A conservative multiscale method for stochastic highly heterogeneous flow

Yiran Wang∗, Eric Chung† and Shubin Fu‡

March 23, 2022

Abstract

In this paper, we propose a local model reduction approach for subsurface flow problems in stochastic and highly heterogeneous media. To guarantee the mass conservation, we consider the mixed formulation of the flow problem and aim to solve the problem in a coarse grid to reduce the complexity of a large-scale system. We decompose the entire problem into a training and a testing stage, namely the offline coarse-grid multiscale basis generation stage and online simulation stage with different parameters. In the training stage, a parameter-independent and small-dimensional multiscale basis function space is constructed, which includes the media, source and boundary information. The key part of the basis generation stage is to solve some local problems defined specially. With the parameter-independent basis space, one can efficiently solve the concerned problems corresponding to different samples of permeability field in a coarse grid without repeatedly constructing a multiscale space for each new sample. A rigorous analysis on convergence of the proposed method is proposed. In particular, we consider a generalization error, where bases constructed with one source will be used to a different source. In the numerical experiments, we apply the proposed method for both single-phase and two-phase flow problems. Simulation results for both 2D and 3D representative models demonstrate the high accuracy and impressive performance of the proposed model reduction techniques.

1 Introduction

Many problems of fundamental and practical importance have multiple scales and high contrast properties. For instance, porous media, sedimenting suspensions and fluidized beds exhibit multiscale nature. Solving these problems in a fine grid that is sufficient to capture all small scale information is numerically expensive. Besides, when uncertainty is incorporated in physical models as well as numerical simulations, one needs to parameterize the partial differential equations with some random variables. Using sampling methods, one needs to call a deterministic solver once for each sample. Hence, developing an efficient deterministic fast solving algorithm is indispensable especially when we need to solve the problems repeatedly with a large number of samples. Because of the above considerations, some model reduction methods are necessary. In the past several decades, many researchers proposed various model reduction techniques, for example, upscaling methods [6, 29, 22] and homogenization [12]. These methods aim to obtain a reduced model and one can therefore solve the problem in a coarse grid. However, some important small scale information is hard to be incorporated in the reduced models, which may hinder producing an accurate approximation. Multiscale methods [25, 18, 23, 30, 10, 38, 19], on the other hand, are...
motivated to capture small-scale information of the underlying heterogeneous media in multiscale bases. Even using only a limited number of multiscale bases, the reduced-order solution can still have relatively good accuracy. In other words, multiscale methods do well in attaining a trade-off between accuracy and computation costs. Although one need first generate multiscale basis functions before the online simulation, one could efficiently solve the problem on a coarse grid and thus save online computational cost substantially. In addition to the spatial heterogeneities, here we also aim to construct coarse basis functions that are independent of the stochastic parameters so we do not need to recompute the multiscale bases.

In subsurface modeling, local mass conservation is vitally important for the transportation of the solute. To this end, mixed multiscale finite element methods [1, 15, 2, 20], mortar multiscale methods [15, 8, 33, 4], finite volume methods [31, 43, 24, 44] and some kinds of post-processing approaches [35, 2, 42] are proposed. Among these mass conservative multiscale methods, the mixed multiscale finite element method (MMsFEM) [15] has been successfully applied for various types of flow simulations. In MMsFEM, one velocity multiscale basis is constructed via solving a locally defined flow problem while piecewise constant functions defined in each coarse-element are utilized for the pressure. Although the MMsFEM succeeds in many scenarios, it fails to deal with very complicated porous media, and this motivates the development of the mixed generalized multiscale finite element method (MGMsFEM) [20, 41]. In MGMsFEM, multiple multiscale basis functions for velocity are constructed and thus can capture more complicated media information.

The construction of generalized multiscale space contains two steps. First, one needs to solve a series of local problems with a zero Neumann boundary condition to obtain some local snapshot spaces that include all possible local solutions. We then extract the dominant modes of the rich snapshot space by solving well-designed spectral problems which are motivated by analysis. The final multiscale basis function space is the direct sum of all local multiscale spaces. It is worth mentioning that the construction of local basis function in different local regions are independent of each other, therefore parallel computation can be easily adopted and thus these stages are cheap. These multiscale basis functions include important local permeability information and can generate a good coarse-grid solution without using too many computing resources. However, it does not include global media and source information and this brings about a precision limitation [21, 11, 18]. The residual-driven multiscale basis functions were then proposed [21, 11, 18] to improve the accuracy. These residual-driven multiscale basis functions are computed via solving local problems using local residuals which means they include global media information, boundary condition information and source information. Besides, one can iteratively construct multiple local residual-driven bases and it is shown that with only a small number of these bases one can obtain highly accurate coarse-grid solutions. The oversampling technique [46] can be utilized to further boost the performance of coarse-grid approximation. We will follow the main steps in [46] to generate efficient multiscale basis functions for coarse-grid simulations of the stochastic flow problems. The basis space will include both the local permeability dependent multiscale basis functions and residual-driven multiscale basis functions. Both these two types of multiscale basis functions are constructed before the repeated online simulations and can be parallelized without too many difficulties.

To deal with the uncertainty in the media, we use the Karhunen-Loève (KL) expansion [28] in a reversed fashion. In particular, one can parameterize the stochastic permeability fields with a set of random variables. This process is divided into three steps. First, we choose a covariance function. Then, a spectral problem is solved to obtain some eigenfunctions corresponding to the dominating eigenvalues. The final stochastic field is generated from combinations of the selected eigenfunctions, where the same number of random variables serve as coefficients. Since a minority of eigenfunctions can contain most energy, one can obtain a low-rank representation of stochastic field. Based on this design, a sample permeability field will be straightly attained using a sample coefficient. Even though the number of samples is significantly reduced, repeatedly construction of multiscale bases for each sample is not judicious. To handle this challenge, many efforts can be found in literature [32, 40, 33, 26, 47, 27, 35, 13, 14, 31, 8, 37]. Here we apply the MGMsFEM with a representative
permeability field to construct a parameter independent multiscale space. The involved field is also called the training permeability field. Once the multiscale space is constructed, one could make use of the approximation space to cheaply solve the flow problems corresponding to different samples on a coarse grid. We rigorously analyze the proposed method by deriving a bound for the generalization error. We show that this error is a consequence of different sources and permeability fields between training and testing stages. To verify theoretical convergence results, we test the proposed method for both single-phase flow and two-phase flow simulations. For the two-phase model, we use the finite volume method to solve the cheap transport equation on a fine grid and efficiently solve the elliptic problem on a coarse grid with parameter independent multiscale basis functions. We will predict the water cut in the two-phase model, which is a significant quantity in real applications.

We demonstrate the efficiency of our methods using five benchmark models, where both 2D and 3D media are considered. The observed error decay behavior confirms our theoretical analysis and our method can fast generate reliable coarse-grid solutions.

The paper is organized as follows. We first present some preliminaries in Section 2 and then describe the construction of multiscale basis functions in details in Section 3. Section 4 is devoted to analyze the proposed local model reduction method. The numerical experiments are provided in Section 5, where we will present simulation results for both the single-phase and two-phase flow models. The paper is concluded in last Section.

2 Preliminaries

In this section, we will give some notations that will be used in the following presentations. We consider the following problem in a given domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ and a sample space $\Omega_r$:

\[
\kappa(x; \omega)^{-1} v + \nabla p = 0, \quad x \in \Omega, \quad \omega \in \Omega_r,
\]

\[
\text{div}(v) = f(x), \quad x \in \Omega,
\]

with zero Neumann boundary condition $v \cdot n = 0$. Here $\kappa(x; \omega)$ is a possibly highly heterogeneous random permeability field with high contrast. In particular, define $\kappa_{\text{contra}} = \frac{\max_{x \in \Omega} \kappa(x; \omega)}{\min_{x \in \Omega} \kappa(x; \omega)}$ and $\kappa_{\text{contra}} \gg 1$. We let $f$ be a given source term.

First of all, we define the following inner products and norms.

\[
\langle p, q \rangle = \int_{\Omega} pq, \quad A(u, v) = \int_{\Omega} (\kappa(x; \omega))^{-1} u \cdot v,
\]

\[
\|p\|_{L^2}^2 = \int_{\Omega} |p|^2, \quad \|u\|_{a}^2 = A(u, u).
\]

Then, we introduce the coarse and fine grids. Let $T_H$ be the coarse mesh which partitions $\Omega$ into a set of disjoint coarse-scale elements $K_i$ with diameter $H$. In particular, we have $\Omega = \bigcup_{i=1}^{N_{e,c}} K_i$, where $N_{e,c}$ is the number of coarse-scale elements. Define $E_i$ to be a coarse-scale edge of coarse-scale element $K_i$ if $E_i = \partial K_i \cap \partial K_j$ or $E_c = \partial K_i \cap \partial \Omega$. Let $E_c(K_i)$ be the set of all the coarse-scale edges of $K_i$. Then we define $E_c = \bigcup_{i=1}^{N_{e,c}} E_c(K_i)$. We further define a coarse neighborhood $D_i$ to be a combination of two coarse-scale elements sharing a common coarse-scale edge. More specifically, we define

\[
D_i := \{ K \in T_H : E_i \subset \partial K \}, \quad i = 1, \ldots, N_{E,c},
\]

where $N_{E,c}$ is the number of coarse-scale edges. Moreover, we define an oversampled coarse neighborhood corresponding to a coarse-scale edge $E_i$ by $D_i^+$, which is constructed by enlarging $D_i$ in some way. An example is shown in Figure. Based on the coarse grid $T_H$, we further construct a fine grid $T_h$. In particular, for each coarse-scale element $K_i$, we partition it into some fine-scale elements with a fine mesh size $h$ and we denote the mesh by $T_h(K_i)$. Then $T_h = \bigcup_{i=1}^{N_{e,c}} T_h(K_i)$. Correspondingly,
we define $E_f(K_i)$ to be all fine-scale edges in the partition $\mathcal{T}_h(K_i)$. In addition, let $E'_f(K_i)$ be the interior edges in $\mathcal{T}_h(K_i)$. And $E_f = \cup_{i=1}^{N_e} E_f(K_i)$. We illustrate the above notations in Figure 1.

The variational formulation of the exact problem (1)-(2) is: find $v \in (L^2(\Omega))^2$ and $p \in H^1_0(\Omega)$ such that

$$\int_{\Omega} \kappa^{-1} v \cdot u - \int_{\Omega} \text{div}(u)p = 0, \; \forall u \in (L^2(\Omega))^2, \quad (5)$$

$$\int_{\Omega} \text{div}(v)q = \int_{\Omega} fq, \; \forall q \in H^1_0(\Omega), \quad (6)$$

with zero Neumann boundary condition $v \cdot n = 0$ on $\partial \Omega$. To solve $v$ and $p$ in a finite-dimensional space, we first introduce the fine-grid space. Let $V_f$ be the lowest Raviart Thomas vector field ($RT_0$) and $Q_f$ be space of piecewise constant functions on the fine mesh. We set $V_f = \text{span}\{\psi_{f,1}, \cdots, \psi_{f,N_{e,f}}\}$ and $Q_f = \text{span}\{p_{f,1}, \cdots, p_{f,N_{e,f}}\}$, where $N_{e,f}$ and $N_{e,f}$ are the numbers of inner edges and blocks on the fine grid respectively. $V_f^0 := \{v \in V_f : v \cdot n = 0 \text{ on } \partial \Omega\}$. We obtain the fine-grid solution (reference solution) $(v_f, p_f)$ by solving the following system.

$$A(v_f, w) - \langle \text{div}(w), p_f \rangle = 0, \; \forall w \in V_f^0,$$

$$\langle \text{div}(v_f), q \rangle = \langle f, q \rangle, \; \forall q \in Q_f,$$

(7) where $v_f \cdot n = 0$ on $\partial \Omega$.

We further introduce the multiscale space $(V_{ms}, Q_{ms})$. Here $Q_{ms}$ is the pressure multiscale space spanned by piecewise constant functions on each coarse-scale element, and $V_{ms}$ is the velocity multiscale space. To construct $V_{ms}$, we first construct a snapshot space $V_{snap}$ and then reduce it to a lower-dimensional approximation space $V_{ms,0}$ by some well-designed local spectral problems. Finally, we perform the residual-driven enrichment iterations on $V_{ms,0}$ to get the final multiscale space $V_{ms}$. We define $V_{ms,k}$ to be the multiscale space on the enrichment level $k$.

After we obtain the $(V_{ms}, Q_{ms})$, we seek the multiscale solution $(v_{ms}, p_{ms})$ by solving the following system.

$$A(v_{ms}, u) - \langle \text{div}(u), p_{ms} \rangle = 0, \; \forall u \in V_{ms},$$

$$\langle \text{div}(v_{ms}), q \rangle = \langle f, q \rangle, \; \forall q \in Q_{ms},$$

(8)
where \( \mathbf{v}_{\text{ms}} \cdot \mathbf{n} = 0 \). In the following part, we will rewrite (7) and (8) in a matrix formulation. Let \( \mathbf{v}_{f,v} \) and \( p_{f,v} \) to be the corresponding coefficient vector for reference solution \( \mathbf{v}_f \) and \( p_f \). \( A_f[i,j] = \mathcal{A}(\psi_{f,i}, \psi_{f,j}) \) and \( B_f[i,j] = \langle \text{div}(\psi_{f,i}), p_{f,j} \rangle \). Then we obtain \( \mathbf{v}_{f,v} \) and \( p_{f,v} \) by solving

\[
\begin{bmatrix}
    A_f & B_f^T \\
    B_f & 0
\end{bmatrix}
\begin{bmatrix}
    \mathbf{v}_{f,v} \\
    p_{f,v}
\end{bmatrix} =
\begin{bmatrix}
    \mathbf{0} \\
    F_f
\end{bmatrix}
\]

Similarly, let \( \mathbf{v}_{c,v} \) and \( p_{c,v} \) to be the corresponding coefficient vectors for multiscale solution \( \mathbf{v}_{\text{ms}} \) and \( p_{\text{ms}} \). Define \( U \) to be the matrix that collects all the velocity multiscale solutions and \( M_c \) to be the matrix that provides an embedding operation from \( Q_{\text{ms}} \) to \( Q_f \). \( \mathbf{v}_{c,v} \) and \( p_{c,v} \) are obtained by solving

\[
\begin{bmatrix}
    U^T A_f U & U^T B_f^T M_c \\
    M_c^T B_f U & 0
\end{bmatrix}
\begin{bmatrix}
    \mathbf{v}_{c,v} \\
    p_{c,v}
\end{bmatrix} =
\begin{bmatrix}
    \mathbf{0} \\
    M_c^T F_f
\end{bmatrix}
\]

In the following section, we will give a detailed review of the MGMsFEM.

### 3 Mixed Generalized Multiscale Finite Element Method

In this section, we will present the Generalized Multiscale Finite Element Method in mixed formulation (MGMsFEM). As mentioned before, the construction of multiscale space is divided into three steps, which will be described in the following subsections. In particular, we first construct a set of local snapshot space. Then a reduced multiscale space is further constructed by solving local snapshot problems. The last step is to construct some residual-driven bases to enrich the multiscale space obtained in the second step. The last two steps are called offline stage I and II. The corresponding multiscale bases in the above two stages are called multiscale basis type I and multiscale basis type II, where the latter one are residual-driven bases. The whole process of our algorithm is shown in Figure 2. Throughout this section, we take a deterministic permeability field \( \kappa = \kappa(x, \xi) \), where \( \xi \) is a sample from the random space \( \Omega_r \).

**Figure 2: A training-testing algorithm.** \( \kappa_1 \) and \( \kappa_2 \), \( f_1 \) and \( f_2 \) are permeability fields and sources used in training and testing stages. Training stage contains three steps and it is performed offline. The online simulation/testing stage is conducted with different parameters \( \kappa_2 \) and \( f_2 \), which may be different with those used in the training stage.

#### 3.1 Snapshot space

In this subsection, a snapshot space will be constructed to represents all possible local solutions. First, a set of local snapshot spaces \( \{V_{\text{snap}}^i\} \) are spanned by local snapshot functions, which are solved in the fine grid. Then, the final snapshot space is defined by \( V_{\text{snap}} = \bigoplus_{E_i \in E_c} V_{\text{snap}}^i \).

Suppose \( E_i \in E_c \) and \( E_i \) can be partitioned into some fine-scale edges, i.e. \( E_i = \bigcup_{j=1}^{L_i} e_j \). Here \( L_i \) is the number of fine-scale edges on the coarse-scale edge \( E_i \). To obtain \( V_{\text{snap}}^i \), we solve the following system:

\[
\begin{align*}
\mathbf{\kappa}^{-1} \mathbf{v}_j^i + \nabla p_j^i &= 0 \quad \text{in } D_i, \\
\text{div}(\mathbf{v}_j^i) &= \beta_j^i \quad \text{in } D_i.
\end{align*}
\]
subject to the zero Neumann boundary condition \( \mathbf{v}^T \cdot \mathbf{n} = 0 \) on \( \partial D_i \), where \( \mathbf{n} \) is a fixed unit-normal vector for \( E_i \). We solve (9) separately in \( K_{i,1} \) and \( K_{i,2} \) (see the Figure 1 for illustration). Hence we use an additional boundary condition \( \mathbf{v}^T \cdot \mathbf{n} = \delta^T_j \), where \( \delta^T_j \) is the Kronecker delta. In particular, it is defined as follows.

\[
\delta^T_j = \begin{cases} 
1, & \text{on } e_j, \\
0, & \text{on } E_i \setminus e_j, \\
\end{cases} 
\]

Here, a compatible condition \( \int_{K_{i,p}} \beta^T_j \mathbf{v}^T \cdot \mathbf{n}, p = 1, 2 \). We solve the above local problem in each \( D_i \) to generate \( V^{i}_{\text{snap}} \). Finally, we form the snapshot space by using \( V_{\text{snap}} = \bigoplus_{E_i \in \mathcal{E}_c} V^{i}_{\text{snap}} \).

### 3.2 Offline stage I

In this subsection, we perform a dimension reduction on \( V_{\text{snap}} \), whose dimension is comparable to that of fine-grid space \( V_f \). The resulted small-dimensional space is referred as the multiscale space in the offline stage I. First of all, we define two symmetric and positive definite bilinear operators \( a(\cdot, \cdot) \) and \( s(\cdot, \cdot) \) on \( V^{i}_{\text{snap}} \times V^{i}_{\text{snap}} \). More specifically,

\[
a(v, u) = \int_{E_i} \kappa^{-1} (v \cdot n)(u \cdot n), \\
s(v, u) = \frac{1}{H} \left( \int_{D_i} \kappa^{-1} v \cdot u + \int_{D_i} \text{div}(v) \text{div}(u) \right),
\]

for \( v, u \in V^{i}_{\text{snap}} \), and \( n \) is the fixed unit normal vector for \( E_i \). We solve the following local spectral problem to get \( \lambda \in \mathbb{R} \) and \( v \in V^{i}_{\text{snap}} \),

\[
a(v, u) = \lambda s(v, u), \quad \forall u \in V^{i}_{\text{snap}}.
\]

After solving (11) in \( D_i \), we arrange the eigenvalues in ascending order \( \lambda^i_1 \leq \lambda^i_2 \leq \cdots \leq \lambda^i_{L_i} \).

Let \( \psi^i_j \) be the eigenfunction corresponding to \( \lambda^i_j \). To reduce the dimension, we here keep the eigenfunctions associated with \( l_i \) smallest eigenvalues and use them to span the local offline space \( V^{i}_{\text{ms}} \), i.e., \( V^{i}_{\text{ms}} = \text{span}\{\psi^i_1, \ldots, \psi^i_{l_i}\} \). Then the global offline space is \( V_{\text{ms}} = \bigoplus_{E_i \in \mathcal{E}_c} V^{i}_{\text{ms}} \).

The eigenvalue problem (11) can be recast in the following matrix formulation. Define \( A^{i}_{\text{snap}} \) and \( S^{i}_{\text{snap}} \) as follows.

\[
A^{i}_{\text{snap}}[k,j] = a(v^i_k, v^i_j), \quad S^{i}_{\text{snap}}[k,j] = s(v^i_k, v^i_j), \quad \forall v^i_k, v^i_j \in V^{i}_{\text{snap}}.
\]

Define \( \mathbf{v}_{f,v} \) to be the coefficient vector of the solution \( v \) to (11) w.r.t the snapshot basis functions in \( V^{i}_{\text{snap}} \). Then we can get \( \mathbf{v}_{f,v} \) and \( \lambda \) by solving the following matrix formulation:

\[
A^{i}_{\text{snap}} \mathbf{v}_{f,v} = \lambda S^{i}_{\text{snap}} \mathbf{v}_{f,v}.
\]

### 3.3 Offline stage II

The multiscale basis function constructed in last section only depends on local permeability field, where no source or boundary condition of the solution is included in the basis construction. To compensate this deficiency, one can construct new bases to enhance the approximation ability of multiscale space, which is the goal of this subsection. We recall that the key idea in the offline stage I is utilizing a set of spectral problems to reduce the dimension of approximation space. However, since our motivation here is to perform some enrichments effectively, we apply a different bases construction method. More specifically, we will enrich the multiscale space from offline stage I by some residual-driven bases. To this end, we first solve (12) in some oversampled regions as follows.}

Figure 1 shows an oversampled domain. For each \( i = 1, \ldots, N_{E,c} \), we solve

\[
\kappa^{-1} \mathbf{v}^{i, +} + \nabla p^{i, +} = 0 \quad \text{in } D^+_i, \\
\text{div}(\mathbf{v}^{i, +}) = \gamma^{i} \quad \text{in } D^+_i
\]

\[
\text{(13)}
\]
subject to the zero boundary condition $\mathbf{v}_j^{i,+} \cdot \mathbf{n}_i = 0$ on $\partial D_i^{+}$. Similarly, we separately solve (13) in $D_{i,1}^{+}$ and $D_{i,2}^{+}$ (Figure 1) with the additional boundary condition $\mathbf{v}_j^{i,+} \cdot \mathbf{n}_i = \delta_j^i$ on $E_i^{+}$. Here $E_i^{+}$ is an oversampled coarse-scale edge shared by $D_{i,1}^{+}$ and $D_{i,2}^{+}$. Moreover, the constant $\gamma_j^i$ is selected such that (13) is solvable. From (13), we can obtain $\mathbf{v}_i^{+}$ supported in $\Omega$.

In particular, given an error tolerance $\tau$, if we restrict $\mathbf{v}_i^{+}$ to span the oversampled local snapshot space $V_{\text{snap}}^{i,+}$. Let $\tilde{V}_{\text{snap}}^{i,+}$ be the divergence free subspace of $V_{\text{snap}}^{i,+}$. In the following part, we seek the residual-driven bases in the oversampled snapshot space.

Suppose $D \subset \Omega$ is a specific domain. Let $V_D$ be the subspace of $V_{\text{snap}}$ that contains functions supported in $D$. In particular, we have $V_D = \bigoplus_{D_i \subset D} V_{\text{snap}}^i$. We further define $\tilde{V}_D$ to be the divergence free subspace of $V_D$. Define the residual operator $R_D$ on $V_D$ by

$$R_D(\mathbf{v}) = \int_D \kappa^{-1} \mathbf{v}_{\text{ms}} \cdot \mathbf{v} - \int_D \text{div}(\mathbf{v}) p_{\text{ms}}, \quad \forall \mathbf{v} \in V_D.$$ 

If we restrict $R_D$ on $\tilde{V}_D$, we have

$$R_D(\mathbf{v}) = \int_D \kappa^{-1} \mathbf{v}_{\text{ms}} \cdot \mathbf{v}, \quad \forall \mathbf{v} \in \tilde{V}_D.$$ 

We denote the operator norm of the residual $R_D$ by $\|R_D\|$. The construction of residual-driven bases is performed by iterations. We denote the multiscale space after $n$ iterations by $V_{\text{ms},n}$. In particular, $V_{\text{ms},0}$ is the offline multiscale space. Let $k$ be the iteration level. We start at $k = 0$. Each iteration contains five steps as follows.

Step 1: Compute the multiscale solution $(\mathbf{v}_k, p_k)$ in the current multiscale space $V_{\text{ms},k} \times Q_{\text{ms}}$. In particular, we solve

$$\int_\Omega \kappa^{-1} \mathbf{v}_{\text{ms}} \cdot \mathbf{u} - \int_\Omega \text{div}(\mathbf{u}) p_{\text{ms}} = 0, \quad \forall \mathbf{u} \in V_{\text{ms},k},$$

$$\int_\Omega \text{div}(\mathbf{v}_k) q = \int_\Omega f q, \quad \forall q \in Q_{\text{ms}}.$$ 

Suppose $V_{\text{ms},k} = \{\psi_1, \ldots, \psi_{N_k}\}$, where $N_k$ is the number of multiscale bases in $V_{\text{ms},k}$. $Q_{\text{ms}} = \{p_1, \ldots, p_{e_c}\}$. We can rewrite (14) in a matrix formulation as follows,

$$\begin{bmatrix} A_{\text{ms},k} & B_{\text{ms},k}^T \\ B_{\text{ms},k} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_k \\ p_k \end{bmatrix} = \begin{bmatrix} 0 \\ F_{\text{ms},k} \end{bmatrix}. $$

Here, $A_{\text{ms},k}$ and $B_{\text{ms},k}$ are constructed using multiscale bases in $V_{\text{ms},k}$. In particular, $A_{\text{ms},k}[i,j] = A(\psi_i, \psi_j)$ and $B_{\text{ms},k}[i,j] = \langle \text{div}(\psi_i), p_j \rangle$.

Step 2: Determine whether an enrichment is needed by some particular error indicators. In particular, given an error tolerance $\tau$, if $\|R_{\Omega}\| > \tau$, we go through the following three steps.

Step 3: Select a set of local regions to perform the residual-driven enrichment. For each $E_i \in \mathcal{E}_c$, we construct a corresponding oversampled neighborhood $D_i^{+}$. See the Figure 1 for an example of an oversampled neighborhood. Let $D_1, \ldots, D_{p_k}$ be a set of local neighborhoods that form a non-overlapping partition of $\Omega$.

Step 4: Obtain the boundary condition for the residual-driven bases. For each $D_i^{+}$, we solve for $\psi_i^{+} \in \tilde{V}_{\text{snap}}^{i,+}$ such that

$$\int_{D_i^{+}} \kappa^{-1} \psi_i^{+} \cdot \mathbf{u} = R_{D_i^{+}}(\mathbf{u}), \quad \forall \mathbf{u} \in \tilde{V}_{\text{snap}}^{i,+}. $$ 

Since $\psi_i^{+}$ is supported in $D_i^{+}$, we need to restrict $\psi_i^{+} \cdot \mathbf{n}_i^{+}$ on the coarse-scale edge $E_i$ and normalize it to obtain $z_i$. 

7
Step 5: For each $D_i$, compute a residual-driven basis $\phi_i$ supported in $D_i$ with computed boundary value $z_i$ by solving

\begin{align}
\kappa^{-1} \phi_i + \nabla p_i &= 0 \text{ in } D_i, \\
\text{div}(\phi_i) &= \beta_i \text{ in } D_i, \\
\phi_i \cdot \mathbf{n}_i &= z_i \text{ on } E_i, \\
\phi_i \cdot \mathbf{n}_i &= 0 \text{ on } \partial D_i,
\end{align}

(16)

where $\beta_i$ is chosen to satisfy $\int_K \beta_i = \int_{\partial K} \phi_i \cdot \mathbf{n}_i$ for each $K \subset D_i$ and $n_i$ is a fixed unit-normal vector for the coarse face $E_i$. Note that the process of solving (16) is unnecessary when $V_{\text{snap}}$ is already obtained. Define $\phi_1, \ldots, \phi_{p_k}$ to be residual-driven bases obtained in iteration $k$. We then update the velocity space by letting $V_{ms}^{k+1} = V_{ms}^k \cup \text{span}\{\phi_1, \ldots, \phi_{p_k}\}$.

Using the above algorithm, we can terminate the residual-driven enrichment if $\|R\| \leq \tau$.

In Figure 3, local profile in a specific coarse neighborhood (see Figure 1) is shown. Choosing a local region $D_i$, we plot the permeability field, a multiscale basis in the offline stage I, and a residual-driven basis corresponding to $D_i$. Moreover, a sample snapshot generated during the construction of multiscale bases is also displayed. From (a), we can see there are some high-contrast variations in the local permeability field. The highest value is over 3000 while the smallest value is $10^{-4}$. The inverse of smallest eigenvalues are shown in log$_{10}$ scale in (b). The decay is rapid especially among the 5 smallest eigenvalues. In (c) and (d), a sample snapshot is displayed, where $x$ and $y$ components are shown in two columns. A sample multiscale basis $\psi$ in the offline stage I and a sample residual-driven multiscale basis $\phi$ are displayed in (e)-(f) and (g)-(h) respectively. The multiscale bases show some distinct variations at some specific places due to the heterogeneity of the permeability field. Besides, the residual-driven basis function has fewer fluctuations compared with the type I basis function $\psi$ but the oscillations in $\psi$ are more centralized, where the distinction is a consequence of different constructions. In particular, the residual-driven basis function is computed based on some residuals while the basis type I is a linear combination of some eigenfunctions that are more divergent. We also remark that although the residual-driven basis is more efficient but it relies on the residual of a specific problem, so if the online simulation problem has a sharp difference with the training problem, residual-driven bases may not work well, and we will show this point in our analysis part.

4 Analysis

Suppose $\Omega$ is a bounded domain in $R^d (d = 2, 3)$. $\Omega_\tau$ is a sample space. We aim at solving the proposed problem:

\begin{align}
\kappa^{-1}(x; \omega)\mathbf{u} + \nabla p &= 0 \text{ in } \Omega, \\
\text{div}(\mathbf{u}) &= f \text{ in } \Omega.
\end{align}

(17) \quad (18)

First of all, we introduce some notations.

1. $\mathbf{u}_f(x; \omega, f)$: a fine-grid velocity solution solved in $V_f$. $p_f(x; \omega, f)$: a fine-grid pressure solution solved in $Q_f$.

2. $\omega_1$ and $f_1$ are a training permeability field and a training source used to generate a multiscale space.

3. $V_{ms,k}$: a velocity multiscale space at enrichment iteration $k$. $V_{ms} := V_{ms,m}$: a final velocity multiscale space is generated after $m$ enrichment iterations.

4. $\mathbf{u}_{ms}(x; \omega, f)$: a velocity multiscale solution solved in $V_{ms}$. $p_{ms}(x; \omega, f)$: a pressure multiscale solution solved in $Q_{ms}$. $\mathbf{u}_{ms}^{1,1,(k)}(x; \omega_1, f_1)$: a velocity multiscale solution corresponding to $\omega_1$ and $f_1$ solved in $V_{ms,k}$.
Figure 3: Sample local permeability field, inverse of sample eigenvalues, a sample snapshot, an offline basis $\psi$, and a residual-driven basis $\phi$ in a specific local neighborhood. (a): Sample permeability field in natural logarithm. (b): Inverse of 20 smallest eigenvalues shown in $\log_{10}$ scale. (c)-(d): $x$ and $y$ components of a sample local snapshot. (e)-(f): $x$ and $y$ components of a sample offline multiscale basis $\psi$ in offline stage I. (g)-(h): $x$ and $y$ components of a sample residual-driven basis $\phi$ in the offline stage II.
5. \( \kappa_{1, \text{max}} := \max_{x \in \Omega} \kappa(x, \omega_1), \kappa_{1, \text{min}} := \min_{x \in \Omega} \kappa(x, \omega_1), \kappa_{1, \text{contra}} := \frac{\kappa_{1, \text{max}}}{\kappa_{1, \text{min}}} \).

6. \( \Omega_1, \ldots, \Omega_P \) are a set of non-overlapping regions such that \( \Omega = \bigcup_{j=1}^P \Omega_j \).

In particular, \( u_f(x; \omega, f) \) and \( p_f(x; \omega, f) \) are velocity and pressure solutions to (19) and (20); \( u_{ms}(x; \omega, f) \) and \( p_{ms}(x; \omega, f) \) are velocity and pressure solutions to (21) and (22) as follows.

\[
\int_{\Omega} \kappa^{-1}(x; \omega) u_f(x; \omega, f) \cdot v - \int_{\Omega} \text{div}(v)p_f(x; \omega, f) = 0, \quad \forall v \in V_f^0, \quad (19)
\]
\[
\int_{\Omega} \text{div}(u_f(x; \omega, f))q = \int_{\Omega} f q, \quad \forall q \in Q_f. \quad (20)
\]
\[
\int_{\Omega} \kappa^{-1}(x; \omega) u_{ms}(x; \omega, f) \cdot v - \int_{\Omega} \text{div}(v)p_{ms}(x; \omega, f) = 0, \quad \forall v \in V_{ms}, \quad (21)
\]
\[
\int_{\Omega} \text{div}(u_{ms}(x; \omega, f))q = \int_{\Omega} f q, \quad \forall q \in Q_{ms}. \quad (22)
\]

In the following part, we give an error estimate of a multiscale solution in an online simulation stage solved in a fixed multiscale space \( V_{ms} \). In particular, we arbitrarily choose a sample permeability field and a source term. Once they are chosen, they are deterministic, which are denoted by \( \kappa_1 \) and \( f_1 \) respectively. We recall that \( \kappa_1 \) and \( f_1 \) are the training permeability field and source used to generate \( V_{ms} \). To simplify the above mentioned notations, we use some abbreviations.

For \( i, j \in \{1, 2\} \), we let \( u_f^{i,j} := u_f(x; \omega_i, f_j) \) and \( p_f^{i,j} := p_f(x; \omega_i, f_j) \); \( u_{ms}^{i,j} := u_{ms}(x; \omega_i, f_j) \) and \( p_{ms}^{i,j} := p_{ms}(x; \omega_i, f_j) \); \( u_{ms}^{1,1(m)} := u_{ms}^{(m)}(x; \omega_1, f_1) \); \( \kappa_i := \kappa(x, \omega_i) \).

Define \( e = \kappa_2^{-\frac{1}{2}}(u_f^{2,2} - u_{ms}^{2,2}) \). To estimate \( e \), we split it into five parts as follows:

\[
e = \kappa_2^{-\frac{1}{2}}u_f^{2,2} - \kappa_1^{-\frac{1}{2}}u_f^{1,2} + \kappa_1^{-\frac{1}{2}}(u_f^{1,2} - u_f^{1,1}) + u_{ms}^{1,1} - u_{ms}^{1,1} + u_{ms}^{1,2} - u_{ms}^{1,2} + \kappa_1^{-\frac{1}{2}}u_{ms}^{1,2} - \kappa_2^{-\frac{1}{2}}u_{ms}^{2,2} \quad (23)
\]

\[
= e_1 + e_2 + e_3 + e_4 + e_5.
\]

In particular,

\[
e_1 = \kappa_2^{-\frac{1}{2}}u_f^{2,2} - \kappa_1^{-\frac{1}{2}}u_f^{1,2},
\]
\[
e_2 = \kappa_1^{-\frac{1}{2}}(u_f^{1,2} - u_f^{1,1}),
\]
\[
e_3 = \kappa_1^{-\frac{1}{2}}(u_f^{1,1} - u_{ms}^{1,1}),
\]
\[
e_4 = \kappa_1^{-\frac{1}{2}}(u_{ms}^{1,1} - u_{ms}^{1,2}),
\]
\[
e_5 = \kappa_1^{-\frac{1}{2}}u_{ms}^{1,2} - \kappa_2^{-\frac{1}{2}}u_{ms}^{2,2}.
\]

We remark that our proposed method is to choose a well-chosen permeability field and a fixed source term, based on which a multiscale space is constructed in an offline stage. It is worth noting that the offline stage contains the process of computing residual-driven bases. Furthermore, we can use the pre-computed multiscale space to solve the problem with a different permeability field and a source term in a testing stage. Hence, the final error may come from three aspects: multiscale approximation, different permeability fields, and different source terms. More specifically, in (23), \( e_3 \) refers to the approximation error of a multiscale space, i.e. we measure the difference of a multiscale solution with a corresponding reference solution, where the permeability field and the source term in the concerned equation are the same with those used in constructing multiscale space. \( e_1 \) and \( e_2 \) are associated with differences of reference solutions to different permeability fields and sources, respectively. \( e_4 \) and \( e_5 \) refer to differences of two multiscale solutions. Besides, the way to derive the
bounds for $e_1$ and $e_5$ is almost the same. Hence, it suffices to derive the error bounds for $e_1$. At the same time, the bounds for $e_2$ and $e_4$ are obtained similarly. We will estimate each part individually and summarize them in the end. We will start with $e_1$ and $e_5$. First of all, we prove the stability of the fine-scale and multiscale velocity solutions.

**Lemma 1.** Suppose $\mathbf{u}_f := \mathbf{u}_f(x; \omega, f)$ and $\mathbf{u}_{ms} := \mathbf{u}_{ms}(x; \omega, f)$ are fine-grid and multiscale velocity solutions to (19)–(20) and (21)–(22), respectively. Then there exists $C$ (independent of $\kappa$) such that the following two estimations hold:

$$
\| \mathbf{u}_f \|_{L^2} \leq C \| f \|_{L^2},
\| \mathbf{u}_{ms} \|_{L^2} \leq C \| f \|_{L^2}.
$$

**Proof.** It suffices to prove (24) since one may apply same strategy to prove (25). Take $q = \text{div}(\mathbf{u}_f)$ in (20) and one have

$$
\| \text{div}(\mathbf{u}_f) \|_{L^2} \leq \int_{\Omega} f \text{div}(\mathbf{u}_f) \leq \| f \|_{L^2} \| \text{div}(\mathbf{u}_f) \|_{L^2},
$$

where we apply the Cauchy-Schwartz inequality. Using the Poincaré inequality and $\mathbf{u}_f \in V_f$, one can obtain

$$
\| \mathbf{u}_f \|_{L^2} \leq C \| \text{div}(\mathbf{u}_f) \|_{L^2}.
$$

Hence (24) is proved. Similarly, we can prove (25). □

We now give estimates $e_1$ and $e_5$.

**Theorem 2.** Recall that $\mathbf{u}_f^{i,2} := \mathbf{u}_f(x; \omega_i, f_2)$ and $\mathbf{u}_{ms}^{i,2} := \mathbf{u}_{ms}(x; \omega_i, f_2)$ are fine-grid and multiscale velocity solutions to (19)–(20) and (21)–(22) with $\omega_i$ and $f_2$, for $i = 1, 2$. Moreover, $p_f^{i,2} := p_f(x; \omega_i, f_2)$ and $p_{ms}^{i,2} := p_{ms}(x; \omega_i, f_2)$ are corresponding fine-grid and multiscale pressure solutions with $\omega_i$ and $f_2$. Then the following estimations hold,

$$
\| \kappa_2^{-\frac{1}{2}} \mathbf{u}_f^{i,2} - \kappa_1^{-\frac{1}{2}} \mathbf{u}_{ms}^{i,2} \|_{L^2} \leq C \| \kappa_2^{-\frac{1}{2}} - \kappa_1^{-\frac{1}{2}} \| \| f_2 \|_{L^2},
\| \kappa_2^{-\frac{1}{2}} \mathbf{u}_{ms}^{i,2} - \kappa_1^{-\frac{1}{2}} \mathbf{u}_f^{i,2} \|_{L^2} \leq C \| \kappa_2^{-\frac{1}{2}} - \kappa_1^{-\frac{1}{2}} \| \| f_2 \|_{L^2}.
$$

**Proof.** Recalling (19) and (20) associated with $\kappa_2$ and $f_2$, we obtain

$$
\int_{\Omega} \kappa_2^{-1} \mathbf{u}_f^{2,2} \cdot \mathbf{v} - \int_{\Omega} \text{div}(\mathbf{v}) p_f^{2,2} = 0, \quad \forall \mathbf{v} \in V_f^0,
\int_{\Omega} \text{div}(\mathbf{u}_f^{2,2}) q = \int_{\Omega} f_2 q, \quad \forall q \in Q_f.
$$

Similarly, with $\kappa_1$ and $f_2$, one can attain

$$
\int_{\Omega} \kappa_1^{-1} \mathbf{u}_f^{1,2} \cdot \mathbf{v} - \int_{\Omega} \text{div}(\mathbf{v}) p_f^{1,2} = 0, \quad \forall \mathbf{v} \in V_f^0,
\int_{\Omega} \text{div}(\mathbf{u}_f^{1,2}) q = \int_{\Omega} f_2 q, \quad \forall q \in Q_f.
$$

Taking $\mathbf{v} = \mathbf{u}_f^{2,2}$ in (28) and $q = p_f^{2,2}$ in (29), it holds that

$$
\int_{\Omega} \kappa_2^{-1} |\mathbf{u}_f^{2,2}|^2 - \int_{\Omega} \text{div}(\mathbf{u}_f^{2,2}) p_f^{2,2} = 0,
\int_{\Omega} \text{div}(\mathbf{u}_f^{2,2}) p_f^{2,2} = \int_{\Omega} f_2 p_f^{2,2}.
$$

11
It follows that
\[ \int_\Omega \kappa_2^{-1} |\mathbf{u}^{2,2}_f|^2 = \int_\Omega f_2 p_f^{2,2}. \]  
(34)

Taking \( \mathbf{v} = \mathbf{u}^{1,2}_f \) in (28) and \( q = p_f^{2,2} \) in (31), it holds that
\[ \int_\Omega \kappa_2^{-1} \mathbf{u}^{2,2}_f \cdot \mathbf{u}^{1,2}_f = \int_\Omega f_2 p_f^{2,2}. \]  
(35)

Then we have
\[ \int_\Omega \kappa_2^{-1} \mathbf{u}^{2,2}_f \cdot \mathbf{u}^{1,2}_f = \int_\Omega \kappa_2^{-1} |\mathbf{u}^{2,2}_f|^2. \]  
(36)

Similarly, one can derive that
\[ \int_\Omega \kappa_1^{-1} \mathbf{u}^{1,2}_f \cdot \mathbf{u}^{2,2}_f = \int_\Omega \kappa_1^{-1} |\mathbf{u}^{1,2}_f|^2. \]  
(37)

Based on (36) and (37), we have
\[ \int_\Omega |\kappa_1^{-\frac{1}{2}} \mathbf{u}^{1,2}_f - \kappa_2^{-\frac{1}{2}} \mathbf{u}^{2,2}_f|^2 = \int_\Omega |\kappa_1^{-\frac{1}{2}} \mathbf{u}^{1,2}_f|^2 + |\kappa_2^{-\frac{1}{2}} \mathbf{u}^{2,2}_f|^2 - 2(|\kappa_1 \kappa_2|)^{-\frac{1}{2}} \mathbf{u}^{1,2}_f \cdot \mathbf{u}^{2,2}_f 
\geq \int_\Omega |\kappa_1^{-\frac{1}{2}} - \kappa_2^{-\frac{1}{2}}|^2 |\mathbf{u}^{1,2}_f|^2 \cdot |\mathbf{u}^{2,2}_f|. \]

We conclude that
\[ \|\kappa_1^{-\frac{1}{2}} \mathbf{u}^{1,2}_f - \kappa_2^{-\frac{1}{2}} \mathbf{u}^{2,2}_f\|_{L^2}^2 \leq \|\kappa_1^{-\frac{1}{2}} - \kappa_2^{-\frac{1}{2}}\|_\infty \|\mathbf{u}^{1,2}_f\|_{L^2} \|\mathbf{u}^{2,2}_f\|_{L^2}, \]
\[ \leq C \|\kappa_1^{-\frac{1}{2}} - \kappa_2^{-\frac{1}{2}}\|_\infty \|f_1 - f_2\|_{L^2}, \]
where the last inequality holds based on Lemma 1. Similarly, (27) can be obtained.

To estimate \( e_2 \), we define a weighted \( L^2 \) norm by \( \|v\|_{\kappa^{-1},\Omega} := \int_\Omega \kappa^{-1}|v|^2 \).

**Theorem 3.** Recall that for \( i = 1, 2, \mathbf{u}^{1,i}_f := \mathbf{u}_f(x; \omega_1, f_i) \) and \( \mathbf{u}^{1,i}_{ms} := \mathbf{u}_{ms}(x; \omega_1, f_i) \) are fine-grid and multiscale velocity solutions to (19)-(20) and (21)-(22) with \( \omega_1 \) and \( f_i \); \( p^{1,i}_f := p_f(x; \omega_1, f_i) \) and \( p^{1,i}_{ms} := p_{ms}(x; \omega_1, f_i) \) are corresponding fine-grid and multiscale pressure solutions with \( \omega_1 \) and \( f_i \). Then, it holds that,
\[ \|\mathbf{u}^{1,1}_f - \mathbf{u}^{1,2}_f\|_{\kappa^{-1},\Omega} \lesssim \kappa_{1,\min}^{-\frac{1}{2}} \|f_1 - f_2\|_{L^2}, \]  
(38)
\[ \|\mathbf{u}^{1,1}_{ms} - \mathbf{u}^{1,2}_{ms}\|_{\kappa^{-1},\Omega} \lesssim C_{\text{inf sup}} \kappa_{1,\min}^{-\frac{1}{2}} \|f_1 - f_2\|_{L^2}, \]  
(39)
where \( \kappa_{1,\min} := \min_{x \in \Omega} \kappa(x; \omega_1) \).

**Proof.** Recall that \( \kappa_1 := \kappa(x; \omega_1) \). Similarly as the proof in Theorem 2, we subtract (19) and (20) corresponding with \( f_1 \) and \( f_2 \) to get
\[ \int_\Omega \kappa_1^{-1}(\mathbf{u}^{1,1}_f - \mathbf{u}^{1,2}_f) \cdot \mathbf{v} - \int_\Omega \text{div}(\mathbf{v})(p^{1,1}_f - p^{1,2}_f) = 0, \quad \forall \mathbf{v} \in V_f^0, \]  
(40)
\[ \int_\Omega \text{div}(\mathbf{u}^{1,1}_f - \mathbf{u}^{1,2}_f) q = \int_\Omega (f_1 - f_2) q, \quad \forall q \in Q_f. \]  
(41)
Take \( v = u_{f}^{1,1} - u_{f}^{1,2}, q = p_{f}^{1,1} - p_{f}^{1,2} \) and it holds that
\[
\|u_{f}^{1,1} - u_{f}^{1,2}\|_{\kappa_{1}^{-1}, \Omega} = \int_{\Omega} (f_{1} - f_{2})(p_{f}^{1,1} - p_{f}^{1,2}).
\]

By Cauchy-Schwartz inequality,
\[
\|u_{f}^{1,1} - u_{f}^{1,2}\|_{\kappa_{1}^{-1}, \Omega} \leq \|p_{f}^{1,1} - p_{f}^{1,2}\|_{L^{2}} \|f_{1} - f_{2}\|_{L^{2}}.
\] (42)

The Raviart-Thomas elements satisfy the following inf-sup condition [7]:
\[
\|q_{h}\|_{L^{2}} \lesssim \sup_{v_{h} \in V_{f}} \frac{\int_{\Omega} \text{div}(v_{h})q_{h}}{\|v_{h}\|_{H(\text{div}; \Omega)}}, \quad \forall q_{h} \in Q_{h}.
\] (43)

Using (40), we obtain
\[
\|p_{f}^{1,1} - p_{f}^{1,2}\|_{L^{2}} \lesssim \kappa_{1, \min}^{-\frac{1}{2}} \|u_{f}^{1,1} - u_{f}^{1,2}\|_{\kappa_{1}^{-1}, \Omega}.
\] (44)

By (42) and (44), we conclude that
\[
\|u_{f}^{1,1} - u_{f}^{1,2}\|_{\kappa_{1}^{-1}, \Omega} \lesssim \kappa_{1, \min}^{-\frac{1}{2}} \|f_{1} - f_{2}\|_{L^{2}}.
\] (45)

The process to prove (39) is similar. We subtract (21) and (22) associated with \( f_{1} \) and \( f_{2} \) and it holds that
\[
\int_{\Omega} \kappa_{1}^{-1}(u_{ms}^{1,1} - u_{ms}^{1,2}) \cdot \nu - \int_{\Omega} \text{div}(\nu)(p_{ms}^{1,1} - p_{ms}^{1,2}) = 0, \quad \forall \nu \in V_{ms},
\] (46)
\[
\int_{\Omega} \text{div}(u_{ms}^{1,1} - u_{ms}^{1,2})q = \int_{\Omega} (f_{1} - f_{2})q, \quad \forall q \in Q_{ms}.
\] (47)

Taking \( \nu = u_{ms}^{1,1} - u_{ms}^{1,2} \) and \( q = p_{ms}^{1,1} - p_{ms}^{1,2} \), it follows that
\[
\|u_{ms}^{1,1} - u_{ms}^{1,2}\|_{\kappa_{1}^{-1}, \Omega} \leq \|p_{ms}^{1,1} - p_{ms}^{1,2}\|_{L^{2}} \|f_{1} - f_{2}\|_{L^{2}}.
\] (48)

From Theorem 4.2 in [20], we have
\[
\|p\|_{L^{2}} \lesssim C_{\text{inf-sup}} \sup_{\nu \in V_{ms, 0}} \frac{\int_{\Omega} \text{div}(\nu)p}{\|v_{h}\|_{H(\text{div}; \Omega_{1}^{-1})}}, \quad \forall p \in Q_{ms},
\] (49)

where \( V_{ms, 0} \) is the offline multiscale space.

\[
C_{\text{inf-sup}} = (1 + \max_{1 \leq i \leq N_{k, e}} \min_{r} \int_{D_{i}} \kappa_{1}^{-1} |\psi_{r}^{i}|^{2})^{\frac{1}{2}},
\] (50)

where the minimum is taken over all indices \( r \) with the property \( \int_{E_{i}} \psi_{r}^{i} \cdot m_{i} \neq 0 \). Here \( m_{i} \) is an outer normal vector to \( E_{i} \). Since \( V_{ms} \) is an enriched multiscale space from \( V_{ms, 0}, V_{ms, 0} \subset V_{ms} \). Then the following holds,
\[
\|p\|_{L^{2}} \lesssim C_{\text{inf-sup}} \sup_{\nu \in V_{ms}} \frac{\int_{\Omega} \text{div}(\nu)p}{\|v_{h}\|_{H(\text{div}; \Omega_{1}^{-1})}}, \quad \forall p \in Q_{ms},
\] (51)

Using (46), we have
\[
\|p_{ms}^{1,1} - p_{ms}^{1,2}\|_{L^{2}} \lesssim \kappa_{1, \min}^{-\frac{1}{2}} \|u_{ms}^{1,1} - u_{ms}^{1,2}\|_{\kappa_{1}^{-1}, \Omega}.
\] (52)

Combining (48)-(52), (39) is attained.
In the following part, we estimate $e_3$. Given a region $D$, we recall that $V_D := \bigoplus_{D \subset \Omega} V_{\text{snap}}^1$, $\tilde{V}_D$ is defined to be the divergence free subspace of $V_D$. Then we define the residual operator $R^{(m)}_D$ as a linear functional on $V_D$ on the enrichment level $m$ by

$$R^{(m)}_D(v) = \int_D \kappa^{-1}_m u^{(m)} \cdot v - \int_D \text{div}(v)p^{(m)}_m, \quad v \in V_D.$$ 

If we restrict $R^{(m)}_D$ on $\tilde{V}_D$, it holds that

$$R^{(m)}_D(v) = \int_D \kappa^{-1}_m u^{(m)} \cdot v.$$ 

Moreover, we define the norm of $R^{(m)}_D$ by

$$\|R^{(m)}_D\|_{(V_D)^*} = \sup_{v \in V_D} \frac{|R^{(m)}_D(v)|}{\|v\|_{H(\text{div};D;\kappa^{-1}_m)}}.$$ 

**Theorem 4.** [41] Recall that $u^{1,1}_f := u_f(x;\omega_1, f_1)$, $u^{1,1}_{ms} := u_{ms}(x;\omega_1, f_1)$ are fine-grid and multiscale velocity solutions to (19) and (20) with $\omega_1$ and $f_1$. Suppose $u^{1,1}_{ms}$ is the multiscale solution at the enrichment level $m$, i.e., $u^{1,1}_{ms} = u^{1,1}_{ms}(m)$, which is the solution to (14) with $\omega_1$ and $f_1$. Then we have the following estimation:

$$\|u^{1,1}_f - u^{1,1}_{ms}\|_{\kappa^{-1}_m, \Omega} \leq C_{\text{err}} \sum_{j=1}^{N_{e}} \|R^{(0)}_{D_j}\|_{(V_{\text{snap}}^1)}(\lambda^{(j)}_{r+1})^{-1} - \sum_{i=1}^{m} \sum_{j=1}^{P} \|R^{(i)}_{D_j}\|_{(V_{D_j})^*},$$

where $C_{\text{err}} = C_{\text{h}} \frac{H}{h}$ and $C_{\text{V}}$ depends on the polynomial order of the fine-grid basis functions in $V_{\text{snap}}$.

**Theorem 5.** For each $\omega_2$ and $f_2$, we recall that $u^{2,2}_f := u_f(x;\omega_2, f_2)$, $u^{2,2}_{ms} := u_{ms}(x;\omega_2, f_2)$ are fine-grid and multiscale velocity solutions to (19)-(20) and (21)-(22) with $\omega_2$ and $f_2$. Combining the previous conclusions, the following estimation holds,

$$\|u^{2,2}_f - u^{2,2}_{ms}\|_{\kappa^{-1}_m, \Omega} \leq C\|\kappa_2^{-\frac{1}{2}} - \kappa_1^{-\frac{1}{2}}\|_{L^\infty} \|f_2\|_{L^2} + (1 + C_{\text{inf,sup}})\kappa_{1,\text{min}}^{-\frac{1}{2}}\|f_1 - f_2\|_{L^2} +$$

$$C_{\text{err}} \sum_{j=1}^{N_{e}} \|R^{(0)}_{D_j}\|_{(V_{\text{snap}}^1)}(\lambda^{(j)}_{r+1})^{-1} - \sum_{i=1}^{m} \sum_{j=1}^{P} \|R^{(i)}_{D_j}\|_{(V_{D_j})^*}. \quad (53)$$

Here we recall that $C_{\text{inf,sup}}$ is defined in [50].

We remark that the this theorem states that the total error is a consequence of three parts, i.e. different sources $f_1$ and $f_2$, different permeability fields $\kappa_1$ and $\kappa_2$, and multiscale approximation error, respectively. As for the last part, it can be interpreted as a truncated error since only a few eigenvectors of the local spectral problems are included in the offline stage I. However, since there is a second stage, the offline stage II, we can compensate the previous approximation error by some residual-driven bases, which is characterized by subtracting the summation of some residuals.

5 Numerical results

In this part, we present numerical results. In the following tests, we assume that the logarithmic permeability field $Y(x, \omega) = \log(\kappa(x, \omega))$ is a second-order stationary Gaussian random field. To perform the KL expansion of the random field, we need to define a covariance function as follows:

$$C(x, z) = \sigma^2 \exp \left( -\frac{|x_1 - z_1|^2}{2\eta_1^2} - \frac{|x_2 - z_2|^2}{2\eta_2^2} \right), \quad (54)$$

where $\sigma$ and $\eta_1, \eta_2$ are the standard deviation and the correlation length in each direction, respectively.
where \( \mathbf{x} = (x_1, x_2) \) and \( \mathbf{z} = (z_1, z_2) \) are two arbitrary variables in the spatial domain \( \Omega \). Moreover, \( \eta_1 \) and \( \eta_2 \) are correlation lengths in first and second directions. Suppose \( (\lambda_i, f_i), i = 1, \ldots, +\infty \), are eigenvalues and corresponding eigenfunctions of \( C(\mathbf{x}, \mathbf{z}) \). Without loss of generality, \( \{\lambda_i\} \) are arranged in a nonincreasing order. One can approximate \( Y(\mathbf{x}, \omega) \) by

\[
Y(\mathbf{x}, \omega) \approx E[Y(\mathbf{x}, \omega)] + \sum_{i=1}^{N_k} \mu_i \sqrt{\lambda_i} f_i,
\]

(55)

where \( \{\mu_i\}_{i=1}^{N_k} \) are standard identically independent Gaussian random variables.

We consider five deterministic fields to serve as \( E[Y(\mathbf{x}, \omega)] \), denoted by \( \kappa_i \), \( i = 1, \ldots, 5 \). Moreover, we set \( \eta_1 = \eta_2 = \eta \). We both apply two-dimensional and three-dimensional models. In Table I, we show the number of truncated KL expansion terms \( N_k \) used in five models with two different choices of correlation lengths \( \eta \). More specifically, we use \( \eta = \frac{1}{8}, \frac{1}{16} \) in three 2D permeability fields and for 3D case, \( \eta = \frac{1}{4}, \frac{1}{8} \). Here, we emphasize that due to a fast decay of eigenvalues of \( C(\mathbf{x}, \mathbf{z}) \), the summation of a small number of largest eigenvalues can account for almost all the energy. Thus, the eigenvectors corresponding to the dominating eigenvalues can be used to represent the Gaussian field without losing much important information. Thanks to this design, the computational effort contained in parameterizing the stochastic permeability field is greatly reduced, which is another contribution of our proposed reduced-order method.

\( \Omega \) is set to be \([0, 2.2] \times [0, 0.6] \) for \( \kappa_1 \) and \( \kappa_2 \). For \( \kappa_3, \kappa_4 = [0, 1.28] \times [0, 1.28] \). As for the three-dimensional case, \( \Omega = [0, 2.2] \times [0, 0.6] \times [0, 0.3] \) for \( \kappa_2 \) and \([0, 0.64] \times [0, 0.64] \) for \( \kappa_5 \). \( \kappa_1, \kappa_2 \) and \( \kappa_4 \) are extracted from SPE10 model, which serves as a benchmark in estimating the effect of upscaling and multiscale (ms for short) approaches. The SPE model is characterized by a high contrast and strong heterogeneity. \( \kappa_1 \) is the 36-th layer and \( \kappa_2 \) are the 85-th layer. \( \kappa_4 \) is composed of the last 30 layers of SPE model. The \( \kappa_3 \) and \( \kappa_5 \) are two high-contrast models, where the contrast is as big as \( 10^4 \). In Table II, we calculate the mesh sizes of fine-scale and coarse-scale mesh applied to five permeability fields. Before the discussion, we define \( v_f \) and \( v_{ms} \) to be the fine-grid solution and the GMsFEM solution respectively. Furthermore, the relative velocity error \( e_v \) is defined as follows,

\[
e_v := \frac{\int_\Omega \kappa^{-1} |v_f - v_{ms}|^2}{\int_\Omega \kappa^{-1} |v_f|^2}.
\]

We define \( f_1 \) and \( f_2 \) for 2D sources as follows.

\[
f_1 = \begin{cases}
1, & x \in \zeta_i, \\
-1, & x \in \zeta_4, \\
0, & \text{otherwise},
\end{cases}
\]

\[
f_2 = \begin{cases}
1, & x \in \zeta_i, i = 1, \ldots, 4 \\
-4, & x \in \zeta_5, \\
0, & \text{otherwise},
\end{cases}
\]

(56)

where \( \zeta_i, i = 1, \ldots, 4 \) are fine-scale elements at the four corners of \( \Omega \). In particular, \( \zeta_1 = [0, h] \times [0, h] \), \( \zeta_2 = [L_x - h, L_x] \times [0, h] \), \( \zeta_3 = [0, h] \times [L_y - h, L_y] \), \( \zeta_4 = [L_x - h, L_x] \times [L_y - h, L_y] \). Moreover \( \zeta_5 \) is the fine-scale element at the center of \( \Omega \). More specifically, \( \zeta_5 = \left[\frac{L_x - h}{2}, \frac{L_x + h}{2}\right] \times \left[\frac{L_y - h}{2}, \frac{L_y + h}{2}\right] \). We call \( f_1 \) a two-point source while \( f_2 \) is called five-point source. In 3D case, we can similarly define a two-point source and a five-point source as follows.

\[
f_3 = \begin{cases}
1, & x \in \iota_i, \\
-1, & x \in \iota_4, \\
0, & \text{otherwise},
\end{cases}
\]

\[
f_4 = \begin{cases}
1, & x \in \iota_i, i = 1, \ldots, 4 \\
-4, & x \in \iota_5, \\
0, & \text{otherwise},
\end{cases}
\]

(57)

where \( \iota_i \) is the cube with bottom side \( \zeta_i \), for \( i = 1, \ldots, 5 \). In particular, \( \iota_i = \zeta_i \times [0, L_z] \). Unless specified, sources are chosen to be \( f_1 \) in 2D case and \( f_3 \) in 3D case.

Before the presentation of numerical results, we first show a comparison of numbers of unknowns in solving a fine-scale flow equation and using multiscale bases to solve a flow equation. Without
Figure 4: Permeability fields
loss of generality, we can use a 2D case for illustration. Define $N_{\text{fine},x} := L_x/h$ and $N_{\text{fine},y} := L_y/h$ to be the numbers of fine-scale elements in the horizontal and vertical directions. Correspondingly, define $N_{\text{ms},x} := L_x/H$ and $N_{\text{ms},y} := L_y/H$ to be the numbers of coarse-scale elements in two directions. Since the same number of multiscale bases are used in each local neighborhood, we let $N_b$ to be the dimension of each local multiscale space. Besides, we define “A + B” to denote that A and B bases are incorporated in each local neighborhood in offline stage I and II, respectively. We then compare the DOF in fine-scale and coarse-scale equations. The Raviart-Thomas mixed finite element discretization of the flow equation gives rise to a linear system with $N_{\text{fine},x} \times N_{\text{fine},y} + (N_{\text{fine},x} - 1) \times N_{\text{fine},y} + N_{\text{fine},x} \times (N_{\text{fine},y} - 1)$ degrees of freedom. In contrast, with multiscale bases, one can reduce the original flow equation to a system with $N_{\text{ms},x} \times N_{\text{ms},y} + N_{b} \times [(N_{\text{ms},x} - 1) \times N_{\text{ms},y} + N_{\text{ms},x} \times (N_{\text{ms},y} - 1)]$ degrees of freedom. In one example of our simulation, we choose $N_{\text{fine},x} = 220$ and $N_{\text{fine},y} = 60$, while $N_{\text{ms},x} = 11$ and $N_{\text{ms},y} = 3$. Besides, we choose $N_b = 3$, which is sufficient to give a relatively good approximation in our cases. Under the above settings, the fine-scale equation system has 39320 unknowns while the multiscale solver can result in a system with 189 unknowns. From this example, we can see a great reduction is achieved by using multiscale bases.

In 3D case we observe a more significant reduction. In Table 4 and 5, we show Dof, training time as well as testing time for SPE model ($\kappa_4$) and high-contrast model ($\kappa_5$), respectively. In particular, the training time refers to the time spent in constructing multiscale bases while the testing time counts the time cost in solving a lower-dimensional system once based on the prepared multiscale bases. We first present the DOF and solving time of full-order system. For SPE model, the number of unknowns is 1562400 and a single solve needs 58.3s. As for the high-contrast model, the fine-grid system has 1036288 unknowns and it requires 31.86s per single solve. So it is time-consuming to solve such big systems. On the contrary, the multiscale systems are relatively smaller. Since $\kappa_4$ has about 90s with $\kappa_5$ solve only needs about 2.92s. Even though the training process will increase computation time, this preparation effort is obviously cost-effective.

We then compare the DOF in fine-scale and coarse-scale equations. The Raviart-Thomas mixed bases give rise to a system with 3312 unknowns and the corresponding solving time is 2.41s per solve, which is remarkably shortened. For $\kappa_4$, “2+1” bases give rise to a system with 3312 unknowns and the corresponding solving time is 2.41s per solve, which is remarkably shortened. For $\kappa_5$, the system with 2 + 1 bases has DOF 4544 and each solve only needs about 2.92s. Even though the training process will increase computation time, about 90s with $\kappa_4$, the training process is only needed once. When there are numerous samples, this preparation effort is obviously cost-effective.

### 5.1 Single-phase flow

In this subsection, we consider a single-phase flow modeled by equations (1) and (2). In Table 2, we show the velocity errors corresponding to $\kappa_1$, $\kappa_2$ and $\kappa_3$ with different numbers of local bases when the correlation length $\eta = \frac{1}{2}$. More specifically, we both consider offline cases (from “3 + 0” to “20 + 0”) and residual-driven cases (from “1 + 1” to “4 + 2”). Here, we use the “20 + 0” error to serve as a lower bound of multiscale approximation biases with $\kappa_1$ and $\kappa_2$ since in this case the velocity multiscale space is as big as the velocity snapshot space. We review that the dimension of a local velocity snapshot space is equal to the number of fine-scale edges on a single coarse-scale edge. From observations, we can obtain two main conclusions. Firstly, as more offline bases are used, the velocity error steadily decreases. As one can observe from the Table 2, the error of $\kappa_1$ decreases from 22.1% at “3+0” to 3.11% at “20+0”, where 20 is the maximum for offline bases used in each local region. Secondly, the residual-driven bases can remarkably reduce errors. In other words, one can obtain a good accuracy with relatively few residual-driven bases. The error (3.47%) corresponding to “2 + 1” case in $\kappa_1$ is comparable to the “20 + 0” case (3.11%), which is sufficient to demonstrate the power of residual-driven bases. Since the “2 + 1” case is nearly as accurate as the “best” case (“20+0”), it is reasonable that adding more residual-driven bases can not provide an apparent error reduction. As one can see, the “2+1” case has error 3.47% while the “2 + 2” has error 3.34%. The error reduction from adding residual-driven bases is also significant when $\kappa_3$ is used. The error in “3 + 1” case is 0.37%, less than 1/15 of that in the “3 + 0” case. Apart from the two above conclusions, we can also observe that there is an evident accuracy improvement as one enriches the multiscale space from “1 + 1” to “2 + 1” bases. More specifically, errors associated
with $\kappa_3$ decrease from 9.05% to 0.88% by starting with more than one offline bases. The results from $\eta = \frac{1}{16}$ are recorded in the Table 3, which share some similarities with the case $\eta = \frac{1}{8}$. The $L_2$ errors corresponding to $\kappa_1$ decreases from 20.98% at the “3 + 0” case to 4.14% at the “20 + 0” case. After adding one single residual-driven basis, the error straightly decreases to 4.32% at the “3 + 1” case, approximately as low as the “20 + 0” case. Hence, only one residual-driven basis can improve the approximation such that it is nearly as accurate as the snapshot solution, which is sufficient to show the strong power of residual-driven bases. Moreover, when we compare the above two cases (\(\eta = \frac{1}{8}, \frac{1}{16}\)), we can find that errors are slightly larger with a smaller correlation length $\eta$. This is resulted from the fact that when $\eta$ is larger, the covariance function defined in $f_5$ will contribute to a more complicated permeability field while the velocity solutions are closely related with the permeability field. Hence, a smaller correlation length will increase the difficulty in approximating the velocity solutions.

We now show the effect of correlation lengths on coarse-scale approximations. We further demonstrate the result when different sources are used in training and testing stages. In Figure 5, we plot velocity errors corresponding to 500 samples, where the correlation lengths are $\eta = \frac{1}{8}, \frac{1}{16}$. For the smaller $\eta$, we both present the errors when the sources are different or the same in offline and online stages. With the same source $f_1$, the mean error of $\eta = \frac{1}{8}$ is 0.0271 and the variance is $1.59 \times 10^{-4}$. For $\eta = \frac{1}{16}$, the mean error is 0.052 and the variance is $3.98 \times 10^{-4}$. With $f_1$ in testing and $f_2$ in testing, the mean error of $\eta = \frac{1}{8}$ is 0.0320 and variance is $1.59 \times 10^{-4}$. As one can verify from the figures, the errors corresponding to a smaller correlation length is more divergent than those corresponding to a larger correlation length. This reflects that there is a higher variation among different sample permeability fields that are generated with a smaller correlation length. Moreover, the errors corresponding to different sources in training and testing are only slightly larger than the ones with same sources, which to some extent shows a good generalization power of our method.

![Figure 5: Velocity errors of 500 samples with $\eta = 1/8, 1/16$. Left: $\eta = 1/8$. Mean: 0.0320. Variance: $1.44 \times 10^{-4}$. We use $f_1$ in training while $f_2$ is used in testing. Middle: $\eta = 1/8$. Mean: 0.0271. Variance: $1.59 \times 10^{-4}$. $f_1$ is used in both training and testing. Right: 1/16. Mean: 0.053. Variance: $3.98 \times 10^{-4}$. $f_1$ is used in both training and testing. The number of multiscale bases are “2 + 1”.

In the three-dimensional case, we choose $\eta = \frac{1}{4}, \frac{1}{8}$ for both SPE model and high-contrast model. In the Table 4 and 5, we show the errors associate with $\kappa_4$ and $\kappa_5$, respectively. According to the numbers of fine-scale elements in a single coarse-scale element, the dimensions of local velocity snapshot spaces for $\kappa_4$ and $\kappa_5$ are 100 and 64. Here, we use at most 32 offline multiscale bases in each local region. Since the numbers of offline bases have not attained the maximums, i.e. 100 for $\kappa_4$ and 64 for $\kappa_5$, the shown smallest velocity error without using residual-driven bases is attained with “32 + 0” bases, which is still relatively high, especially for $\kappa_4$ case with the error about 20%. Moreover, there is a remarkable accuracy improvement from the “32 + 0” to the “1 + 1” case with $\kappa_4$. In particular, with $\eta = \frac{1}{4}$, the error 2% in the “1 + 1” case is less than 1/10 of that in the “3 + 0” case, which suffices to show the significance of residual-driven bases. On the other hand,
the effect of adding residual-driven bases with \( \kappa_5 \) is not as significant as the one with \( \kappa_4 \) but it is still manifest. The error with “1 + 1” bases accounts for nearly 1/8 of that with “32 + 0” bases. Hence, we can see that the error reduction in three-dimensional case is remarkable by including residual-driven bases. Also, the improvement is greater when the permeability field shows greater heterogeneity. Besides, we show the errors corresponding to a smaller correlation length \( \frac{1}{8} \) in the last columns of the Table 4 and 5. The error decay shares some similarities with the case \( \eta = \frac{1}{4} \). One could observe that the residual-driven bases are more powerful in \( \kappa_4 \) than \( \kappa_5 \). Besides, due to a smaller correlation length, a higher level of randomness is included in the stochastic permeability field and thus yields a larger average error. When \( \eta = \frac{1}{8} \), the smallest error 1% corresponding to \( \kappa_4 \) is attained with the case of “4 + 2”, which is bigger than 0.5% with \( \eta = \frac{1}{4} \).

| \( \kappa_1 \) | 220 \times 60 | 11 \times 3 | \( \eta = 1/4 \) | 38 | 136 |
| \( \kappa_2 \) | 220 \times 60 | 11 \times 3 | \( \eta = 1/8 \) | \( \eta = 1/16 \) |
| \( \kappa_3 \) | 128 \times 128 | 8 \times 8 | 46 | 150 |
| \( \kappa_4 \) | 220 \times 60 \times 30 | 22 \times 6 \times 3 | 48 | 283 | \( \eta = 1/8 \) |
| \( \kappa_5 \) | 64 \times 4 | 8 \times 4 | 48 | 281 | \( \eta = 1/16 \) |

Table 1: Fine-scale and coarse-scale mesh sizes (second and third columns), numbers of truncated KL expansion terms (last three columns) in 2D and 3D permeability fields with two different correlation lengths \( \eta \). \( \kappa_1 \), \( \kappa_2 \) and \( \kappa_3 \) are three 2D permeability fields; \( \kappa_4 \) and \( \kappa_5 \) are two 3D permeability fields. In 2D case, we use \( \eta = 1/8, 1/16 \), while in 3D case, \( \eta = 1/4, 1/8 \).

| \( \eta \) | \( e_v(\kappa_1) \) | \( e_v(\kappa_2) \) | \( e_v(\kappa_3) \) |
| 3+0 | 0.2110 | 0.0819 | 0.0631 |
| 6+0 | 0.1107 | 0.0426 | 0.0335 |
| 8+0 | 0.0575 | 0.0379 | 0.0302 |
| 16+0 | 0.0317 | 0.0307 | 0.0030 |
| 20+0 | 0.0311 | 0.0305 | \( / \) |
| 1+1 | 0.0537 | 0.0540 | 0.0905 |
| 2+1 | 0.0347 | 0.0360 | 0.0088 |
| 2+2 | 0.0334 | 0.0334 | 0.0039 |
| 3+1 | 0.0331 | 0.0330 | 0.0037 |
| 4+1 | 0.0323 | 0.0360 | 0.0037 |
| 5+1 | 0.0324 | 0.0317 | 0.0037 |
| 4+2 | 0.0320 | 0.0319 | 0.0036 |

Table 2: Average velocity errors corresponding to \( \kappa_1 \), \( \kappa_2 \) and \( \kappa_3 \) with the correlation length 1/8. 500 samples are included.
Table 3: Average velocity errors corresponding to $\kappa_1$, $\kappa_2$ and $\kappa_3$ with the correlation length $1/16$. 500 samples are included.

| $N_b$ | $\epsilon_v(\kappa_1)$ | $\epsilon_v(\kappa_2)$ | $\epsilon_v(\kappa_3)$ |
|-------|-------------------------|-------------------------|-------------------------|
| 3+0   | 0.2098                  | 0.0855                  | 0.0640                  |
| 6+0   | 0.1130                  | 0.0489                  | 0.0349                  |
| 8+0   | 0.0626                  | 0.0451                  | 0.0316                  |
| 16+0  | 0.0417                  | 0.0383                  | 0.0083                  |
| 20+0  | 0.0414                  | 0.0380                  | /                       |
| 1+1   | 0.0619                  | 0.0568                  | 0.1111                  |
| 2+1   | 0.0441                  | 0.0429                  | 0.0116                  |
| 2+2   | 0.0435                  | 0.0411                  | 0.0087                  |
| 3+1   | 0.0432                  | 0.0406                  | 0.0085                  |
| 4+1   | 0.0431                  | 0.0397                  | 0.0085                  |
| 5+1   | 0.0436                  | 0.0397                  | 0.0084                  |
| 4+2   | 0.0430                  | 0.0398                  | 0.0084                  |

Table 4: Numbers of unknowns (Dof) and computational time for $\kappa_4$. Average velocity errors corresponding to $\kappa_4$ with correlation lengths $1/4$ and $1/8$. 500 samples are included. $T_{\text{train}}$ is the CPU time for constructing the multiscale space and $T_{\text{test}}$ is the CPU time for solving a corresponding coarse-scale equation for a single sample.

| $N_b$ | Dof  | $T_{\text{train}}$ (s) | $T_{\text{test}}$ (s) | $\epsilon_v(\kappa_4; 1/4)$ | $\epsilon_v(\kappa_4; 1/8)$ |
|-------|------|------------------------|------------------------|-----------------------------|-----------------------------|
| 3+0   | 3312 | 10.56                  | 2.54                   | 0.256                       | 0.257                       |
| 8+0   | 8172 | 19.30                  | 6.07                   | 0.218                       | 0.218                       |
| 16+0  | 15948| 38.71                  | 15.98                  | 0.212                       | 0.212                       |
| 32+0  | 31500| 90.97                  | 61.90                  | 0.202                       | 0.201                       |
| 1+1   | 2340 | 96.00                  | 2.11                   | 0.020                       | 0.021                       |
| 2+1   | 3312 | 91.13                  | 2.41                   | 0.011                       | 0.013                       |
| 2+2   | 4284 | 150.71                 | 3.24                   | 0.006                       | 0.010                       |
| 2+3   | 5256 | 211.45                 | 3.95                   | 0.006                       | 0.011                       |
| 3+2   | 5256 | 98.21                  | 3.24                   | 0.006                       | 0.011                       |
| 3+1   | 4284 | 166.95                 | 4.12                   | 0.008                       | 0.011                       |
| 4+1   | 5256 | 111.79                 | 3.97                   | 0.008                       | 0.012                       |
| 5+1   | 6228 | 122.00                 | 4.54                   | 0.006                       | 0.010                       |
| 4+2   | 6228 | 187.16                 | 4.49                   | 0.005                       | 0.010                       |
Table 5: Numbers of unknowns (Dof) and computational time for \( \kappa_5 \). Average velocity errors corresponding to \( \kappa_5 \) with correlation lengths 1/4 and 1/8. 500 samples are included. \( T_{\text{train}} \) is the CPU time for constructing the multiscale space and \( T_{\text{test}} \) is the CPU time for solving a corresponding coarse-scale equation for a single sample.

| \( N_b \) | Dof  | \( T_{\text{train}} \) (s) | \( T_{\text{test}} \) (s) | \( \varepsilon_v(\kappa_5; 1/4) \) | \( \varepsilon_v(\kappa_5; 1/8) \) |
|----------|------|-----------------|-----------------|----------------|----------------|
| 3+0      | 4544 | 8.05            | 2.91            | 0.356          | 0.355          |
| 8+0      | 11246| 22.01           | 11.17           | 0.235          | 0.235          |
| 16+0     | 22016| 64.65           | 37.85           | 0.091          | 0.091          |
| 32+0     | 43520| 216.70          | 149.94          | 0.031          | 0.031          |
| 1+1      | 3200 | 49.22           | 1.50            | 0.046          | 0.047          |
| 2+1      | 4544 | 64.99           | 2.92            | 0.045          | 0.047          |
| 2+2      | 5888 | 118.09          | 4.34            | 0.006          | 0.015          |
| 2+3      | 7232 | 179.82          | 5.80            | 0.004          | 0.014          |
| 3+1      | 5888 | 77.49           | 4.27            | 0.044          | 0.045          |
| 3+2      | 7232 | 139.41          | 7.65            | 0.006          | 0.014          |
| 4+1      | 7232 | 92.81           | 5.791           | 0.043          | 0.045          |
| 4+2      | 8576 | 166.07          | 7.62            | 0.005          | 0.014          |
| 5+1      | 8576 | 107.04          | 7.25            | 0.044          | 0.045          |

5.2 Two-phase flow and transport

In this subsection, we consider a two-phase flow problem in a bounded domain \( \Omega \in \mathbb{R}^2 \). The two phases are water and oil, denoted by \( w \) and \( o \), respectively. In particular, water is injected to a heterogeneous reservoir, which displaces the trapped oil to a production well. To simplify the model, we assume there is no capillary pressure and only consider a gravity-free environment. Moreover, the two phases are assumed to be immiscible and incompressible. The two-phase problem is modeled by an elliptic equation coupled with a transport equation. Before presenting the equations, we first list some notations.

1. \( S \): water saturation; \( v \): total Darcy velocity; \( p \): pressure.
2. \( \kappa_{\alpha}, \mu_{\alpha} \): relative permeability and viscosity of phase \( \alpha, \ alpha = w, o \).
3. \( f, r \) are external forcing terms.
4. \( \xi(S) \): total mobility.
5. \( F(S) \): fractional flow function for water phase.
6. \( \partial D_{\text{out}} \): outer flow boundary.

Under the above assumptions, the two-phase flow problem is formulated as a simplified coupling system, where the flow equation is as follows:

\[
-\xi(S)\kappa(\mathbf{x}; \omega)\nabla p = v \text{ in} \ \Omega, \\
\text{div}(v) = f \text{ in} \ \Omega, \\
v \cdot n = 0 \text{ on} \ \partial\Omega.
\] (58)

\( \xi(S) \) is defined by

\[
\xi(S) := \frac{\kappa_{rw}(S)}{\mu_w} + \frac{\kappa_{ro}(S)}{\mu_o}.
\]
Here

\[ \kappa_{rw}(S) = S^2, \quad \kappa_{ro}(S) = (1 - S)^2, \quad \mu_w = 1, \quad \mu_o = 5. \]

Moreover, the transport equation is

\[ S_t + \text{div}(F(S)v) = r, \tag{59} \]

where \( F(S) \) is defined as

\[ F(S) := \frac{\kappa_{rw}(S)/\mu_w}{\kappa_{rw}(S)/\mu_w + \kappa_{ro}(S)/\mu_o}. \]

In this work, \( f \) and \( r \) are the same, which are defined in (56).

There is an extremely essential quantity called water cut in the above flow problem, which describes the composition of oil-water two-phase flow at the producer. The measurement of water cut can be used to improve oil recovery ratio. Water cut is defined to be

\[ q_w / (q_w + q_o), \]

where \( q_w := \int_{\partial D_{out}} F(S)v \cdot n dS \) and \( q_t := \int_{\partial D_{out}} v \cdot n dS \).

We then describe our scheme to solve the two-phase flow problem. To solve (59), we use the finite volume method on the fine grid. In particular, in each fine-scale element \( \zeta_i \), we solve the following equation,

\[ |\zeta_i| S_{\eta, i}^{n+1} - S_{\eta, i}^n \Delta t + \int_{\partial \zeta_i} \hat{S}^n (v \cdot n) = r_i |\zeta_i|, \]

where \( |\zeta_i| \) is the area of \( \zeta_i \) and \( \hat{S} \) is the upwind flux. Besides, \( r_i \) is the average value of \( r \) on \( \zeta_i \).

To solve the coupling system formed by (58) and (59), we apply a standard Implicit Pressure Explicit Saturation (IMPES) scheme \[16, 17\]. We define \( S^n \) and \( v^n \) to be the saturation and velocity at the time step \( n \). In particular, for a given \( S^n \) from the time step \( n \), we implicitly solve the flow equation (58) to obtain a velocity solution \( v^{n+1} \) at the time step \( n + 1 \). Then we explicitly solve \( S^{n+1} \) at the time \( t_{n+1} \) using (59). In other words, we update the saturation with the velocity solution at the current time step. We give the procedures of the IMPES scheme in Table 6. Since a velocity solution computed with mixed finite element method is called a reference velocity solution, we call the corresponding saturation the reference saturation. Similarly, the saturation updated with multiscale velocity bases is denoted by multiscale saturation or approximation of saturation. We will use these definitions in the following discussions.

| Table 6: The IMPES scheme |
|---------------------------|
| Input: \( S^n \) |
| Step 1: Seek \( v^{n+1} \) with (58) using \( S^n \). |
| Step 2: Seek \( S^{n+1} \) with (59) using \( S^n \) and \( v^{n+1} \). |
| Output: \( S^{n+1} \). |

In a fixed model case, where the permeability field is deterministic, we measure the saturation error by the relative difference in \( L^2 \) norm between a reference solution and a corresponding approximation; in stochastic case, where different permeability fields are used, we use the relative error of the average saturation computed using MGMsFEM according to the average reference saturation. The mathematical definitions of the concerned errors are given below:

\[ e_{s,\text{fixed}} := \frac{\int_{\Omega} |s_f - s_{ms}|^2}{\int_{\Omega} |s_f|^2}, \quad e_{s,\text{stoc}} := \frac{\int_{\Omega} |\bar{s}_f - \bar{s}_{ms}|^2}{\int_{\Omega} |\bar{s}_f|^2}, \tag{60} \]

where \( \bar{s}_f \) and \( \bar{s}_{ms} \) are defined to be the average reference saturation and the average multiscale saturation.
In Figure 6, we show the dynamics of saturation errors associated with different numbers of multiscale bases, i.e. “1 + 1”, “2 + 1”, “4 + 0” and “8 + 0”. The results are from three groups: a fixed permeability field, two types of random permeability fields with correlation lengths $\eta = \frac{1}{4}$, $\frac{1}{8}$. For each choice of correlation length, we generate 500 sample permeability fields. We then compute the approximation of saturation and the reference saturation for each sample. As shown in (59), one needs to update velocity in a time-marching process. The approximation of velocity is updated by MGMsFEM, while the reference velocity is solved by the mixed finite element method on the fine grid. After we compute 500 approximations and reference solutions of saturation, we compute the average approximation $s_{\text{ms}}$ and the average reference saturation $\bar{s}_f$ and compute the $e_{s,\text{stoc}}$ defined in (60). We place the results from the fixed model in the middle and the two random cases on both sides for a better comparison. Four observations could be obtained. First of all, the performance of multiscale bases are almost the indistinguishable in three cases, which means that the randomness included in the permeability field has a small impact on the approximation effects of multiscale bases. Thus, the extension of MGMsFEM to the concerned two-phase model in random case is shown to be effective. Secondly, the saturation errors corresponding to different numbers of multiscale bases change in a similar pattern. Specifically, the errors first increase from the initial time to an intermediate time point and then decreases to a level, where it remains steady. The turning points of the four curves are around $t = 300$, where we updated the multiscale bases based on the latest saturation. Moreover, residual-driven enrichment is powerful if more than one bases are utilized in each local region in offline stage I. As shown in the graph, the “1 + 1” case has the largest error among four instances. In particular, the error has exceeded 15% at the peak time. However, once one more offline basis function is added, the error dramatically declined even though only one residual-driven basis is used. More specifically, the “2+1” approximation becomes the most accurate one, which is even better than “8 + 0” case. Hence, the residual-driven bases are powerful when sufficient offline bases are used. Last but not least, the effect of using more offline bases is still apparent when no residual-driven bases are involved. In Figure 6, an evident improvement can be noticed from “4 + 0” to “8 + 0”. More specifically, the error is almost halved when the number of bases is doubled. Consequently, even though the residual-driven bases are powerful, one cannot disregard the significance of offline bases.

In Figure 7, we compare an average reference saturation with an average multiscale saturation associated with “2+1” bases respectively, where the averages are computed among 100 samples. To better visualize the comparison, the absolute difference between the above two concerned quantities is shown in the last column. We demonstrate two time steps: $t = 250$ and $t = 750$. The reference saturation is shown on the left. The flows move from the four vertical edges and extend to the center. One can see that the simulation is faster where the corresponding permeability shows a greater variation. In particular, the flows starting from the lower-left and upper-right corners shift further than the other two corners within the same time. The simulations of $t = 250$ and $t = 750$ are displayed in the first and second rows. The flows have not intersected at the center until $t = 750$. At $t = 250$, the multiscale saturation is a relatively accurate approximation of the reference saturation because most detailed information is shown to be preserved in the multiscale saturation. In other words, it is difficult to find noticeable differences between them. At $t = 750$, more details are displayed. Overall, the multiscale saturation is indistinguishable from the reference. However, there is an observable distinction at the lower-right corner, where the permeability shows strong discontinuity in the subfigure 4(d). The permeability shown in Figure 4 is highly discontinuous at this region, which reveals some connections between the permeability field and the saturation. In other places, there are barely big deviations between the approximation and the fine saturation, which shows the effect of multiscale bases.

In Figure 8, we show water-cut curves corresponding to a reference and four approximations. The reference water cut is computed by MFEM on a fine grid hence we use “fine” to denote this curve. Four approximation water cut curves are solved with MGMsFEM, where the numbers of multiscale bases are “1 + 1”, “2 + 1”, “4 + 0”, and “8 + 0”. We demonstrate the comparisons with $\eta = \frac{1}{4}$, $\frac{1}{8}$ from $t = 0$ to 1000. From the two figures, one can observe that all the water cuts first
increase slowly then fast and tend steady, which shows the dynamics of fraction of water at the producer. The differences between the approximations and the references are indistinguishable, which holds for all the approximations. Water cut is indeed a function of saturation at specific points, which is the reason that the approximations with only bases in offline stage I are nearly as accurate as those computed with bases in both two offline stages. It is worth mentioning that even though the approximation bias can be noticed at the middle time, the errors tend to vanish as the water tends to be saturated.

Figure 6: Comparison of saturation errors of different numbers of multiscale bases: “1+1”, “2+1”, “4+0”, and “8+0”. Left: error of the average saturations (the mean of 100 samples) with $\eta = 1/4$; middle: error of the saturation corresponding to $\kappa_4$; right: error of the average saturations (the mean of 100 samples) with $\eta = 1/8$. $t$ ranges from 0 to 1000.

Figure 7: Comparison of the average fine saturation and the average multiscale saturation with “2+1” bases at $t = 250$ and $t = 750$. 100 samples are used, where the mean permeability field is $\kappa_4$. 
Figure 8: Average water-cut curves computed with different methods under two choices of correlation lengths, $\eta = 1/4, 1/8$. Five curves are corresponding to MFEM and MGMsFEM with four numbers of multiscale bases (“8+0”, “4+0”, “2+1” and “1+1”). 100 samples are used, where the mean permeability field is $\kappa_4$.

6 Conclusion

In this work, our objective is to provide a computationally efficient method for stochastic groundwater flow problems. In particular, we apply a mixed generalized multiscale finite element method to a single-phase flow equation and a two-phase flow equation coupled with a transport equation, where randomness is included in permeability fields of the above two problems. With a well-chosen permeability field, one can construct a set of locally-defined multiscale bases during two stages in training, which will be further used to solve equations in a coarse mesh with an arbitrary permeability field in a testing stage.

We utilize a mixed formulation of the flow equation to achieve a conservation of mass in local regions, which is essential in flow problems. The multiscale space in offline stage I is constructed based on a much larger snapshot space, where snapshot bases are resulted from different boundary conditions. Using a set of well-designed spectral problems, we can distill some important modes from the snapshot space. Since the constructions are performed independently in each local patch, one can use a parallel computing to achieve higher efficiency. Based on a set of local residuals derived from the equation, some residual-driven bases are generated to enhance the accuracy of the solutions to the flow equation and further decrease the errors in saturation solutions to the transport equation in the two-phase model. We emphasize that although the residual-driven construction is conducted in a fine-grid mesh, the computations are restricted in some local patches. In other words, the computation cost exists in the residual-driven enrichment is limited and the constructions of residual-driven bases can also be implemented in parallel. Overall, the final multiscale space is spanned by multiscale bases in two offline stages. Because the total number of bases is remarkably reduced compared with the previous snapshot space, the computation is significantly accelerated. Hence, our method can be employed to speed up the uncertainty quantification problems in subsurface flows, where a large number of samples are investigated. What’s more, the proposed method does not require any interpolation in the spatial domain and can be applied to solve the equation associated with any realizations sampled from a large stochastic space.

We demonstrate numerical simulations for a single-phase flow and a two-phase flow. More specifically, we use stochastic high-contrast models and SPE models. For each model, both two-
dimensional and three-dimensional cases are investigated. For the single-phase flow, our method focuses on reducing the velocity errors, while for the latter one, a good accuracy in the saturation solutions is the target. Our results show that the multiscale space generated with a training field can be applied to different sample fields as well as different sources. Furthermore, some residual-driven bases can improve the overall accuracy. In particular, in the single-phase flow, we show test errors corresponding to sample fields and a source that are different from what are used in the training stage. The errors are small if sufficient bases are used. The use of residual-driven bases can remarkably increase the accuracy and efficiency especially in 3D case. For the two-phase flow model, we show a comparison of reference saturations and approximations of saturation at two specific time points. The approximations can well capture most details in the references, which means the differences between these two saturations are almost difficult to distinguish. Besides, we also compute water cut in the context of two-phase flow. Since water cut is a deterministic function of saturation, the approximations of water cut based on multiscale space are pretty accurate compared to the reference water cut, which is similar to approximation reference saturations. Besides, we would like to remark that the proposed method can be extended to a more general case, where different types of grids (not necessarily Cartesian grids) are used for a spatial discretization.

Generally, the proposed method is promising in dealing with groundwater flow problems, which is guaranteed by theoretical proofs and verified in numerical results. In the future, there are still some interesting and non-trivial problems to explore. For example, we may consider a more sophisticated random space and combining our method with a stochastic collocation method [5]. Besides, we can further apply the proposed algorithm to some inverse problems, where the associated forward process can solved by our method. Last but not least, this method can serve as a base to some non-intrusive methods like deep learning. One can combine this method with deep learning as our recent work [39].

Acknowledgement

The research of Eric Chung is partially supported by the Hong Kong RGC General Research Fund (Project numbers 14304719 and 14302620) and CUHK Faculty of Science Direct Grant 2020-21.

References

[1] J.E. Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Modeling & Simulation*, 2:421–439, 2004.

[2] Jørg E Aarnes, Yalchin Efendiev, and Lijian Jiang. Mixed multiscale finite element methods using limited global information. *Multiscale Modeling & Simulation*, 7(2):655–676, 2008.

[3] T. Arbogast, G. Pencheva, M.F. Wheeler, and I. Yotov. A multiscale mortar mixed finite element method. *Multiscale Modeling & Simulation*, 6(1):319–346, 2007.

[4] Todd Arbogast and Kirsten J Boyd. Subgrid upscaling and mixed multiscale finite elements. *SIAM Journal on Numerical Analysis*, 44(3):1150–1171, 2006.

[5] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[6] John W Barker and Sylvain Thibeau. A critical review of the use of pseudorelative permeabilities for upscaling. *SPE Reservoir Engineering*, 12(02):138–143, 1997.
[7] Franco Brezzi and Michel Fortin. Variational formulations and finite element methods. In *Mixed and hybrid finite element methods*, pages 1–35. Springer, 1991.

[8] Ido Bright, Guang Lin, and J Nathan Kutz. Classification of spatiotemporal data via asynchronous sparse sampling: Application to flow around a cylinder. *Multiscale Modeling & Simulation*, 14(2):823–838, 2016.

[9] Lawrence Bush and Victor Ginting. On the application of the continuous Galerkin finite element method for conservation problems. *SIAM Journal on Scientific Computing*, 35(6):A2953–A2975, 2013.

[10] V. Calo, Y. Efendiev, and J. Galvis. A note on variational multiscale methods for high-contrast heterogeneous flows with rough source terms. *Advances in Water Resources*, 34:1177–1185, 2011.

[11] Ho Yuen Chan, Eric T Chung, and Yalchin Efendiev. Adaptive mixed GMsFEM for flows in heterogeneous media. *Numerical Mathematics: Theory, Methods and Applications*, 9(4):497–527, 2016.

[12] G. A. Chechkin, A. L. Piatniski, and A. S. Shamev. *Homogenization: Methods and Applications*, volume 234 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 2007.

[13] Ke Chen, Qin Li, Jianfeng Lu, and Stephen J Wright. Randomized sampling for basis function construction in generalized finite element methods. *Multiscale Modeling & Simulation*, 18(2):1153–1177, 2020.

[14] Ke Chen, Qin Li, Jianfeng Lu, and Stephen J Wright. A low-rank schwarz method for radiative transfer equation with heterogeneous scattering coefficient. *Multiscale Modeling & Simulation*, 19(2):775–801, 2021.

[15] Z. Chen and T.Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, 72:541–576, 2002.

[16] Zhangxin Chen, Guanren Huan, and Baoyan Li. An improved IMPES method for two-phase flow in porous media. *Transport in porous media*, 54(3):361–376, 2004.

[17] Zhangxin Chen, Guanren Huan, and Yuanle Ma. *Computational methods for multiphase flows in porous media*. SIAM, 2006.

[18] E. Chung, Y. Efendiev, and W. Leung. An online generalized multiscale discontinuous Galerkin method (GMsDGM) for flows in heterogeneous media. *Communications in Computational Physics*, 21(2):401–422, 2017.

[19] Eric Chung, Yalchin Efendiev, Yanbo Li, and Qin Li. Generalized multiscale finite element method for the steady state linear boltzmann equation. *Multiscale Modeling & Simulation*, 18(1):475–501, 2020.

[20] Eric T Chung, Yalchin Efendiev, and Chak Shing Lee. Mixed generalized multiscale finite element methods and applications. *Multiscale Modeling & Simulation*, 13(1):338–366, 2015.

[21] Eric T Chung, Yalchin Efendiev, and Wing Tat Leung. Residual-driven online generalized multiscale finite element methods. *Journal of Computational Physics*, 302:176–190, 2015.

[22] Louis J Durlofsky. Coarse scale models of two phase flow in heterogeneous reservoirs: volume averaged equations and their relationship to existing upscaling techniques. *Computational Geosciences*, 2(2):73–92, 1998.
[23] Y. Efendiev, J. Galvis, S. Ki Kang, and R.D. Lazarov. Robust multiscale iterative solvers for nonlinear flows in highly heterogeneous media. *Numer. Math. Theory Methods Appl.*, 5(3):359–383, 2012.

[24] Hadi Hajibeygi, Giuseppe Bonfigli, Marc Andre Hesse, and Patrick Jenny. Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227(19):8604–8621, 2008.

[25] Christopher Harder, Diego Paredes, and Frédéric Valentin. A family of Multiscale Hybrid-Mixed finite element methods for the Darcy equation with rough coefficients. *Journal of Computational Physics*, 245:107 − 130, 2013.

[26] Thomas Y Hou, Qin Li, and Pengchuan Zhang. Exploring the locally low dimensional structure in solving random elliptic pdes. *Multiscale Modeling & Simulation*, 15(2):661–695, 2017.

[27] Thomas Y Hou, Dingjiong Ma, and Zhiwen Zhang. A model reduction method for multiscale elliptic PDEs with random coefficients using an optimization approach. *Multiscale Modeling & Simulation*, 17(2):826–853, 2019.

[28] SP Huang, ST Quek, and KK Phoon. Convergence study of the truncated Karhunen–Loève expansion for simulation of stochastic processes. *International journal for numerical methods in engineering*, 52(9):1029–1043, 2001.

[29] Oleg Iliev, Raytcho Lazarov, and Joerg Willems. Fast numerical upscaling of heat equation for fibrous materials. *Computing and visualization in science*, 13(6):275–285, 2010.

[30] V. Kippe J. E. Aarnes and K. A. Lie. Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. *Advances in Water Resources*, 28(257 – 271), 2005.

[31] L. Jiang and I.D. Mishev. Mixed multiscale finite volume methods for elliptic problems in two-phase flow simulations. *Commun. Comput. Phys.*, 11(1):19–47, 2012.

[32] Lijian Jiang and Qiuqi Li. Reduced multiscale finite element basis methods for elliptic PDEs with parameterized inputs. *Journal of Computational and Applied Mathematics*, 301:101–120, 2016.

[33] Shi Jin, Peng Qi, and Zhiwen Zhang. An Eulerian surface hopping method for the Schrödinger equation with conical crossings. *Multiscale Modeling & Simulation*, 9(1):258–281, 2011.

[34] Huan Lei, Xiu Yang, Bin Zheng, Guang Lin, and Nathan A Baker. Constructing surrogate models of complex systems with enhanced sparsity: quantifying the influence of conformational uncertainty in biomolecular solvation. *Multiscale Modeling & Simulation*, 13(4):1327–1353, 2015.

[35] Sijing Li, Zhiwen Zhang, and Hongkai Zhao. A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction. *Multiscale Modeling & Simulation*, 18(3):1242–1271, 2020.

[36] Lars H Odsæter, Mary F Wheeler, Trond Kvamsdal, and Mats G Larson. Postprocessing of non-conservative flux for compatibility with transport in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 315:799–830, 2017.

[37] Na Ou, Guang Lin, and Lijian Jiang. A low-rank approximated multiscale method for PDEs with random coefficients. *Multiscale Modeling & Simulation*, 18(4):1595–1620, 2020.

[38] Maria Vasilyeva, Masoud Babaei, Eric T Chung, and Denis Spiridonov. Multiscale modeling of heat and mass transfer in fractured media for enhanced geothermal systems applications. *Applied Mathematical Modelling*, 67:159–178, 2019.
[39] Yiran Wang, Eric Chung, and Shubin Fu. A deep learning based reduced order modeling for stochastic underground flow problems. *arXiv preprint arXiv:2111.13372*, 2021.

[40] Yiran Wang, Eric Chung, and Shubin Fu. A local–global multiscale method for highly heterogeneous stochastic groundwater flow problems. *Computer Methods in Applied Mechanics and Engineering*, 392:114688, 2022.

[41] Yiran Wang, Eric Chung, Shubin Fu, and Zhaoqin Huang. A comparison of mixed multiscale finite element methods for multiphase transport in highly heterogeneous media. *Water Resources Research*, 57(5):e2020WR028877, 2021.

[42] Yiran Wang, Eric Chung, Shubin Fu, and Michael Presho. Online conservative generalized multiscale finite element method for highly heterogeneous flow models. *Computational Geosciences*, pages 1–17, 2021.

[43] Yixuan Wang, Hadi Hajibeygi, and Hamdi A Tchelepi. Monotone multiscale finite volume method. *Computational Geosciences*, 20(3):509–524, 2016.

[44] Zhiliang Xu, Yingjie Liu, Huijing Du, Guang Lin, and Chi-Wang Shu. Point-wise hierarchical reconstruction for discontinuous Galerkin and finite volume methods for solving conservation laws. *Journal of Computational Physics*, 230(17):6843–6865, 2011.

[45] Yanfang Yang, Eric T Chung, and Shubin Fu. An enriched multiscale mortar space for high contrast flow problems. *Communications in Computational Physics*, 23(2):476–499, 2018.

[46] Yanfang Yang, Shubin Fu, and Eric T Chung. Online mixed multiscale finite element method with oversampling and its applications. *Journal of Scientific Computing*, 82(2):1–20, 2020.

[47] Zhiwen Zhang, Maolin Ci, and Thomas Y Hou. A multiscale data-driven stochastic method for elliptic PDEs with random coefficients. *Multiscale Modeling & Simulation*, 13(1):173–204, 2015.