid node (vertex) on the boundary of K, we find $\psi_{l,K} \in V_h(K)$ by solving

$$a_{DG}(\psi_{l,K}, v) = 0 \text{ in } K, \qquad \psi_{l,K} = g_l \text{ on } \partial K, \tag{3.1}$$

where $g_l, \quad l = 1, ..., J_K$ are the local boundary conditions. In particular, let δ_l be the delta function such that $\delta_l = 1$ at the lth node of ∂K and $\delta_l = 1$ on the rest nodes of ∂K, where $l = 1, ..., J_K$ with J_K being the total number of boundary nodes. Then for Laplace equation in 2D, we have $g_l = \delta_l$; for Elasticity equation in 3D, we have $g_l = (\delta_l, 0, 0)$ or $(0, \delta_l, 0)$ or $(0, 0, \delta_l)$. The functions $\psi_{l,K}$ defined above are known as harmonic extensions.

The linear span of the harmonic extensions form the local snapshot space

$$V^b_{\text{snap}}(K) = \text{span}\{\psi^b_{1,K}, \ \psi^b_{2,K}, \ ..., \psi^b_{J_K,K}\}, \tag{3.2}$$

where J_K is the number of snapshot basis in local domain K, and the subscript b refers to *boundary*.

We then need to select some important modes in $V^b_{\text{snap}}(K)$ in order to form a reduced order space $V^b_H(K)$. This is done by solving the local spectral problem

$$A\Phi^b = \lambda^b S\Phi^b, \quad [a_{mn}] = a_{DG}(\psi^b_{m,K}, \psi^b_{n,K}), \quad [s_{mn}] = s(\psi^b_{m,K}, \psi^b_{n,K}), \tag{3.3}$$

where $s(\psi^b_{m,K}, \psi^b_{n,K}) = \int_{\partial K} \kappa \psi^b_{m,K}\, \psi^b_{n,K}\, ds$.

We choose the first M^b eigenvectors $\{\Phi^b_1, \Phi^b_2, ..., \Phi^b_{M^b}\}$ corresponding to the first M^b smallest eigenvalues, $\lambda^b_1 \le \lambda^b_2 \le ... \le \lambda^b_{M^b}$. The offline basis are defined using these eigenvectors $\phi^b_{i,K} = \sum_j (\Phi^b_i)_j \psi^b_{j,K}$ where $(\Phi^b_i)_j$ is the jth component of the vector Φ^b_i. Then we obtain the space $V^b_H(K)$ by

$$V^b_H(K) = \text{span}\{\phi^b_{1,K},\ \phi^b_{2,K},\ ..., \phi^b_{M^b,K}\}. \tag{3.4}$$

These multiscale basis functions in $V^b_H(K)$ are called boundary basis functions.

Remark 3.1. The snapshot basis functions descibed above are referred to as the Type 1 ($V^b_{\text{snap}}(K)$).

Remark 3.2. One can also use all the fine grid nodal basis as snapshots. This snapshot space are referred to the Type 2 ($V^b_h(K)$).

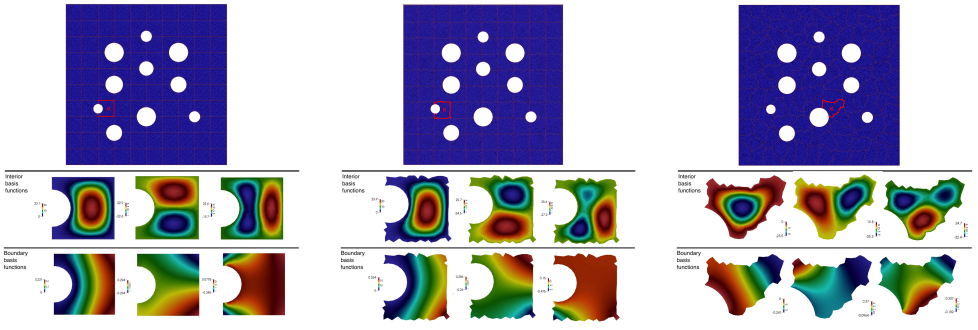

FIGURE 1. Top: Computational meshes for structured (left), quasi-structured (middle) and unstructured (right) coarse meshes with 100 local domains. Bottom: The first three boundary and interior basis functions for the marked coarse region.

The interior basis functions. The local snapshot space $V^i_{\text{snap}}(K)$ for the coarse grid block K is defined as $V^i_{\text{snap}}(K) = V^0_h(K)$. For the dimension reduction on the snapshot space, we use following spectral problem to identify the important modes: find $\phi^i_{m,K}$ in $V^0_h(K)$ satisfying

$$a_{DG}(\phi^i_{m,K}, v) = \lambda^i_m(\phi^i_{m,K}, v), \quad \forall v \in V^0_h(K). \tag{3.5}$$

Note that the subscript i refers to *interior* here. We arrange the eigenvalues in an ascending order $\lambda^i_1 \le \lambda^i_2 \le ... \le \lambda^i_{M^i}$. We select the first M^i eigenfunctions $\phi^i_1, \phi^i_2, ..., \phi^i_{M^i}$ corresponding to the first M^i smallest eigenvalues. The space $V^i_H(K)$ is spanned by these functions

$$V^i_H(K) = \text{span}\{\phi^i_{1,K}, \phi^i_{2,K}, ..., \phi^i_{M^i,K}\}. \tag{3.6}$$

The multiscale basis functions in $V_H^i(K)$ are called interior basis functions.

In Figure 1, we present the first three interior and boundary basis in perforated domain for structured, quasi-structured and unstructured coarse meshes. We also compare the eigenvalues for the interior and boundary basis functions when we use diffrent type of meshes. When we calclulate boundary basis functions using Type 1 (see Remark 3.1) and Type 2 (see Remark 3.2) snapshots, we can observe that the first 30 eigenvalues are similar.

Chung at. al [5] presented a complete stability and convergence analysis of the discontinious multiscale method for elliptic problem (without perforations). We expect a similar behavior for the current algorithm. The error of the multiscale method depends on the coarse-scale mesh size (H), the penalty parameter (γ_c), and the number of interior and boundary basis functions choosen. Let $\Lambda^i = 1/\lambda^i_{M^i+1}$ and $\Lambda^b = 1/\lambda^b_{M^b+1}$, we have

$$\begin{aligned} ||u_h - u_H||^2_{DG} \leq & \sum_{K\in\mathcal{T}_H} \frac{H}{\kappa}\Lambda^b_m \left(1 + \frac{\gamma_c H}{h}\Lambda^b_m\right) \int_{\partial K} |\kappa \nabla u_p \cdot n_{\partial K}|^2 \\ & + \sum_{K\in\mathcal{T}_H} H^2 \Lambda^i_K ||f||^2_{L^2} + O(h), \end{aligned} \tag{3.7}$$

where $\|u\|^2_{DG} = \int_\Omega |\nabla u|^2 + \frac{1}{h}\sum_{E\in\mathcal{E}^H}\int_E [u]^2$ and u_p is the projection of the u_h in the snapshot space.

Remark 3.3. The interior basis functions described above will only be used for the case of the nonzero right hand side in equation (2.1) (see (3.7)).

4. Randomized snapshot space with oversampling

In this section, we present the numerical results for the Laplace problem in perforated domain with some big inclusions and additional small inclusions, and the permeability coefficient is $\kappa = 1$. We remark that we can observe similar behavior in the solutions for the elasticity problem, so we will omit the numerical results for elasticity problem from now on. We introduce the construction of randomized snapshots, and then show the multiscale solution using both standard snapshot and randomized snapshot space. Note that the standard snapshot space means the snapshot we calculated for Type 1 (see in Remark 3.1). We also compare the convergence behavior for both snapshots. We calculate relative errors between coarse-scale u_H and fine-scale u_h solutions using following weighted L^2 and H^1 norms

$$\|u\|^2_{L^2} = \sum_K \int_K \kappa |u|^2, \quad \|u\|^2_{H^1} = \sum_K \int_K \kappa |\nabla u|^2.$$

We consider the following test cases:

- *Case 1:* For source term $f = 1$ and boundary conditions: $u = 0$ on global boundaries and $u = 0$ on boundary of inclusions.
- *Case 2:* For source term $f = 0$ and boundary conditions: $u = 1$ on global boundaries and $u = 0$ on boundary of inclusions.

In the numerical examples, the fine mesh constains 44126 cells and the DOF (degrees of freedom) of fine-scale system is $DOF_{\text{fine}} = 132378$. For the coarse scale approximation, we take structured coarse mesh with 100 local coarse blocks.

In Table 1, we present relative errors for the Laplace problem for *Case 1* and *Case 2* using standard snapshots. We note that for *Case 1* with nonzero source term, besides several boundary basis, we need to add one additional interior basis in order to reduce the errors (see (3.7)). We observe this phenomenon by comparing left column of Table 1 and middle column of Table 1. For example, when we take 16 basis, the L^2 error reduces from 7.3% to 0.9% after adding one interior basis. On the right column of Table 1, we present the convergence history for *Case 2*, and we only take boundary basis functions. In this case, adding interior basis hardly impact the results and can be omitted here. Next, we introduce

Case 1, $M_H^i = 0$			*Case 1*, $M_H^i = 1$			*Case 2*,$M_H^i = 0$		
$M_H^b(DOF_c)$	L^2	H^1	$M_H^b(DOF_c)$	L^2	H^1	$M_H^b(DOF_c)$	L^2	H^1
2 (400)	90.7	94.7	2 (400)	82.7	90.2	2 (200)	26.9	92.2
4 (500)	36.6	53.9	4 (500)	32.7	48.5	4 (400)	5.4	37.4
8 (900)	10.3	31.7	8 (900)	8.3	25.7	8 (800)	1.1	16.7
16 (1700)	7.3	27.3	16 (1700)	0.9	9.7	16 (1600)	0.2	7.0

TABLE 1. Laplace problem for *Case 1* and *Case 2* on structured coarse mesh with 100 local domains. Left: *Case 1* using only boundary basis. Middle: *Case 1* using one additional interior basis. Right: *Case 2* using only boundary basis.

the randomized snapshot and oversampling techniques. The construction of randomized oversampling snapshot can substantially save the computational cost for snapshot calculations. In this algorithm, instead of solving local harmonic problems (3.1) for each fine grid node on the boundary of the local domain, we solve a small number of local harmonic extension problems in oversampled domains with random boundary conditions [4]. More precisely, we let

$$\psi_{j,K^+} = r_j, \quad \text{on} \quad \partial K^+,$$

where r_j are independent identical distributed standard Gaussian random vectors on the fine grid nodes of the boundary. Note that when we use randomized snapshots, we only generate a fraction of the snapshot vectors by using random boundary conditions. This can substantially reduce the size of the snapshot sapce. The oversampling techniques are also employed, that is, we solve local problems in the extended local domain $K^+ = K + m$, where m is the number of the fine cell layers that are added to the original local domain K. The oversampling strategy is used to reduce the boundary effects in the construction of randomized snapshot basis functions. Fine-scale and coarse-scale solutions using standard and randomized snapshot spaces on the structured coarse mesh are presented in Figure 2. We observe good coarse-scale solutions for both snapshot spaces.

The L^2 and H^1 relative errors for randomized snapshot space are presented in Table 2 for different sizes of the snapshot space using oversampling techniques. We report the results when the size of the randomized snapshots is 25%, 30% and 35% of the size of the standard snapshot. The results for oversampling with 2 fine cells layers are shown on the top of each tables, the results for oversampling with 4 fine layers are shown on the bottom part. We observe that the errors are comparable when we use standard or randomized snapshots. For example, in

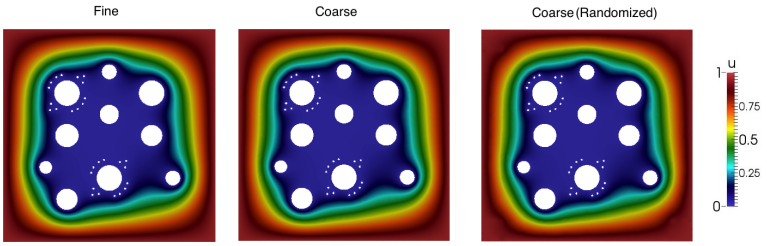

FIGURE 2. Laplace problemusing structured coarse mesh with 100 local domains for *Case 2* ($f = 0$). $DOF_{\text{fine}} = 132378$ and $DOF_c = 2000$ ($M_H^b = 20$). Left: Fine-scale solution. Middle: Coarse-scale solution using standard snapshot space (0.1% L^2-error). Right: Coarse-scale solution using randomized snapshot space with oversampling $K + 4$(1.9% L^2-error).

the *Case 2* ($f = 0$) for coarse scale system with size 2000, coarse-scale solution have 0.1% of L^2 error for standard snapshot space and 1.9% for the randomized snapshot space (the size of the randomized snapshots is 35% of whole snapshot space) with oversampling $K^+ = K+4$ when we take 20 boundary basis functions. We note that as we increase the size of randomized snapshot spaces or enlarge the oversampling size of local domain, we can get better results. The comparisons are shown in Figure 3.

25%			*30%*			*35%*		
$M_H^b(DOF_c)$	L^2	H^1	$M_H^b(DOF_c)$	L^2	H^1	$M_H^b(DOF_c)$	L^2	H^1
$K^+ = K + 2$			$K^+ = K + 2$			$K^+ = K + 2$		
2 (200)	29.4	-	2 (200)	31.9	-	2 (200)	32.5	-
4 (400)	17.6	68.4	4 (400)	10.9	53.5	4 (400)	8.3	46.3
8 (800)	15.3	52.5	8 (800)	6.5	35.5	8 (800)	4.5	31.5
16 (1600)	12.6	45.8	16 (1600)	5.5	27.4	16 (1600)	3.4	19.9
$K^+ = K + 4$			$K^+ = K + 4$			$K^+ = K + 4$		
2 (200)	33.7	-	2 (200)	33.9	-	2 (200)	33.8	-
4 (400)	8.3	44.9	4 (400)	8.2	44.3	4 (400)	6.9	53.9
8 (800)	5.0	33.4	8 (800)	5.2	33.1	8 (800)	4.3	31.7
16 (1600)	3.4	17.5	16 (1600)	3.0	16.5	16 (1600)	2.1	14.9

TABLE 2. Laplace problem for *Case 1* using randomized snapshot space on structured coarse mesh with 100 local domains. Oversampling with 2 and 4 fine layers.

We conclude that in perforated domain with additional small inclusions, when we use randomized snapshot space with oversampling (see Figure 3): (1) we can get solutions with a desired accuracy using randomized snapshots; (2) oversampling randomized snapshot space can reduce the model size and therefore save the computational cost for snapshot calculations.

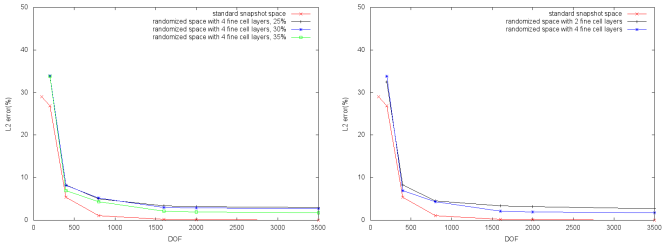

FIGURE 3. The L^2 relative errors for different coarse system on structured coarse mesh with 100 local domains for *Case 2*: Left: error comparision for the different size of randomized snapshot space with oversampling domain $K^+ = K+4$; Rrght: errors for the 35% randomized snapshot space for different size of oversampling layer $K^+ = K + 2$ and $K^+ = K + 4$.

5. Numerical examples in arbitrary perforated domain or high-constrast medium

In this section, we consider a multiscale solution using arbitrary geometry or medium. We generate the geometry/medium with random distribution of the inclusions/particles, given ratio between volume of inclusions and domain volume. Note that the inclusions/particles can have different sizes. For the coarse discretization in this section, we consider to use structured, quasi-structured and unstructured coarse meshes. We use open-source Gmsh sofware [12] for mesh constructions.

For the structured coarse mesh, we refine the coarse mesh and obtain the fine mesh which is campatible with the coarse edges. For the construction of the unstructured coarse mesh, we use Metis graph partitioner [14]. Usually our graph partitioners aim at: (1) obtain similar size in each partition of blocks (local domains), and (2) minimize the number of edges on the boundary of each block to reduce the communication for standard parallel solvers. Numerical implementation of the fine-scale and multiscale solvers are based on the FEniCS library [16].

We will consider numerical example both in perforated domains and high-contrast medium. In Figure 4, we show interior and boundary basis functions for perforated domain (left) and high contrast medium (right) in unstructured coarse mesh with 100 local domains. In this example, we use boundary basis functions of Type 2 (see Remark 3.2).

5.1. Perforated domains with random distribution of the inclusions. We consider similar test cases *Case 1* and *Case 2* as in previous sections for Laplace problem. We take $\kappa = 1$ in the perforated domain with random distribution of inclusions. The fine mesh constains 38006 cells. For the coarse mesh, we test with quasi-structured and unstructured meshes with $25, 50$ and 100 local domains, but only show the error history for 100local domains.

First, we will compare the results when we use quasi-unstructured mesh and unstructured mesh for the same geometry (see left of Figure 4). In Table 3,

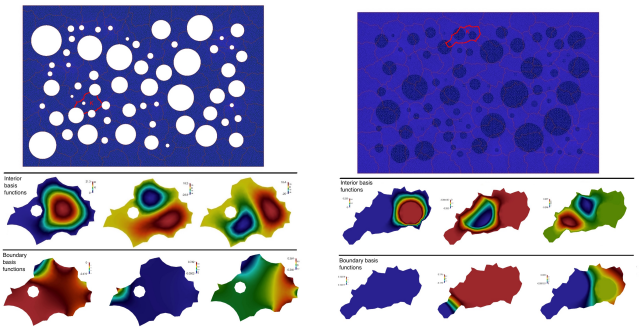

FIGURE 4. Multiscale basis functions in perforated (left) and hign-contrast (right) domains with 100 local domains in unstructured coarse mesh.

we present the L^2 and H^1 relative errors for *Case 1* and *Case 2* using quasi-unstructured (left of Table 3) amd unstructured (right of Table 3) coarse meshes with 100 local coarse domains. We omit the error tables when we use 25 and 50 local domains, and remark that errors can be reduced as we take larger number of local domains. In particular, for *Case 1*, when we take 20 boundary basis functions and 1 interior basis function using unstructutred coarse mesh, the L^2 error is 20.8% for 25 local domains, and 10.9% L^2 error for 50 local domains, and we can reduce error to 6.7% for 100 local domains. For *Case 2*, we can observe generally better results compared with *Case 1*. This behavior can be explained by the convergence analysis (3.7). Also in *Case 2*, the number of local domains has less influence on the errors. Compared with quasi-structured and unstructured coarse meshes, we obtain similar results. Thus for the numerical examples in the following sections, we will use unstructured coarse meshes due to they are easier to generate.

100 local domains		
$M_H^b(DOF_c)$	L^2	H^1
Case 1, $M_H^i = 1$		
2 (200)	67.5	79.6
4 (400)	49.0	62.7
8 (800)	21.2	37.9
20 (2000)	7.8	23.6
Case 2, $M_H^i = 0$		
2 (300)	25.8	66.7
4 (500)	18.8	49.7
8 (900)	5.4	25.4
20 (2100)	0.9	10.5

100 local domains		
$M_H^b(DOF_c)$	L^2	H^1
Case 1, $M_H^i = 1$		
2 (200)	73.0	81.7
4 (400)	53.1	64.9
8 (800)	24.4	40.0
20 (2000)	6.7	22.3
Case 2, $M_H^i = 0$		
2 (300)	25.8	66.7
4 (500)	18.8	49.7
8 (900)	5.4	25.4
20 (2100)	0.9	10.5

TABLE 3. Laplace problem in domain with random inclusions. Numerical results for *Case 1* (top) and *Case 2* (bottom) using coarse grid with 100 local domains. $DOF_{\text{fine}} \approx 114018$. Left: quasi-unstructured mesh. Right: unstructured mesh.

Next, we consider the domains with different ratio m between volume of inclusions and the volume of the whole domain. We take $m = 0.35$ and 0.2.

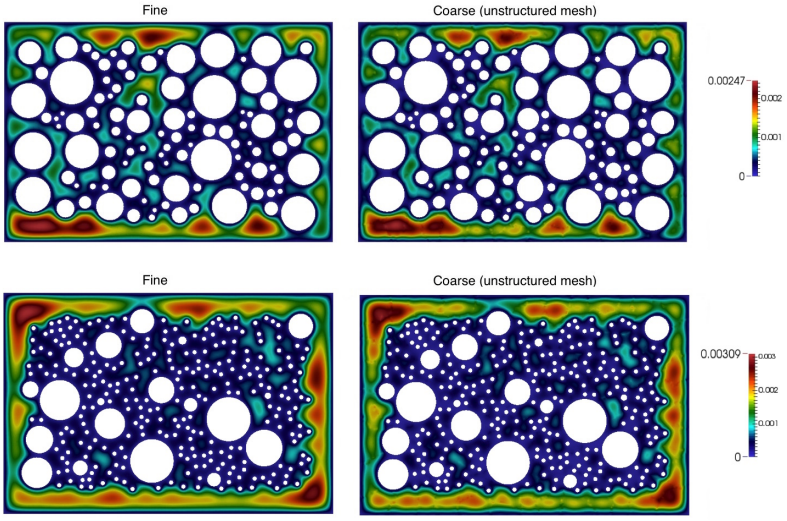

FIGURE 5. Laplace problem in domain with random inclusions for $m = 0.35$ and 0.2. Numerical solutions for *Case 1* using unstructured coarse grid with 100 local domains. Coarse-scale solutions are performed using 20 boundary basis functions and one interior basis function, $DOF_c = 2100$. Top: Fine and coarse solution for $m = 0.35$, $DOF_{\text{fine}} = 103692$, L^2-error is 8.0%. Bottom: Fine and coarse solution for $m = 0.2$, $DOF_{\text{fine}} = 139380$, L^2-error is 8.0%.

In Figure 5, we depict the numerical results for *Case 1* ($f = 1$) for two different geometries with $m = 0.35$ and 0.2. We observe good accuracy for these cases when the multiscale system is only 1.7% of fine-scale system size (when the number of basis $M_H^b = 20$ and $M_H^i = 1$).

In Table 4, we present errors for *Case 1* and *Case 2*, when we take m equal 0.35 and 0.2 respectively. Note that in these examples, we use unstructured coarse meshes. Remark that we also test for $m = 0.14$ but we will omit the solution figure and the error table for this case, we only present error behavior of this case in Figure 6. For *Case 1* we observe that the increasing of ratio between volume of inclusions and domain volume leads to the increasing of the errors, and we can reduce errors by taking more local domains. For *Case 2*, the accuracy of multiscale solution depends not much less on the m and additional small inclusions.

From the presented numerical results for perforated domains, we get following conclusions:

- As we mentioned, we can get more accurate solutions using the structured coarse mesh. However, the difficulty lies in generating the corresponding fine mesh, which are compatible with the coarse edges. The quasi-structured and unstructured meshes can give us a good accuracy

100 local domains		
$M_H^b(DOF_c)$	L^2	H^1
Case 1, $M_H^i = 1$		
4 (400)	49.5	62.1
8 (800)	22.3	39.5
16 (1600)	9.8	27.8
20 (2000)	8.0	25.4
Case 2, $M_H^i = 0$		
4 (500)	14.7	38.7
8 (900)	4.7	22.0
16 (1700)	1.2	11.2
20 (2100)	0.8	8.1

100 local domains		
$M_H^b(DOF_c)$	L^2	H^1
Case 1, $M_H^i = 1$		
4 (400)	53.0	66.6
8 (800)	27.5	44.5
16 (1600)	10.4	28.8
20 (2000)	8.0	26.1
Case 2, $M_H^i = 0$		
4 (500)	14.0	46.3
8 (900)	4.7	24.7
16 (1700)	1.1	12.5
20 (2100)	0.8	10.4

TABLE 4. Laplace problem in domain with random inclusions for $m = 0.35$ (left, $DOF_{\text{fine}} = 103692$) and $m = 0.2$ (right, $DOF_{\text{fine}} = 139380$). Numerical results for *Case 1* (top) and *Case 2* (bottom) using unstructured coarse grid.

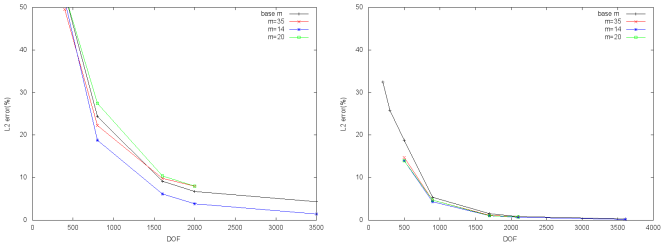

FIGURE 6. The L^2 relative errors for different size of the coarse-scale system for *Case 1* (left) and *Case 2* (right) for different perforated domains.

in solutions and they are much easier to construct. The quasi-structured coarse mesh works slightly better than the unstructured mesh.

- For unstructured coarse mesh we can easily refine coarse mesh and obtain more accurate results. However, when we increase the number of the local domains we also increase the size of coarse-scale system.
- L^2 and H^1 relative errors are similar in *Case 2*, when the domains have different ratios between volume of inclusions and the volume of whole domain (see the left of the Figure 6). Taking 20 boundary basis functions in unstructured coarse mesh with 50 local domains, we obtain 1.0% of L^2 error for $m = 0.35$, 1.1% for $m = 0.2$.
- For *Case 1*, L^2 and H^1 relative errors are very sensitive to the geometry (see the left of the Figure 6). The errors are increasing when we enlarge the ratio between volume of inclusions and domain volume. For 20 boundary basis functions and one interior basis function using coarse unstructured coarse mesh with 50 local domains, we obtain 16.2% of L^2 error for $m = 0.35$ and 12.9% for $m = 0.2$.

5.2. High-contrast medium with random distribution of particles. In this section, we consider the numerical results for Laplace problem with source term $f = 1$ or $f = 0$ and zero Diriclet boundary condition in high-contrast medium:

- *Case 3*: For high-constrast domain with $\kappa = 1$ on background, $\kappa = 10^4$, 10^3 and 10^2 for different types of high-permeability particles.

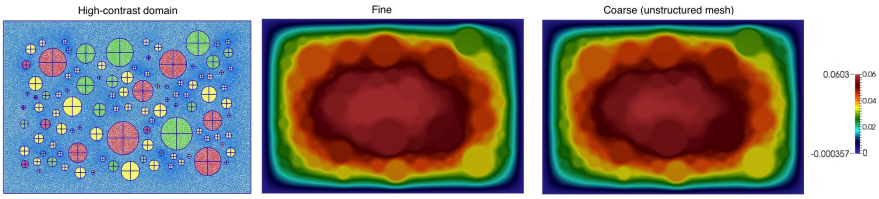

FIGURE 7. Laplace problem in high-constrast domain(left) in the *Case 3* with $f = 1$. Fine-scale solution (middle) and coarse-scale solution (right) using unstructured coarse grid with 50 local domains. Coarse-scale solutions are performed using 20 boundary basis functions and one interior basis function $DOF_c = 1800$ with 3.7% of L^2-error. $DOF_{\text{fine}} = 159966$.

$f = 1$, $M_H^i = 1$			$f = 0$, $M_H^i = 0$		
$M_H^b(DOF_c)$	L^2	H^1	$M_H^b(DOF_c)$	L^2	H^1
4 (250)	87.8	91.8	4 (200)	13.0	91.6
16 (850)	13.4	27.8	16 (800)	2.3	27.9
35 (1800)	3.7	15.8	35 (1750)	0.8	16.4

TABLE 5. Laplace problem for high-contrast domain for *Case 3* using unstructured coarse mesh. Numerical results for $f = 1$ (top) and $f = 0$ (bottom) using unstructured coarse mesh. $DOF_{\text{fine}} = 139380$. Left: 25 local domains. Middle: 50 local domains. Right: 100 local domains.

In Figure 7, we show fine-scale and coarse scale solutions for *Case 3*. In Tables 5, we present errors for *Case 3* with $f = 1$ and $f = 0$ using unstructured coarse mesh. The results show that our proposed algorithm works efficiently for problems with high-contrast medium. Similarly as in the previous sections, when we take $f = 1$, we need to add some interior basis to reduce the errors. For $f = 0$, only taking boundary basis can give us good results. We also test *Case 3* using different distributions of particles in the high-contrast medium with various high-permeability values, our results show equally good behaviors.

6. Numerical results for three-dimensional problems

In this section, we present numerical results for the Laplace problem in perforated domains for 3D case and discuss computational costs. We consider problems

with zero force term and $\kappa = 1$. We use the following Dirichlet boundary conditions: $u = 1$ on the global boundary and $u = 0$ on the boundary of inclusions. As for penalty term, we use $\gamma_{\text{fine}} = 4.0$ for fine-scale calculation, $\gamma_{\text{fine}} = 0$, and $\gamma_c = 4.0$ for coarse-scale calculation. For coarse-scale calculation we consider structured and unstructured coarse meshes (see Figure 8. For the construction of the unstructured coarse mesh, we use Metis graph partitioner [Metis].

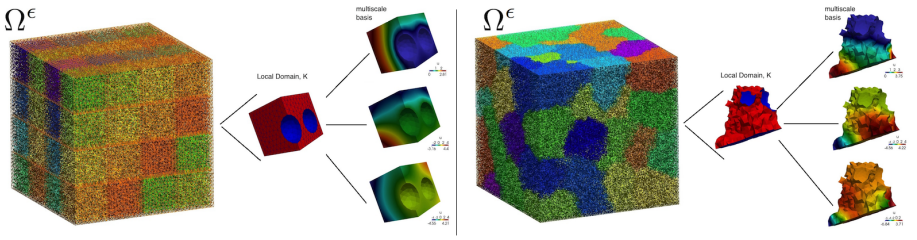

FIGURE 8. Structured (left) and untructured (right) coarse mesh in 3D. Global (Ω_ϵ) and local (K) domains and corresponding first 3 boundary basis functions in K

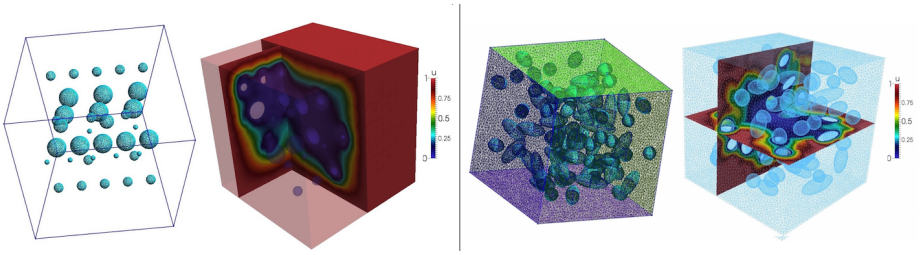

FIGURE 9. Perforated domain and coarse-scale solution using structured coarse mesh with 512 local domains (left) and unstructured coarse mesh with 216 local domains (right).

In Figure 9, we present the 3D domain and the numerical solution using multiscale solver. The coarse mesh with 512 local domains. For construction of the geometry we used Netgen software. Fine-scale mesh contains $N_c = 390756$ cells and size of the fine-scale system is $DOF_{\text{fine}} = 1588356$. The solution was obtained using 40 boundary basis functions, where the coarse-scale solver has $DOF_c = 20480$ (1.2% from DOF_{fine}).

At offline stage, we compute multiscale basis function in each local domain (coarse cell). Since calculations in each local domain can be performed independently, we can calculate multicale basis functions in parallel by run the program with multiple processers whose number is equal to number of the local domains. For numerical solution in Figure 9, we use 512 local domains. Computational time for construction of the multiscale basis functions in one local domains is 17.3 seconds (local domain contains $N_c = 787$ cells). If we divide fine mesh into 216 local domains, the computational time increases twice to 35.9 seconds (local domain contains $N_c = 1548$ cells). For 128 and 64 local domains, we spend 87.6 seconds (local domain contains $N_c = 3112$ cells) and 184.4 seconds (local

domain contains $N_c = 6327$ cells), respectively. After one online computation, we construct (initiating time) and solve (solving time) coarse-scale system. Solving time for the coarse-scale system on coarse mesh with 512 local domains using 20 multiscale basis functions is 17.5 seconds and initiating time is 59.6 seconds. When we increase the number of the multiscale basis functions to 40, the initiating time increases twice, but solving time stays the same. Remark that the size of the coarse-scale system is equal to the number of local domains multiplying by number of basis functions.

Numerical solution for the unstructured coase mesh is presented in Figures 9. The coarse mesh contains 216 local domains and fine mesh contains $N_c = 1005559$ cells. Coarse-scale system have size $DOF_c = 4320$ and fine-scale system size is $DOF_{\text{fine}} = 4022236$. Multiscale basis construcion in one local domains with $N_c = 4559$ cells takes 14.3 second (offline stage) and is performed in parallel. Initialializing and solving the coarse-scale system takes 126 and 46 seconds, respectively.

In general, we can improve the accuracy of our approaches by using adaptivity [5] and online method [6]. One can find some related works in the review paper [7]. We can also use parallel computations in assembling and solving coarse scale system.

7. Conclusion

In this paper, we consider multiscale problems in perforated domains. Our aim is to solve the problem on a coarse grid, where each coarse grid contains many perforations. We use Generalized Multiscale Finite Element framework to construct multiscale basis functions in each coarse block. These multiscale basis functions are coupled via IPDG framework, which allows parallel computations of basis functions and avoiding parition of unity functions. Because of the latter, one can easily impose various boundary conditions in each coarse domain. We present numerical results for 2D and 3D cases.

References

[1] J.E. Aarnes, Y. Efendiev, and L. Jiang. Analysis of multiscale finite element methods using global information for two-phase flow simulations. *SIAM J. Multiscale Modeling and Simulation*, 7:2177–2193, 2008.

[2] G. Allaire and R. Brizzi. A multiscale finite element method for numerical homogenization. *SIAM J. Multiscale Modeling and Simulation*, 4(3):790–812, 2005.

[3] Donald L Brown and Daniel Peterseim. A multiscale method for porous microstructures. *arXiv preprint arXiv:1411.1944*, 2014.

[4] Victor M. Calo, Yalchin Efendiev, Juan Galvis, and Guanglian Li. Randomized oversampling for generalized multiscale finite element methods. *Multiscale Modeling & Simulation*, 14(1):482–501, 2016.

[5] E. Chung, Y. Efendiev, and W. T. Leung. An adaptive generalized multiscale discontinuous galerkin method (GMsDGM) for high-contrast flow problems. *preprint, available as arXiv:1409.3474*, 2014.

[6] E. T. Chung, Y. Efendiev, and W. T. Leung. Residual-driven online generalized multiscale finite element methods. *Journal of Computational Physics*, 302:176–190, 2015.

[7] Eric Chung, Yalchin Efendiev, and Thomas Y Hou. Adaptive multiscale model reduction with generalized multiscale finite element methods. *Journal of Computational Physics*, 320:69–95, 2016.

[8] Eric T Chung, Maria Vasilyeva, and Wang Yating. A conservative local multiscale model reduction technique for stokes flows in heterogeneous perforated domains. *arXiv preprint arXiv:1608.07268*, 2016.

[9] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized multiscale finite element methods (GMsFEM). *Journal of Computational Physics*, 251:116–135, 2013.

[10] Y. Efendiev, J. Galvis, and X. H. Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, 230(4):937–955, 2011.

[11] Yalchin Efendiev, Juan Galvis, R Lazarov, M Moon, and Marcus Sarkis. Generalized multiscale finite element method. symmetric interior penalty coupling. *Journal of Computational Physics*, 255:1–15, 2013.

[12] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[13] Oleg Iliev, Ralf Kirsch, Zahra Lakdawala, Stefan Rief, and Konrad Steiner. Modeling and simulation of filtration processes. In *Currents in Industrial Mathematics*, pages 163–228. Springer, 2015.

[14] George Karypis and Vipin Kumar. Metis–unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

[15] L. Le Bris, F. Legoll, and A. Lozinski. An MsFEM type approach for perforated domains. *Multiscale Modeling & Simulation*, 12(3):1046–1077, 2014.

[16] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

Eric T. Chung

Department of Mathematics, The Chinese University of Hong Kong (CUHK), Hong Kong SAR.

E-mail address: tschung@math.cuhk.edu.hk

Yalchin Efendiev

Department of Mathematics & Institute for Scientific Computation (ISC), Texas A&M University, College Station, TX 77843-3368.

E-mail address: efendiev@math.tamu.edu

Maria Vasilyeva

Department of Computational Technologies, Institute of Mathematics and Informatics, North-Eastern Federal University, Yakutsk, 677980, Republic of Sakha (Yakutia), Russia

Institute for Scientific Computation, Texas A&M University, College Station, TX 77843.

E-mail address: vasilyevadotmdotv@gmail.com

Yating Wang
Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA.
E-mail address: wytgloria@gmail.com

Received: September 15, 2016; Accepted: October 11, 2016