Comput. Methods Appl. Mech. Engrg. 200 (2011) 1251–1262

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

A multiscale preconditioner for stochastic mortar mixed finite elements

Mary F. Wheeler [b], Tim Wildey [b,*], Ivan Yotov [a]

[a] Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA
[b] The Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, Austin, TX 78712, USA

ARTICLE INFO

Article history:
Received 16 June 2009
Received in revised form 6 August 2010
Accepted 11 October 2010
Available online 27 October 2010

Keywords:
Stochastic collocation
Mixed finite element
Stochastic partial differential equations
Flow in porous media
Interface preconditioner

ABSTRACT

The aim of this paper is to introduce a new approach to efficiently solve sequences of problems that typically arise when modeling flow in stochastic porous media. The governing equations are based on Darcy's law with a stochastic permeability field. Starting from a specified covariance relationship, the log permeability is decomposed using a truncated Karhunen–Loève expansion. Multiscale mortar mixed finite element approximations are used in the spatial domain and a nonintrusive sampling method is used in the stochastic dimensions. A multiscale mortar flux basis is computed for a single permeability, called a *training permeability*, that captures the main characteristics of the porous media, and is used as a preconditioner for each stochastic realization. We prove that the condition number of the preconditioned interface operator is independent of the subdomain mesh size and the mortar mesh size. Computational results confirm that our approach provides an efficient means to quantify the uncertainty for stochastic flow in porous media.

1. Introduction

In groundwater flow problems, it is physically impossible to know the exact permeability at every point in the domain. This is due to the prohibitively large scope of realistic domains, inhomogeneity in the media, and also the natural randomness occurring at very small scales. One way to cope with this difficulty is to model permeability (or porosity) as a stochastic function, determined by an underlying random field with an experimentally determined covariance structure.

The development of efficient stochastic methods that are applicable for flow in porous media has drawn significant interest in recent years [13,48]. Stochastic sampling methods have grown in popularity due to their nonintrusive nature in terms of modifying a legacy simulation code. The best known sampling method is Monte Carlo simulation (MCS), which involves repeated generation of random samplings (realizations) of input parameters followed by the application of the simulation model in a "black box" fashion to generate the corresponding set of stochastic responses. These responses are further analyzed to yield statistical moments or distributions. The major drawback of MCS is the high computational cost due to the need to generate valid representative statistics from a large number of realizations at a high resolution level.

One approach for improving the efficiency of non-sampling methods is the stochastic collocation method [5,45,44]. It combines a finite element discretization in physical space with a collocation at specially chosen points in probability space. As a result, a sequence of uncoupled deterministic problems need to be solved, just like in MCS. However, the stochastic collocation method shares the approximation properties of the stochastic finite element method [6,41,15], making it more efficient than MCS. Choices of collocation points include tensor product of zeros of orthogonal polynomials [5,45], sparse grid approximations [17,32,40,45], and probabilistic collocation [26].

A tremendous amount of research over the past thirty years has been devoted to the efficient parallel solution of deterministic subsurface flow problems. Non-overlapping domain decomposition methods are popular due to their ease of parallel implementation, physically meaningful interface conditions, and ability to handle different physical models in different subdomains. The mortar finite element method is a generalization of these methods which allows for nonconforming discretizations on the subdomain interfaces and greater flexibility in the choice of interface approximation spaces. This approach can also be interpreted as a nonstandard variational multiscale method [3] with the subdomain problems serving as the fine scale (h) and the interface problem representing the coarse scale (H). This interpretation allows for *a posteriori* error estimates and adaptive mesh refinement for both the subdomain and mortar scales [42,34]. Furthermore, higher order mortars can be employed, allowing for optimal fine scale convergence with coarser mortar grids. The coarse scale interface problem is usually solved via an iterative method such as the conjugate gradient or GMRES [24] and requires solving subdomain problems in parallel on each iteration. Therefore the method relies on the availability of an efficient and robust preconditioner, e.g.,

* Corresponding author.
E-mail address: twildey@ices.utexas.edu (T. Wildey).

0045-7825/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2010.10.015

Neumann–Neumann [28], balancing [27,14,35], or multigrid [43]. An advantage of the multiscale mortar formulation is that coarser mortar grids imply fewer interface iterations resulting in fewer number of subdomain solves.

Recently, an alternative algorithm, based on the multiscale interpretation, has been developed for the mortar multiscale method [21]. In this approach, a set of multiscale flux basis functions are computed for the coarse scale interface problem. As a result the subdomain solves on each interface iteration are replaced by simple linear combinations of the multiscale flux basis functions. The cost of computing a multiscale basis function is solving a subdomain problem. The number of multiscale flux basis functions per subdomain is equal to the number of coarse scale mortar degrees of freedom for this subdomain. In [21], this was shown to outperform the original algorithm with the balancing preconditioner in many cases. The multiscale basis approach has been applied to stochastic problems in [22]. There, a multiscale flux basis is computed for each stochastic realization, which may lead to substantial computational cost if many realizations are needed.

In this paper, we develop a multiscale preconditioner for the mortar mixed finite element method of [3] that alleviates the need to recompute the multiscale basis for each realization, when the method is combined with a nonintrusive stochastic sampling method. A multiscale mortar flux basis is computed for a single permeability field, called a *training permeability*, that captures the main characteristics of the porous media. The resulting coarse scale interface operator is used as a preconditioner for the interface operators in subsequent realizations. While both our method and the method from [21] involve solving a coarse scale interface problem with a Krylov space method, as in [3], there is an important difference between the two approaches. In [21], the multiscale flux basis is used to replace the subdomain solves on each interface iteration by linear combinations of basis functions. In this paper, the multiscale flux basis is used as a preconditioner, which leads to a reduced number of interface iterations, while subdomain solves are still needed at each iteration. The current approach has an advantage when applied in stochastic setting. A direct application of the algorithm of [21], as it is done in [22], requires computing a new multiscale flux basis for each stochastic realization. With the approach in this paper, only one multiscale flux basis based on a training permeability is computed and used as a preconditioner for all realizations. The cost for computing the multiscale basis on a given subdomain is solving subdomain problems for each coarse scale mortar degree of freedom associated with this subdomain. Note that this is independent of the number of global mortar degrees of freedom. This approach is usually significantly less expensive compared to solving subdomain problems on each interface iteration or recomputing the multiscale basis for each realization, as confirmed by the computational experiments presented in Section 7.

The computational efficiency of our algorithm depends also on the cost of applying the preconditioner and the number of interface iterations. We prove that the condition number of the preconditioned system is bounded by a constant independent of the subdomain mesh size, the number of subdomains, and the mortar mesh size. As a result, the number of preconditioned interface iterations is very small. This is confirmed in our computational experiments. The cost of applying the preconditioner is solving an interface problem for the training operator. This is done by an iterative procedure that requires computing the action of the training operator on each iteration. This action is computed via a linear combination of the multiscale basis functions. Since the basis is precomputed, no additional subdomain solves are required for the preconditioner, resulting in a very efficient algorithm.

We should note that the multiscale preconditioner approach developed in this paper is applicable not only to stochastic modeling, but to other problems where multiple problems with perturbed characteristics need to be solved. One example is applying the multiscale mortar method to multiphase flow in porous media. This is a time dependent nonlinear problem with coefficients that change at each time step and each nonlinear iteration. The multiscale preconditioner can be computed for a fixed set of training parameters and applied over a large number of time steps. This work is the subject of a forthcoming publication.

A number of authors have recently considered domain decomposition methods [50,25] and multiscale methods [4,29, 18,46,16] for stochastic partial differential equations. These papers focus on the combination of a deterministic algorithm with a stochastic approximation method for efficient uncertainty quantification for porous media flow. While this paper is amongst the first to combine the mortar multiscale method with a nonintrusive sampling technique (see also [22]), our focus is on reusing of the multiscale basis associated with the training permeability to reduce the computational cost in each of the subsequent realizations. After completing this paper, we became aware of [25] which combines a Schwartz overlapping domain decomposition method and a nonintrusive sampling algorithm with a Krylov recycling to reduce the computational cost in assembling the Krylov basis for each stochastic realization. This paper shares the concept of recycling information for different realizations, but we recycle the coarse scale interface operator rather than a Krylov basis.

The rest of the paper is organized as follows. In Section 2, we introduce the model problem and the Karhunen–Loéve expansion. The nonintrusive stochastic methods are defined in Section 3. In Section 4, we define the mortar mixed finite element method. A multiscale domain decomposition formulation of the method is given in Section 5. The multiscale basis preconditioner is introduced and analyzed in Section 6. In Section 7 numerical results are provided to confirm the theory and illustrate the efficiency of the method for flow in porous media. Finally, Section 8 contains our concluding remarks.

2. Model problem

We begin with the mixed formulation of Darcy flow. Let $D \subset \mathbb{R}^d, d = 2, 3$ be a bounded Lipschitz domain and Ω be a stochastic event space with probability measure P. The Darcy velocity $\mathbf{u}$ and the pressure p satisfy P-almost everywhere in Ω:

$$\begin{cases} \mathbf{u} = -K(\mathbf{x},\omega)\nabla p, & \text{in } D, \\ \nabla\cdot\mathbf{u} = f, & \text{in } D, \\ p = p_b, & \text{on } \partial D. \end{cases} \tag{2.1}$$

For simplicity we assume Dirichlet boundary conditions in the analysis. More general boundary conditions can also be considered via standard techniques. The permeability K is a diagonal tensor with uniformly positive and bounded elements in D. To simplify the presentation, we will assume that K is a scalar function. Since the permeability K is a stochastic function, p and $\mathbf{u}$ are also stochastic.

Throughout this paper the expected value of a random variable $\xi(\omega)$ with probability density function (p.d.f) $\rho(y)$ will be denoted:

$$E[\xi] = \int_\Omega \xi(\omega)dP(\omega) = \int_{\mathbb{R}} y\rho(y)dy.$$

2.1. The Karhunen–Loève (KL) expansion

In order to guarantee positive permeability almost surely in Ω, we consider the log transformed permeability $Y = \ln(K)$. Let the mean removed log permeability be denoted by Y', so that $Y = E[Y] + Y'$. Its covariance function $C_Y(\mathbf{x},\bar{\mathbf{x}}) = E[Y'(\mathbf{x},\omega)Y'(\bar{\mathbf{x}},\omega)]$

is symmetric and positive definite, and hence can be decomposed into the series expansion:

$$C_Y(\mathbf{x},\bar{\mathbf{x}}) = \sum_{i=1}^{\infty} \lambda_i f_i(\mathbf{x}) f_i(\bar{\mathbf{x}}). \tag{2.2}$$

The eigenvalues λ_i and eigenfunctions f_i of this series are computed using C_Y as the kernel of the Type II Fredholm integral equation:

$$\int_D C_Y(\mathbf{x},\bar{\mathbf{x}}) f(\mathbf{x}) d\mathbf{x} = \lambda f(\bar{\mathbf{x}}). \tag{2.3}$$

The Karhunen–Loève expansion of the log permeability can be written as

$$Y(\mathbf{x},\omega) = E[Y](\mathbf{x}) + \sum_{i=1}^{\infty} \xi_i(\omega)\sqrt{\lambda_i} f_i(\mathbf{x}), \tag{2.4}$$

where ξ_i are mutually uncorrelated random variables with zero mean and unit variance [23]. In the case where Y' is given by a Gaussian process, the ξ_i are independent.

At this point, the KL expansion is truncated after N terms, which is reasonable to do as typically the λ_i decay rapidly [49]. If the expansion is truncated prematurely, the permeability may appear too smooth. Increasing N introduces more heterogeneity into the permeability realizations. This truncation allows us to write $Y(\mathbf{x},\omega) = Y(\mathbf{x},\xi_1(\omega),\ldots,\xi_N(\omega))$. The images of the random variables $\xi_i(\Omega)$ make up a finite dimensional set $\mathbb{S}_N = \prod_{i=1}^N \xi_i(\Omega) \subset \mathbb{R}^N$. In the case of Gaussian random variables, $\mathbb{S}_N = \mathbb{R}^N$. If ρ_i corresponds to the p.d.f. of each ξ_i, then the joint p.d.f. for the random vector $(\xi_1,\ldots,\xi_N)$ is defined to be $\rho = \prod_{i=1}^N \rho_i$. Then we can write $Y(\mathbf{x},\omega) = Y(\mathbf{x},\mathbf{y})$, where $\mathbf{y} = (y_1,\ldots,y_N)$ and $y_i = \xi_i(\omega)$.

We use the following specific covariance function (in 2-D) originally taken from [49], in which λ_i and $f_i(\mathbf{x})$ can be found analytically:

$$C_Y(\mathbf{x},\bar{\mathbf{x}}) = \sigma_Y^2 \exp\left[-\frac{|x_1-\bar{x}_1|}{\eta_1} - \frac{|x_2-\bar{x}_2|}{\eta_2}\right]. \tag{2.5}$$

Here σ_Y and η_i denote variance and correlation length in the ith spatial dimension, respectively. These eigenvalues will decay at a rate asymptotic to $O(1/N^2)$ and for this particular case can be computed analytically.

When the exact eigenvalues and eigenfunctions of the covariance function C_Y can be found, the KL expansion is the most efficient method for representing a random field. However, in most cases, closed-form eigenfunctions and eigenvalues are not readily available and numerical procedures need be performed for solving the integral Eq. (2.3). Efficient methods for numerically computing the KL expansion are reported in [38].

2.2. Variational formulation

Appealing to the Doob–Dynkin Lemma [33], the p.d.f. for the permeability K carries through to the solution of Eq. (2.1), so that $(\mathbf{u}p)$ has the form:

$$\mathbf{u}(\mathbf{x},\omega) = \mathbf{u}(\mathbf{x},\xi_1(\omega),\ldots,\xi_N(\omega)) = \mathbf{u}(\mathbf{x},y_1,\ldots,y_N) \quad \text{and}$$
$$p(\mathbf{x},\omega) = p(\mathbf{x},\xi_1(\omega),\ldots,\xi_N(\omega)) = p(\mathbf{x},y_1,\ldots,y_N).$$

Let D be decomposed into nonoverlapping subdomain blocks D_i, so that $\overline{D} = \cup_{i=1}^n \overline{D_i}$, and $D_i \cap D_j = \emptyset$ for $i \neq j$. The blocks need not share complete faces, i.e., they need not form a conforming partition. Let $\Gamma_{ij} = \partial D_i \cap \partial D_j$, $\Gamma = \cup_{1\leqslant i<j\leqslant n} \Gamma_{ij}$, and $\Gamma_i = \partial D_i \cap \Gamma = \partial D_i \setminus \partial D$ denote interior block interfaces. Let:

$$\mathbf{V}_i = H(\mathrm{div}; D_i), \quad \mathbf{V} = \bigoplus_{i=1}^n \mathbf{V}_i,$$
$$W_i = L^2(D_i), \quad W = \bigoplus_{i=1}^n W_i = L^2(D),$$
$$M = \left\{\mu \in H^{1/2}(\Gamma) : \mu|_{\Gamma_i} \in (\mathbf{V}_i \cdot \nu_i)^*, \quad i = 1,\ldots,n\right\},$$

where ν_i is the outer unit normal to ∂D_i and $(\cdot)^*$ denotes the dual space.

Let $(\cdot,\cdot)_S$, $S \subset \mathbb{R}^d$, denote the $L^2(S)$ inner product, and let $\langle\cdot,\cdot\rangle_G$, $G \subset \mathbb{R}^{d-1}$, denote the $L^2(G)$ inner product or duality pairing. Following: [2,19,22], a weak form of (2.1) seeks $\mathbf{u}(\mathbf{x},\omega) \in \mathbf{V} \otimes L^2(\mathbb{S}_N)$, $p(\mathbf{x},\omega) \in W \otimes L^2(\mathbb{S}_N)$ and $\lambda(\mathbf{x},\omega) \in M \otimes L^2(\mathbb{S}_N)$ such that, for each i:

$$\int_{\mathbb{S}_N} \left(K^{-1}\mathbf{u},\mathbf{v}\right)_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}_N} \Big((p,\nabla\cdot\mathbf{v})_{D_i} - \langle\lambda,\mathbf{v}\cdot\nu_i\rangle_{\Gamma_i} - \langle g,\mathbf{v}\cdot\nu_i\rangle_{\partial D_i\setminus\Gamma}\Big)\rho(\mathbf{y}) d\mathbf{y}, \tag{2.6}$$

$$\int_{\mathbb{S}_N} (\nabla\cdot\mathbf{u},w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}_N} (f,w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}, \tag{2.7}$$

$$\sum_{i=1}^n \int_{\mathbb{S}_N} \langle \mathbf{u}\cdot\nu_i, \mu\rangle_{\Gamma_i} \rho(\mathbf{y}) d\mathbf{y} = 0, \tag{2.8}$$

for all $\mathbf{v} \in \mathbf{V}_i \otimes L^2(\mathbb{S}_N)$, $w \in W_i \otimes L^2(\mathbb{S}_N)$, and $\mu \in M \otimes L^2(\mathbb{S}_N)$. Note that λ is the pressure on the block interfaces Γ and that (2.8) enforces weak continuity of $\mathbf{u}\cdot\nu$ on each Γ_{ij}.

3. Nonintrusive stochastic methods

Given the log permeability as a truncated KL expansion, the problem is now formulated in the finite dimensional space $D \otimes \mathbb{S}_N \subset \mathbb{R}^{d+N}$. At this point, there are several ways in which to discretize the problem. The Stochastic Finite Element Method (SFEM) [15] considers solving the problem using full $d+N$ dimensional finite elements. The resulting system is significantly large, may be difficult to set up, and the solution algorithm does not easily lend itself to parallelization.

A less intrusive approach is to use d-dimensional finite elements in the spatial domain D, and to sample the stochastic space $\mathbb{S}_N$ only at certain points. The Monte Carlo method is the most popular of these sampling techniques. The advantage of this approach is that the resulting deterministic FEM problems are completely uncoupled, and may be solved in parallel. The disadvantage of the Monte Carlo method is that the convergence rate with respect to the stochastic space is slow, $O\left(1/\sqrt{\widetilde{M}}\right)$, where $\widetilde{M}$ is the number of realizations.

The Stochastic Collocation Method is another nonintrusive approach that improves upon the Monte Carlo method by sampling at specially chosen collocation points in order to form a polynomial interpolant in the stochastic space. Different varieties of stochastic collocation arise by considering different sets of collocation points. The simplest approach is a full tensor product grid of collocation points. For relatively small stochastic dimension N, the tensor product stochastic collocation method converges much faster than Monte Carlo when combined with finite element spatial approximation [19,5,45,44]. In this paper, we consider only the nonintrusive stochastic methods thereby reducing the stochastic problem to a system of uncoupled deterministic problems.

It should be noted that full tensor product grids of collocation points suffer from the so-called "curse of dimensionality". Increasing the number of terms in the truncated KL expansion Eq. (2.4) increases the number of stochastic dimensions, which exponentially increases the number of points in a full tensor product grid. For example, if k collocation points are used in each stochastic dimension, $\widetilde{M} = k^N$. To cope with this problem, more advanced collocation techniques are possible such as the so called probabilistic collocation method (see e.g., [26]) and the Smolyak sparse grids (see e.g., [44,32,31]). These advanced collocation techniques will not be considered in this paper, but it is straightforward to extend the multiscale preconditioning strategy defined in Section 6 to these methods.

In both the Monte Carlo and the stochastic collocation methods, the goal is to find, for $1 \leqslant m \leqslant \widetilde{M}$, $\mathbf{u}^{\{m\}} \in \mathbf{V}$, $p^{\{m\}} \in W$ and $\lambda^{\{m\}} \in M$ such that for each i:

$$\begin{cases} \left((K^{\{m\}})^{-1}\mathbf{u}^{\{m\}}, \mathbf{v}\right)_{D_i} = (p^{\{m\}}, \nabla\cdot\mathbf{v})_{D_i} \\ \quad -\langle \lambda^{\{m\}}, \mathbf{v}\cdot\nu_i\rangle_{D_i} - \langle g, \mathbf{v}\cdot\nu_i\rangle_{\partial D_i\setminus\Gamma}, \\ \left(\nabla\cdot\mathbf{u}^{\{m\}}, w\right)_{D_i} = (f,w)_{D_i}, \\ \sum_{i=1}^{n}\langle \mathbf{u}^{\{m\}}\cdot\nu_i, \mu\rangle_{\Gamma_i} = 0, \end{cases} \tag{3.1}$$

for all $\mathbf{v} \in \mathbf{V}_i$, $w \in W_i$, and $\mu \in M$, where $\{\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_{\widetilde{M}}\}$ are the chosen sample points in $\mathbb{S}_N$, and $K^{\{m\}} = K(\mathbf{x}, \mathbf{y}_m)$. We assume that for each permeability realization there exist positive constants $\hat{c}^{\{m\}}, \widehat{C}^{\{m\}}$, and α_i such that:

$$\hat{c}^{\{m\}}\alpha_i\xi^T\xi \leqslant \xi^T K^{\{m\}}(\mathbf{x})\xi \leqslant \widehat{C}^{\{m\}}\alpha_i\xi^T\xi, \quad \forall \xi \in \mathbb{R}^d, \quad \forall \mathbf{x} \in D_i, \quad i = 1, \ldots, n. \tag{3.2}$$

4. The finite element approximation

Let $\mathcal{T}_{h,i}$ be a conforming quasi-uniform affine finite element partition of D_i, $1 \leqslant i \leqslant n$, of maximal element diameter h_i. Note that we need quasi-uniformity and conformity only on each subdomain. Our method allows for spatially varying h_i, but to simplify the discussion, we let $h = \max_{1\leqslant i\leqslant n} h_i$ and analyze the method in terms of this single value h. We allow for the possibility that $\mathcal{T}_{h,i}$ and $\mathcal{T}_{h,j}$ do not align on $\Gamma_{i,j}$. Define $\mathcal{T}_h = \cup_{i=1}^n \mathcal{T}_{h,i}$, and let $\mathcal{E}_h$ be the union of all interior edges (or faces in three dimensions) not including the subdomain interfaces and the outer boundaries. Let:

$$\mathbf{V}_{h,i} \times W_{h,i} \subset \mathbf{V}_i \times W_i,$$

be any of the usual mixed finite element spaces (e.g., those of [9,10,30,11,37]), and let $\mathbf{V}_h$ or, equivalently, $\mathbf{V}_h \cdot \nu$ contain the polynomials of degree k. Then let:

$$\mathbf{V}_h = \bigoplus_{i=1}^{n} \mathbf{V}_{h,i}, \quad W_h = \bigoplus_{i=1}^{n} W_{h,i}.$$

Note that the normal components of vectors in $\mathbf{V}_h$ are continuous between elements within each block D_i, but not across Γ.

Let the mortar interface mesh $\mathcal{T}_{H,i,j}$ be a quasi-uniform finite element partition of $\Gamma_{i,j}$ with maximal element diameter $H_{i,j}$. Let $H = \max_{1\leqslant i,j\leqslant n} H_{i,j}$. In multiscale approximations one takes $H > h$. Define $\mathcal{T}^{\Gamma,H} = \cup_{1\leqslant i<j\leqslant n}\mathcal{T}_{H,i,j}$. For any $\tau \in \mathcal{T}_{H,i,j}$, let:

$$E_\tau = \{E \in \mathcal{T} : \partial E \cap \tau \neq \emptyset\}.$$

Denote by $M_{H,i,j} \subset L^2(\Gamma_{i,j})$ the mortar space on $\Gamma_{i,j}$ containing either the continuous or discontinuous piecewise polynomials of degree q on $\mathcal{T}_{H,i,j}$, where q is at least $k+1$. We remark that $\mathcal{T}_{H,i,j}$ need not be conforming if $M_{H,i,j}$ is discontinuous. Now let:

$$M_H = \bigoplus_{1\leqslant i<j\leqslant n} M_{H,i,j},$$

be the mortar finite element space on Γ. For each subdomain D_i, define a projection $\mathcal{Q}_{h,i} : L^2(\Gamma_i) \to \mathbf{V}_{h,i}\cdot\nu_i|_{\Gamma_i}$ such that, for any $\phi \in L^2(\Gamma_i)$:

$$\langle \phi - \mathcal{Q}_{h,i}\phi, \mathbf{v}\cdot\nu_i\rangle_{\Gamma_i} = 0, \quad \mathbf{v} \in \mathbf{V}_{h,i}. \tag{4.1}$$

We require that the following condition be satisfied [2], where in this paper $\|\cdot\|_{r,S}$ is the usual Sobolev norm of $H^r(S)$ (we may drop r if $r = 0$ and S if $S = D$).

Assumption 4.1. Assume that there exists a constant C, independent of h and H, such that:

$$\|\mu\|_{\Gamma_{i,j}} \leqslant C\left(\|\mathcal{Q}_{h,i}\mu\|_{\Gamma_{i,j}} + \|\mathcal{Q}_{h,j}\mu\|_{\Gamma_{i,j}}\right), \quad \mu \in M_H, \quad 1 \leqslant i < j \leqslant n. \tag{4.2}$$

Condition (4.2) says that the mortar space cannot be too rich compared to the normal traces of the subdomain velocity spaces. Therefore, in what follows, we tacitly assume that $h \leqslant H \leqslant 1$. Condition (4.2) is not very restrictive, and is easily satisfied in practice (see, e.g., [35,47]). In the following, we treat any function $\mu \in M_H$ as extended by zero on ∂D.

In the mixed finite element approximation of (3.1), we seek, for $1 \leqslant m \leqslant \widetilde{M}$, $\mathbf{u}_h^{\{m\}} \in \mathbf{V}_h$, $p_h^{\{m\}} \in W_h$ and $\lambda_H^{\{m\}} \in M_H$ such that, for $1 \leqslant i \leqslant n$:

$$\begin{cases} \left(\left(K^{\{m\}}\right)^{-1}\mathbf{u}_h^{\{m\}}, \mathbf{v}\right)_{D_i} = \left(p_h^{\{m\}}, \nabla\cdot\mathbf{v}\right)_{D_i} \\ \quad -\left\langle \lambda_H^{\{m\}}, \mathbf{v}\cdot\nu_i\right\rangle_{\Gamma_i} - \langle g, \mathbf{v}\cdot\nu_i\rangle_{\partial D_i\setminus\Gamma}, \\ \left(\nabla\cdot\mathbf{u}_h^{\{m\}}, w\right)_{D_i} = (f,w)_{D_i}, \\ \sum_{i=1}^{n}\left\langle \mathbf{u}_h^{\{m\}}\cdot\nu_i, \mu\right\rangle_{\Gamma_i} = 0, \end{cases} \tag{4.3}$$

for all $\mathbf{v} \in \mathbf{V}_{h,i}$, $w \in W_{h,i}$, and $\mu \in M_H$. Strictly within each block D_i, we have a standard mixed finite element method with local conservation over each grid cell. Moreover, $\mathbf{u}_h^{\{m\}}\cdot\nu$ is continuous on any element edge (or face) with weak continuity of the flux across the interfaces with respect to the mortar space M_H.

Remark 4.2. As observed in [3,21], the mortar mixed finite element method may be viewed as a nonstandard multiscale method if a coarse mortar space is chosen. In this case, optimal order error estimates can be obtained by choosing high-order polynomials for the mortars. We refer the reader to [3,34] for more details.

5. A multiscale domain decomposition formulation

Using a substructuring domain decomposition algorithm introduced in [24], the linear system resulting from the mortar mixed finite element method (4.3) on each realization can be reduced to a coarse scale interface problem in the mortar space M_H [47,2,3]. The interface problem can be solved efficiently in parallel via a Krylov space iterative method, with each iteration requiring the solution of subdomain problems.

For each $1 \leqslant m \leqslant \widetilde{M}$, define a bilinear form $d_H^{\{m\}} : L^2(\Gamma)\times L^2(\Gamma) \to \mathbb{R}$ by

$$d_H^{\{m\}}(\lambda,\mu) = \sum_{i=1}^{n} d_{H,i}^{\{m\}}(\lambda,\mu) = -\sum_{i=1}^{n}\left\langle \mathbf{u}_h^{*,\{m\}}(\lambda)\cdot\nu_i, \mu\right\rangle_{\Gamma_i},$$

where the pair $\left(\mathbf{u}_h^{*,\{m\}}(\lambda), p_h^{*,\{m\}}(\lambda)\right) \in \mathbf{V}_h \times W_h$ is computed by solving:

$$\begin{cases} \left((K^{\{m\}})^{-1}\mathbf{u}_h^{*,\{m\}}(\lambda), \mathbf{v}\right)_{D_i} = \left(p_h^{*,\{m\}}(\lambda), \nabla\cdot\mathbf{v}\right)_{D_i} \\ \qquad -\langle \lambda, \mathbf{v}\cdot\nu_i\rangle_{\Gamma_i}, \qquad \mathbf{v} \in \mathbf{V}_{h,i}, \\ \left(\nabla\cdot\mathbf{u}_h^{*,\{m\}}(\lambda), w\right)_{D_i} = 0, \qquad w \in W_{h,i}, \end{cases} \tag{5.1}$$

for each $1 \leqslant i \leqslant n$ for a given $\lambda \in L^2(\Gamma)$. Also define a linear functional $g_H^{\{m\}} : L^2(\Gamma) \to \mathbb{R}$ by

$$g_H^{\{m\}}(\mu) = \sum_{i=1}^{n} g_{h,i}^{\{m\}}(\mu) = \sum_{i=1}^{n}\left\langle \bar{\mathbf{u}}_h^{\{m\}}\cdot\nu_i, \mu\right\rangle_{\Gamma_i},$$

where the pair $\left(\bar{\mathbf{u}}_h^{\{m\}}, \bar{p}_h^{\{m\}}\right) \in \mathbf{V}_h \times W_h$ solves, for $1 \leqslant i \leqslant n$:

$$\begin{cases} \left((K^{\{m\}})^{-1}\bar{\mathbf{u}}_h^{\{m\}},\mathbf{v}\right)_{D_i} = \left(\bar{p}_h^{\{m\}},\nabla\cdot\mathbf{v}\right)_{D_i} - \langle g,\mathbf{v}\cdot\nu_i\rangle_{\partial D_i\setminus\Gamma}, & \mathbf{v}\in\mathbf{V}_{h,i},\\ \left(\nabla\cdot\bar{\mathbf{u}}_h^{\{m\}},w\right)_{D_i} = (f,w)_{D_i}, & w\in W_{h,i}. \end{cases} \tag{5.2}$$

This gives, for each $1\leqslant m\leqslant\widetilde{M}$, the coarse scale interface problem:

$$d_H^{\{m\}}(\lambda_H^{\{m\}},\mu) = g_H^{\{m\}}(\mu),\quad \mu\in M_H. \tag{5.3}$$

We remark that solving (5.2) corresponds to solving subdomain problems with zero Dirichlet conditions along the mortar boundaries, and $g_H^{\{m\}}$ represents the corresponding jump in the flux projected into the mortar space. Solving the coarse scale interface problem (5.3) amounts to finding the proper mortar values to balance the jump in the flux.

It is straightforward to show (see [24,2]) that the multiscale solution to (4.3) can be reconstructed from the solution $\lambda_H^{\{m\}}$ to the interface problem Eq. (5.3) via:

$$\mathbf{u}_h^{\{m\}} = \mathbf{u}_h^{*,\{m\}}\left(\lambda_H^{\{m\}}\right) + \bar{\mathbf{u}}_h^{\{m\}},\quad p_h^{\{m\}} = p_h^{*,\{m\}}\left(\lambda_H^{\{m\}}\right) + \bar{p}_h^{\{m\}}. \tag{5.4}$$

The following result concerning the existence and uniqueness of the coarse scale solution has been shown in [24,2].

Lemma 5.1. *For $1\leqslant m\leqslant\widetilde{M}$, the interface bilinear form $d_H^{\{m\}}(\cdot,\cdot)$ is symmetric and positive definite on $L^2(D)$. If (4.2) holds, then $d_H^{\{m\}}(\cdot,\cdot)$ is positive definite on M_H. Moreover:*

$$d_{H,i}^{\{m\}}(\mu,\mu) = \left((K^{\{m\}})^{-1}\mathbf{u}_h^{*,\{m\}}(\mu),\quad \mathbf{u}_h^{*,\{m\}}(\mu)\right)_{D_i} \geqslant 0. \tag{5.5}$$

We will find it useful to consider the linear algebraic version of the discrete interface operator. For each $1\leqslant m\leqslant\widetilde{M}$ we define a real $N_H\times N_H$ matrix $\mathcal{D}_H^{\{m\}}$ satisfying:

$$\left[\mathcal{D}_H^{\{m\}}\boldsymbol{\lambda},\boldsymbol{\mu}\right] := d_H^{\{m\}}(\lambda,\mu)\quad\forall\lambda,\quad \mu\in M_H, \tag{5.6}$$

where N_H denote the number of degrees of freedom associated with M_H, and $[\cdot,\cdot]$ is the Euclidean scalar product in $\mathbb{R}^{N_H}$. For each $\mu\in M_H$, $\boldsymbol{\mu}$ denotes the vector of its values at the N_H nodes. We note that $\mathcal{D}_H^{\{m\}} = \sum_{i=1}^n \mathcal{D}_{H,i}^{\{m\}}$, where $\mathcal{D}_{H,i}^{\{m\}}$ satisfy:

$$\left[\mathcal{D}_{H,i}^{\{m\}}\boldsymbol{\lambda},\boldsymbol{\mu}\right] = d_{H,i}^{\{m\}}(\lambda,\mu)\quad\forall\lambda,\quad \mu\in M_H.$$

The interface problem Eq. (5.3) can be written as

$$\mathcal{D}_H^{\{m\}}\boldsymbol{\lambda}_H^{\{m\}} = \mathbf{g}_H^{\{m\}}, \tag{5.7}$$

where the operator $\mathcal{D}_H^{\{m\}}$ is the discrete mortar version of the Steklov–Poincaré operator [36].

It is shown in [47], see also [14,35], that there exist positive constants $c^{\{m\}}$ and $C^{\{m\}}$ such that:

$$c^{\{m\}}\sum_{i=1}^n \alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i} \leqslant \left[\mathcal{D}_H^{\{m\}}\boldsymbol{\mu},\boldsymbol{\mu}\right] \leqslant C^{\{m\}}\sum_{i=1}^n \alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i}, \tag{5.8}$$

for all $\mu\in M_H$, where $\mathcal{I}^{\partial D_i}$ is a continuous piecewise linear interpolant defined in [14,47,35]. The constants $C^{\{m\}}$ and $c^{\{m\}}$ do not depend on h or H. They depend only mildly on $K^{\{m\}}$, since the dependence on the characteristic values α_i is given explicitly.

An immediate consequence of Lemma 5.1 is that the conjugate gradient method can be employed to solve the interface problem Eq. (5.3). On each conjugate gradient iteration, the bilinear form $d_H^{\{m\}}(\cdot,\cdot)$ needs to be evaluated, which requires computing $\mathbf{u}_h^{*,\{m\}}(\lambda)$.

In the original implementation of the mortar mixed finite element method [47,2,3], the action of the interface operator on each conjugate gradient iteration is computed by solving subdomain problems Eq. (5.1) to compute $\mathbf{u}_h^{*,\{m\}}(\lambda)$. In [3], the mortar mixed finite element method is shown to be equivalent to a nonstandard multiscale method with the subdomain problems discretized on the fine scale and the mortar interface problem discretized on the coarse scale. This relationship is exploited in [21], where a multiscale mortar flux basis is precomputed by solving (5.1) for each mortar basis function. The multiscale basis functions are constructed by projecting the corresponding boundary fluxes into the mortar space. Then, for each iteration, the action of the interface operator is computed by taking a linear combination of the multiscale basis functions.

More precisely, following [21], we let $\left\{\phi_{H,i}^{\{k\}}\right\}_{k=1}^{N_{H,i}}$ denote a basis for the mortar space $M_{H,i}$ where $N_{H,i}$ is the number of degrees of freedom associated with $M_{H,i}$. A multiscale mortar flux basis, $\left\{\psi_{H,i}^{\{k\}}\right\}_{k=1}^{N_{H,i}}$, is computed by calculating the flux response for each mortar basis function. This process is summarized in Algorithm 1 where $\mathcal{Q}_{h,i}^T:\mathbf{V}_{h,i}\cdot\nu_i\to M_{H,i}$ denotes the L^2-orthogonal projection from the normal trace of the velocity space into the mortar space. In essence, Algorithm 1 computes:

$$\psi_{H,i}^{\{m\},\{k\}} = \mathcal{D}_{H,i}^{\{m\}}\phi_{H,i}^{\{k\}}.$$

Algorithm 1: Construction of the multiscale mortar flux basis

for $i = 1,2,\ldots,n$ **do**
  for $k = 1,2,\ldots,N_{H,i}$ **do**
    (a) Project $\phi_{H,i}^{\{k\}}$ onto the subdomain boundary:

$$\mathcal{Q}_{h,i}\phi_{H,i}^{\{k\}} = \gamma_i^{\{k\}}.$$

    (b) Solve the subdomain problems (5.1) with $\lambda = \gamma_i^{\{k\}}$.
    (c) Project the flux into the mortar space:

$$\psi_{H,i}^{\{m\},\{k\}} = -\mathcal{Q}_{h,i}^T\mathbf{u}_h^{*,\{m\}}\left(\gamma_i^{\{k\}}\right)\cdot\nu_i.$$

    (d) Store the multiscale basis function $\psi_{H,i}^{\{m\},\{k\}}$.
  end for
end for

Note that the number of solves (5.1) per subdomain depends only on the number of mortar degrees of freedom associated with the particular subdomain and may differ throughout the computational domain.

Once the multiscale basis is computed, the computation of the action of the interface operator on every interface iteration is reduced to a linear combination of the basis functions. More precisely, if $\lambda_{H,i} = \sum_{k=1}^{N_{H,i}}\beta_{k,i}\phi_{H,i}^{\{k\}}$:

$$\mathcal{D}_{H,i}^{\{m\}}\lambda_{H,i} = \mathcal{D}_{H,i}^{\{m\}}\sum_{k=1}^{N_{H,i}}\beta_{k,i}\phi_{H,i}^{\{k\}} = \sum_{k=1}^{N_{H,i}}\beta_{k,i}\mathcal{D}_{H,i}^{\{m\}}\phi_{H,i}^{\{k\}} = \sum_{k=1}^{N_{H,i}}\beta_{k,i}\psi_{H,i}^{\{m\},\{k\}},$$

where, by abuse of notation, $\mathcal{D}_{H,i}^{\{m\}}$ is the portion of the matrix, $\mathcal{D}_H^{\{m\}}$, corresponding to $M_{H,i}$. In [21], this approach is applied to deterministic problems and it is shown to reduce the total number of subdomain solves in the case of a large number of subdomains with relatively few mortar degrees of freedom on each interface, or if the permeability is highly heterogeneous.

Remark 5.2. This approach is closely related (but more general) than the substructuring methods developed in e.g., [8,7,1]. If the basis functions for $M_{H,i}$ are chosen to be the Lagrange basis, then this procedure is similar to the construction of the local

contribution to the Schur complement [39,36]. Moreover the latter approaches treat conforming Galerkin finite element methods and not mixed methods.

6. A multiscale preconditioner

In [22], the multiscale basis method is combined with stochastic collocation to model flow in non-stationary random porous media. In that implementation, Algorithm 1 is repeated for each $1 \leqslant m \leqslant \widetilde{M}$. This requires recomputing the multiscale flux basis for each realization. To avoid this extra expense we implement Algorithm 1 for a carefully chosen *training permeability* and use the corresponding interface operator as a preconditioner for each of the subsequent realizations.

Let $\overline{K}$ denote a uniformly positive and bounded *training permeability*. Let $\overline{\mathcal{D}}_H$ be the interface operator defined in Eq. (5.6), corresponding to the global problem with permeability $\overline{K}$.

For each $1 \leqslant m \leqslant \widetilde{M}$, we solve the preconditioned system:

$$\overline{\mathcal{D}}_H^{-1}\mathcal{D}_H^{\{m\}}\boldsymbol{\lambda}_H^{\{m\}} = \overline{\mathcal{D}}_H^{-1}\boldsymbol{g}_H^{\{m\}}, \tag{6.1}$$

using an iterative algorithm such as conjugate gradient or GMRES. Note that in this approach, the action of $\mathcal{D}_H^{\{m\}}$ is computed by solving subdomain problems (5.1) with the true permeability $K^{\{m\}}$, thus avoiding the need to compute a multiscale basis for each realization.

The preconditioner $\overline{\mathcal{D}}_H^{-1}$ is applied using a multiscale flux basis. In particular, we employ Algorithm 1 with subdomain problems Eq. (5.1) with permeability $\overline{K}$ to compute a multiscale flux basis $\overline{\psi}_{H,i}^{\{k\}}$. Note that only one multiscale flux basis is computed.

The action $\overline{\mathcal{D}}_H^{-1}\boldsymbol{\mu}$ is computed by solving an interface problem $\overline{\mathcal{D}}_H\boldsymbol{\lambda} = \boldsymbol{\mu}$, for which we use an iterative algorithm such as conjugate gradient. On each conjugate gradient iteration, the action of the interface operator $\overline{\mathcal{D}}_H$ is computed using the linear combination of multiscale basis functions:

$$\overline{\mathcal{D}}_{H,i}\lambda_i = \sum_{k=1}^{N_{H,i}} \beta_k \overline{\psi}_{H,i}^{\{k\}},$$

thus avoiding the need to solve any additional subdomain problems. As a result, the cost of applying the multiscale preconditioner $\overline{\mathcal{D}}_H^{-1}$ for each stochastic realization is relatively small.

The computational efficiency of our algorithm also depends on the number of iterations for the preconditioned coarse scale interface system Eq. (6.1). To address this issue, we present below a theoretical bound on the condition number.

The following result on the preconditioning of symmetric positive definite operators can be found in [36, Theorem 4.1.5.]

Theorem 6.1. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two symmetric and positive definite $N \times N$ real matrices. Assume that there exists constant $C_1 > 0$ and $C_2 > 0$ such that:*

$$C_1[\mathcal{D}_1\boldsymbol{\mu}, \boldsymbol{\mu}] \leqslant [\mathcal{D}_2\boldsymbol{\mu}, \boldsymbol{\mu}] \leqslant C_2[\mathcal{D}_1\boldsymbol{\mu}, \boldsymbol{\mu}],$$

for all $\boldsymbol{\mu} \in \mathbb{R}^N$ where $[\cdot,\cdot]$ is the Euclidean scalar product in $\mathbb{R}^N$. Then the eigenvalues of the preconditioned matrix $\mathcal{D}_1^{-1}\mathcal{D}_2$ satisfy:

$$C_1 \leqslant \nu_{\min} \leqslant \nu_{\max} \leqslant C_2,$$

and the spectral condition number $\operatorname{cond}(\mathcal{D}_1^{-1}\mathcal{D}_2) := \frac{\nu_{\max}}{\nu_{\min}}$ *satisfies:*

$$\operatorname{cond}(\mathcal{D}_1^{-1}\mathcal{D}_2) \leqslant \frac{C_2}{C_1}.$$

We are ready to present the main theoretical result in this paper.

Theorem 6.2. *Assume that (4.2) holds and that the training operator $\overline{\mathcal{D}}_H$ satisfies a bound similar to (5.8),*

$$\overline{c}\sum_{i=1}^{n}\alpha_i\left|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu\right|_{1/2,\partial D_i} \leqslant \left[\overline{\mathcal{D}}_H\boldsymbol{\mu},\boldsymbol{\mu}\right] \leqslant \overline{C}\sum_{i=1}^{n}\alpha_i\left|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu\right|_{1/2,\partial D_i} \tag{6.2}$$

for all $\mu \in M_H$, where $\overline{C}$ and $\overline{c}$ are positive continuity and coercivity constants. Then, for $1 \leqslant m \leqslant M$:

$$\operatorname{cond}\left(\overline{\mathcal{D}}_H^{-1}\mathcal{D}_H^{\{m\}}\right) \leqslant \frac{\overline{C}C^{\{m\}}}{\overline{c}C^{\{m\}}}, \tag{6.3}$$

i.e., $\overline{\mathcal{D}}_H$ and $\mathcal{D}_H^{\{m\}}$ are uniformly spectrally equivalent.

Proof. From (5.8) and (6.2) we easily derive:

$$\frac{c^{\{m\}}}{\overline{C}}\left[\overline{\mathcal{D}}_H\boldsymbol{\mu},\boldsymbol{\mu}\right] \leqslant \left[\mathcal{D}_H^{\{m\}}\boldsymbol{\mu},\boldsymbol{\mu}\right] \leqslant \frac{C^{\{m\}}}{\overline{c}}\left[\overline{\mathcal{D}}_H\boldsymbol{\mu},\boldsymbol{\mu}\right].$$

The result (6.3) follows from Theorem 6.1. □

Remark 6.3. Although the condition number (6.3) does not depend on h or H, it does depend through (6.2) and (5.8) on how closely the *training permeability*, $\overline{K}$, represents the permeability for each of the realizations, $K^{\{m\}}$. Thus, $\overline{K}$ should be chosen based on the physical or the stochastic properties of the permeability in (2.1).

We conclude this section with a comment on the scalability of this approach. Although the multiscale basis preconditioner bounds the number of subdomain solves for each realization, the cost of applying the preconditioner may grow as the number of subdomains increases or as the subdomain mesh size decreases. The dominant cost of the preconditioner, computing the multiscale basis, is proportional to the number of mortar degrees of freedom per subdomain and thus it does not grow with the number of subdomains. On the other hand, the number of interface iterations for applying the action of the preconditioner may increase with the number of subdomains or when the subdomain grids are refined. Even though the local computations are very inexpensive (linear combinations of the multiscale basis), there is some communication overhead that may affect the scalability of the algorithm. In a related paper [20], we address the theoretical complexity in solving the interface problem for the preconditioner $\overline{\mathcal{D}}_H^{-1}$ using the multiscale basis and investigate the use of preconditioners for this iteration, such as balancing or block Jacobi with a coarse scale correction, and multilevel acceleration methods, such as interface multigrid, to efficiently solve the interface problem. For the numerical results in Section 7, the wall clock time to apply the multiscale basis preconditioner is always substantially less than the time to perform one solve per subdomain.

7. Numerical results

In this section, we present numerical results supporting the theoretical results in Section 6. All results are computed in 2D using the lowest order Raviart–Thomas RT_0 spaces [37,12] on uniform rectangular subdomain grids that do not necessarily match on the interface.

First, we use a deterministic problem to demonstrate that the condition number of the preconditioned system is independent of the subdomain mesh size h, the mortar mesh size H, and the degree of the mortar approximation q. Then we show that the choice of the *training permeability* has a significant impact on the condition number of the preconditioned system. Finally, we apply the multiscale basis preconditioner to several problems with a truncated Karhunen–Loéve expansion of a stochastic permeability field using a nonintrusive stochastic collocation approximation.

7.1. Optimality of the preconditioner

Define $D = [0,1] \times [0,1]$ and consider problem Eq. (2.1) with a deterministic coefficient $K(\mathbf{x}) = 1 - 0.5 \sin(\pi x_1) \sin(\pi x_2)$, $f(\mathbf{x}) = 1$, and $p_b = 0$. We compute the multiscale basis functions based associated with a relatively simple *training permeability*: $\overline{K}(\mathbf{x}) = 1$.

First, we divide D into 16 equal sized subdomains in a 4×4 pattern and choose continuous piecewise linear mortars with $H = 2h$. In this, and the other two cases in this subsection, the outer iterative method is GMRES (unrestarted) with a tolerance of 10^{-8} and the preconditioning system is solved using CG with a tolerance of 10^{-10}. In Table 1, we see that refining the subdomain mesh size has no effect on the number of interface iterations.

Next, we set $h = 1/128$ and use continuous piecewise linear mortars with $H = 2h$. We vary the number of subdomains from 4 (2×2) to 256 (16×16). In Table 2, we observe that increasing the number of subdomains does not affect the number of interface iterations.

Finally, we set $h = 1/128$ and fix the number of subdomains at 16 (4×4). We choose $H = 4h$ and vary the degree of the mortar approximation from one to three and investigate both continuous and discontinuous mortars. In Table 3, we see that the mortar degree does not affect the number of interface iterations.

7.2. Choice of training permeability

Define $D = [0,1] \times [0,1]$ and consider problem Eq. (2.1) with a deterministic coefficient $K(\mathbf{x})$ as shown in Fig. 1(a) and $f(\mathbf{x}) = 0$. The boundary conditions are chosen to induce flow from left to right: $p = 1$ on the left boundary, $p = 0$ on the right boundary, and no flow $\mathbf{u} \cdot \nu = 0$ on the top and bottom boundaries. Also in Fig. 1, we show the three different *training permeabilities* for this problem.

Table 1
Number of interface iterations for Example 7.1 as the subdomain mesh size, h, decreases and the number of subdomains and the mortar degree, m, remain fixed.

h	Subdomains	m	Iterations
1/64	16	1	10
1/128	16	1	10
1/256	16	1	10
1/512	16	1	10

Table 2
Number of interface iterations for Example 7.1 as the number of subdomains increases and the subdomain mesh size, h, and the mortar degree, m, remain fixed.

h	Subdomains	m	Iterations
1/128	4	1	10
1/128	16	1	10
1/128	64	1	10
1/128	256	1	10

Table 3
Number of interface iterations for Example 7.1 as the mortar degree, m, increases and the subdomain mesh size and the number of subdomains remain fixed. Here, (d) denotes discontinuous polynomials.

h	Subdomains	m	Iterations
1/128	16	1	10
1/128	16	2	10
1/128	16	3	10
1/128	16	1(d)	10
1/128	16	2(d)	10
1/128	16	3(d)	10

The first permeability is very simple and reflects none of the physics. The second is somewhat closer, but still does not reflect the discontinuities in the true permeability. The third *training permeability* matches only the order of magnitude in each subdomain. For this example, we use nonmatching grids as shown in Fig. 2(a) and continuous linear mortars. The outer iterative method is GMRES (unrestarted) with a tolerance of 10^{-6} and the preconditioned system is solved using conjugate gradients with a tolerance of 10^{-8}. The computed pressure is shown in Fig. 2(b).

In Fig. 3, we plot the norm of the relative residual using multiscale basis preconditioner from each of the *training permeabilities*. We observe that choosing a *training permeability* that captures as much of the physics as possible significantly reduces the number of interface iterations.

In this example we could have chosen the *training permeability* to match the true permeability and obtained convergence in one step. However, recall that our true interest is in solving a sequence of problems where the permeability is different for each realization.

7.3. Permeabilities generated using a KL expansion

In this section, we compare the performance of a number of domain decomposition strategies for the solution of a series of interface problems generated using a truncated KL expansion with a nonintrusive stochastic method such as Monte Carlo or stochastic collocation.

The four strategies considered in this paper are:

- DD: standard iterative domain decomposition without preconditioning.
- BDD: iterative domain decomposition with a balancing preconditioner.
- MS: multiscale domain decomposition with the multiscale basis recomputed for each realization.
- MSPRE: iterative domain decomposition with multiscale basis preconditioner.

The comparison of the methods is done in terms of number of subdomain solves, which is the dominant computational cost. In DD, one solve per subdomain is required for each interface iteration. In BDD, three solves are required per subdomain per iteration: one Dirichlet solve for the evaluation of the interface operator, and two Neumann solves to apply the preconditioner [14,35]. In MS, instead of solving subdomain problems for each iteration, the action of the interface operator is computed via a linear combination of a precomputed multiscale basis. A new basis is computed for each realization. The dominant cost is the computation of the multiscale basis, which requires as many subdomain solves as the number of mortar degrees of freedom per subdomain. In MSPRE, the iterative procedure requires one solve per subdomain for each iteration. The preconditioner uses the multiscale basis constructed using the *training permeability* which we take to be the mean permeability. Each application of the preconditioner requires solving a coarse scale equation using the *training* multiscale basis, which does not involve additional subdomain solves.

Define $D = [0,1] \times [0,1]$ and consider problem Eq. (2.1) with stochastic permeability $K(\mathbf{x},\omega)$ that has an expected value shown in Fig. 4(a). The boundary conditions are chosen to induce flow from left to right as in the previous example. We discretize D into 25 subdomains in a 5×5 pattern, each with $h = 1/360$ giving 5184 equal sized elements per subdomain. The total number of elements in the mesh is 129,600. We choose continuous quadratic mortars and set $H \approx \sqrt{h}$, giving 17 mortar degrees of freedom per subdomain boundary. The choice of high-order mortars on a coarse mesh

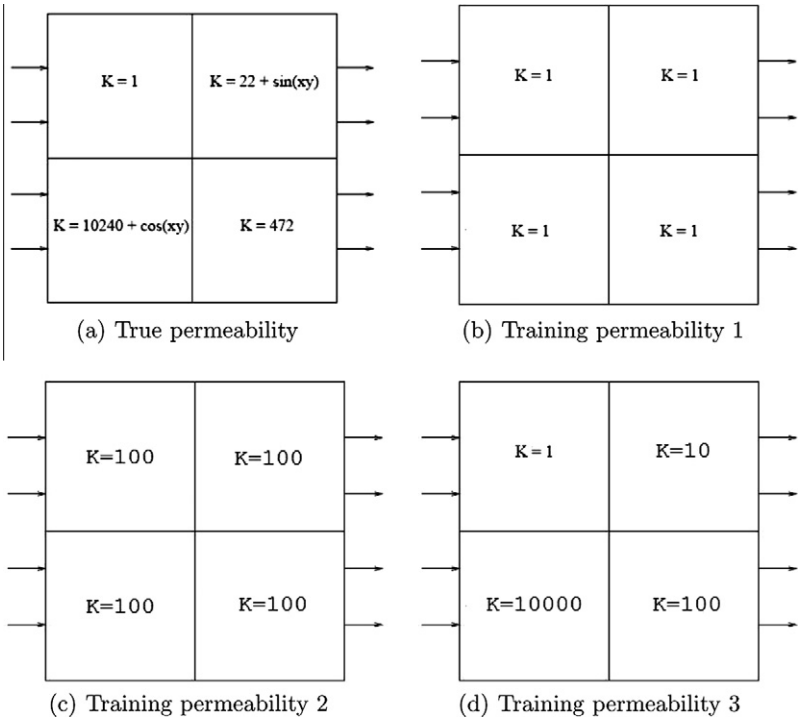


Fig. 1. True and training permeabilities for Example 7.2.

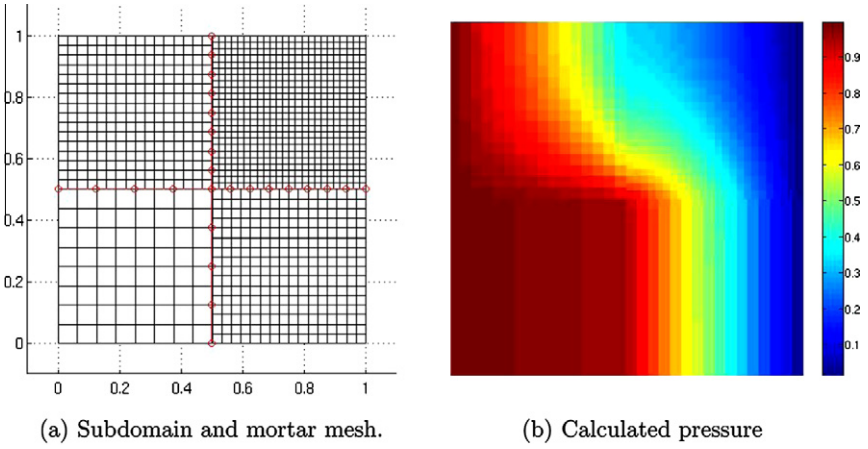


Fig. 2. Discretizations and mixed finite element solution for Example 7.2.

is made to balance the fine scale and coarse scale errors in order to obtain optimal order of accuracy on the fine scale (see Remark 4.2 and [3]).

The maximum number of mortar degrees of freedom associated with a subdomain is 68, but some of the subdomains do not have mortars on all boundaries and therefore have fewer mortar degrees of freedom. In the tables below we report that each subdomain performs 68 solves because if the multiscale basis is computed in a parallel environment, those subdomains with less than 68 mortar degrees of freedom will be waiting for the others to finish. In Fig. 4(b), we plot the magnitude of the velocity field corresponding to the mean of the permeability.

The performance of the multiscale basis preconditioner for each permeability realization depends on the perturbation from the mean permeability. This difference depends on the variance σ_Y, which affects the magnitude of the perturbations from the mean permeability field, and the number of terms in the truncated Karhunen–Loéve expansion, which affects the fine scale variability. We use correlation lengths $\eta_1 = 0.2$ and $\eta_2 = 0.125$ in the x_i – directions. The mean removed permeability field is assumed to be Gaussian, making the random variables mutually uncorrelated with zero mean and unit variance. We test the multiscale preconditioner by varying both σ_Y and the number of terms in the KL expansion.

If the number of terms (stochastic dimension) is relatively low, then we use a stochastic collocation method with collocation points chosen to be a tensor product of the roots of global Hermite polynomials as in [19]. The accuracy of the computed statistics depends on the order of the collocation method. Here, we choose a relatively low order method with two collocation points per dimension since our focus is on the performance of the preconditioner. If the number of terms in the KL expansion is relatively

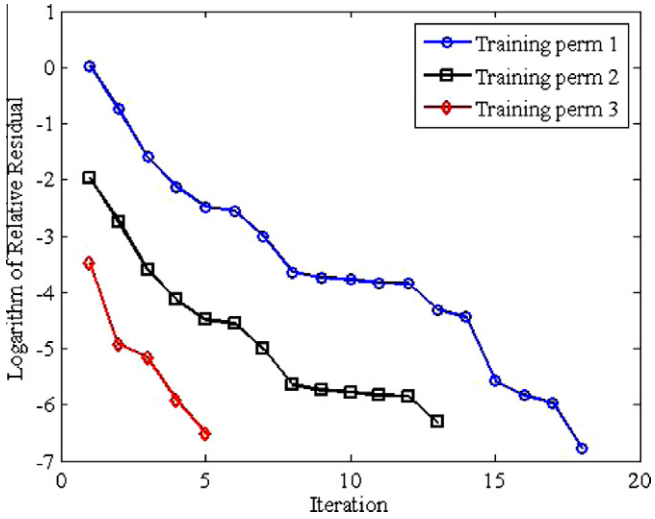


Fig. 3. Norm of the relative residual using the multiscale basis preconditioner corresponding to each of the *training permeabilities* for Example 7.2.

large, then we use a Monte Carlo sampling method to generate the realizations.

All of the following numerical experiments use GMRES for the interface iterations with a tolerance of 10^{-6}. The preconditioner is applied using GMRES using a block Jacobi preconditioner with a coarse scale correction (see [1,20]) with a tolerance of 10^{-8}. The block Jacobi preconditioner is only applicable when the multiscale basis has been assembled. We can also use interface multigrid [43,20] to efficiently solve the interface problem required for the application of the preconditioner.

In the tables below we report some statistics for the number of solves per subdomain: the total number of subdomain solves for all realizations, the average number of subdomain solves per realization, as well as the minimum and maximum number of subdomain solves amongst the realizations. For a fair comparison, the number 68 has been added to the total number of solves for MSPRE to account for the cost in assembling the preconditioner.

7.3.1. Fixed number of kl terms, Varying σ_Y

In the first set of experiments we take 4 terms in the KL expansion with second order stochastic collocation in each dimension, giving $2^4 = 16$ realizations. We test three cases, $\sigma_Y = 0.25$, 1, and 10. In Table 4, we give the performance of the preconditioner for $\sigma_Y = 0.25$ corresponding to the perturbation of the permeability field in each realization being relatively small.

For DD and MSPRE, the number of subdomain solves equals the number of interface iterations. The multiscale basis preconditioner performs very well and reduces significantly the number of

Table 4
Number of solves per subdomain for Example 7.3 with $\sigma_Y = 0.25$, 4 terms in the KL expansion, and 16 stochastic realizations.

Method	Total	Average	Minimum	Maximum
DD	2825	176.6	165	186
BDD	1674	104.6	99	105
MS	1088	68	68	68
MSPRE	64(+68)	4	4	4

Table 5
Number of solves per subdomain for Example 7.3 with $\sigma_Y = 1.0$, 4 terms in the KL expansion, and 16 stochastic realizations.

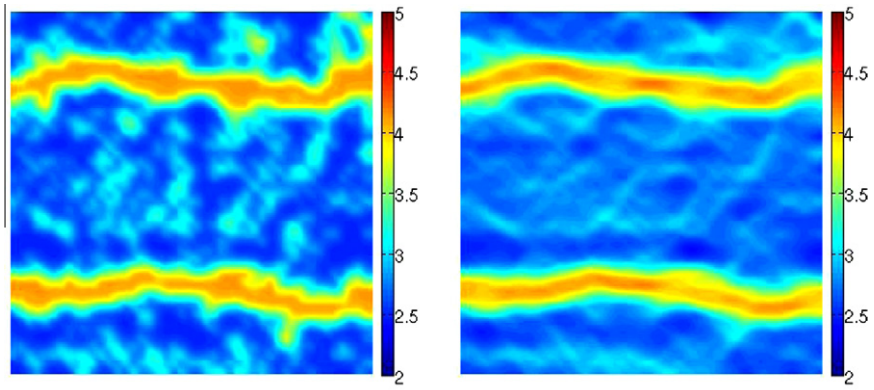
Method	Total	Average	Minimum	Maximum
DD	2960	185	169	195
BDD	1680	105	105	105
MS	1088	68	68	68
MSPRE	84(+68)	5.25	4	6

Table 6
Number of solves per subdomain for Example 7.3 with $\sigma_Y = 10.0$, 4 terms in the KL expansion, and 16 stochastic realizations.

Method	Total	Average	Minimum	Maximum
DD	3827	239.2	195	285
BDD	1656	103.5	99	105
MS	1088	68	68	68
MSPRE	216(+68)	13.3	7	19

Table 7
Number of solves per subdomain for the model problem in Section 7.3 with $\sigma_Y = 1.0$, 9 terms in the KL expansion, and 512 realizations corresponding to the samples points for second order stochastic collocation.

Method	Total	Average	Minimum	Maximum
DD	98544	192.5	160	225
BDD	53106	103.7	99	105
MS	34816	68	68	68
MSPRE	3463(+68)	6.8	5	10



(a) Mean of the permeability field.

(b) Calculated velocity corresponding to the mean permeability.

Fig. 4. Mean permeability and the magnitude of the corresponding velocity solution for Example 7.3.

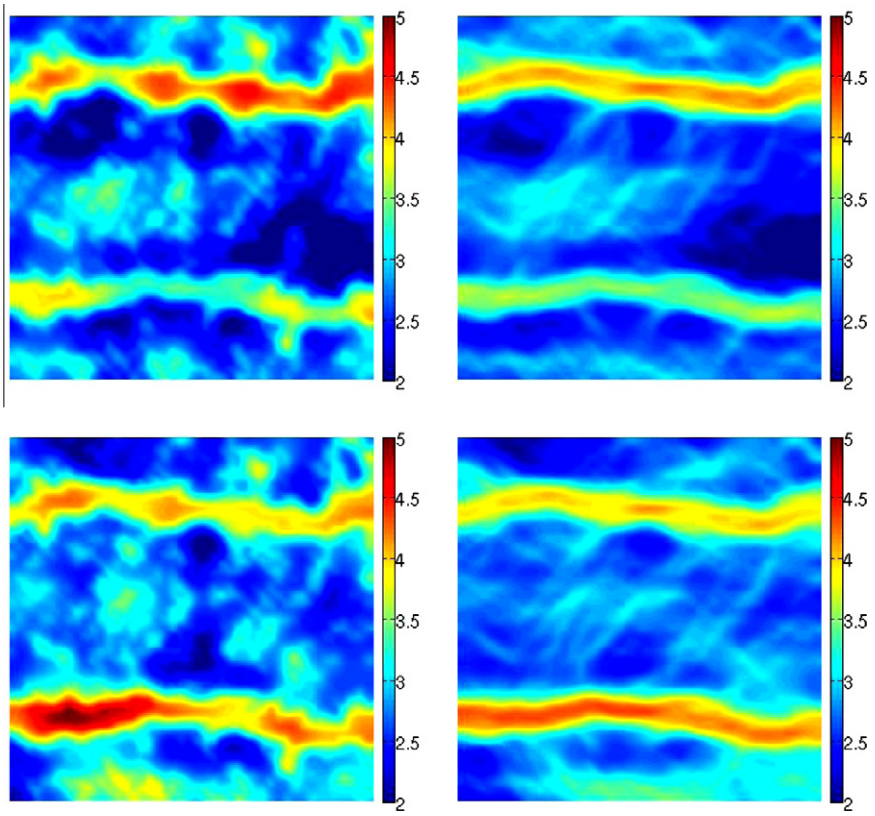


Fig. 5. Two stochastic permeability realizations (left) and the logarithm of the magnitude of the corresponding velocity fields (right) for Example 7.2 with 100 terms in the KL expansion.

Table 8
Number of solves per subdomain for Example 7.3 with $\sigma_Y = 1.0$, 100 terms in the KL expansion, and 1000 Monte Carlo realizations.

Method	Total	Average	Minimum	Maximum
DD	203587	203.6	144	283
BDD	104460	104.5	99	108
MS	68000	68	68	68
MSPRE	11262(+68)	11.3	6	33

subdomain solves. For MS, the number of subdomain solves is determined by the number of multiscale basis functions for each realization. Hence the number of solves does not depend on the number of interface iterations and stays constant over all realizations. The multiscale basis preconditioner performs the fewest number of subdomain solves amongst the methods considered. In Tables 5 and 6, we give the results for $\sigma_Y = 1$ and $\sigma_Y = 10$. As we increase σ_Y the average performance of the multiscale basis preconditioner degrades slightly, but it is still significantly more efficient than the other two methods.

7.3.2. Fixed σ_Y, varying the number of KL terms

In the second set of experiments, we set $\sigma_Y = 1.0$ and vary the number of terms in the KL expansion. First, we compare the number of subdomain solves for each of the four methods using 9 terms in the KL expansion. As in the previous cases, two stochastic collocation points were chosen in each stochastic dimension, giving $2^9 = 512$ realizations. In Table 7, we see that reusing the multiscale basis as a preconditioner requires an order of magnitude fewer solves per subdomains than recomputing the multiscale basis for each realization, and in even larger savings compared to the DD and BDD approaches. Note also that the proportion of the cost associated with the construction the preconditioner (68 solves per subdomain) is smaller in this case in comparison with the previous cases with fewer realizations.

Next, we increase the number of terms in the KL expansion to 100. Since the stochastic dimension is 100, tensor product stochastic collocation using only 2 collocation points per dimension would require 2^{100} realizations which is computationally infeasible. A sparse grid collocation method would greatly reduce the number of realizations required [17,18,32], but our focus is on the performance of the preconditioner for typical realizations rather than actually computing a mean or variance of the solution. Therefore, we use a Monte Carlo sampling technique to generate 1000 realizations and test the preconditioner on these problems. In Fig. 5 we plot two permeability realizations (left) and the logarithm of the magnitude of the corresponding velocity fields (right). The computational cost for the four methods is reported in Table 8. Despite the fact that we have introduced more variability into the permeability realizations, the multiscale basis preconditioner based on the expected permeability still performs quite well and is easily the most efficient of the four methods.

Remark 7.1. We note that decreasing the subdomain mesh size h and increasing the number of subdomains would increase the number of solves per subdomain for DD and, with a smaller polylogarithmic rate, for BDD, while the number of solves per subdomain would remain fixed for MS and MSPRE. Furthermore, refining the mortar mesh, i.e., decreasing H, would also increase the number of solves per subdomain for DD and BDD due to more interface iterations. The effect on MS and MSPRE would also be increased number of solves per subdomain to compute the basis, which is less significant, especially for MSPRE, since the basis is computed only once.

We end this section with a remark on constructing multiple training operators. As the numerical results demonstrate, the multiscale preconditioner is most effective if the realizations are relatively close to the training permeability. If the perturbations grow too large, then using MSPRE may become more expensive than MS or BDD. In this case, multiple preconditioners may be constructed and used only for "nearby" realizations. For example, if a Markov chain Monte Carlo method is used, one multiscale preconditioner may be used for each random walk. If a truncated KL expansion with P terms is used with stochastic collocation, a multiscale basis may be constructed for each realization of a shorter KL expansion with $L < P$ terms. Each realization of the longer KL expansion can be viewed as a perturbation of a particular realization in the shorter expansion. Therefore, the multiscale basis corresponding to the realization in the shorter expansion should be a better preconditioner than the multiscale basis from the mean permeability. We elected not to pursue this issue in this paper for the sake of space, but this may be the subject of future work.

8. Conclusion

We have introduced a new approach for efficient uncertainty quantification by combining a nonintrusive stochastic method with the construction of a multiscale mortar basis. The multiscale basis is constructed for a carefully chosen *training operator* and is reused as a preconditioner for subsequent realizations.

The numerical results confirm that the condition number of the multiscale basis preconditioned interface problem is independent of the subdomain mesh size h and the mortar mesh size H. Furthermore, for stochastic flow in porous media, the preconditioner reuses the multiscale basis, leading to a very efficient algorithm. The performance of the preconditioner is relatively robust with respect to the variance of the stochastic permeability and the dimension of the stochastic space if a suitable *training permeability* is chosen.

Acknowledgments

We thank the referees for their useful suggestions. The first two authors were partially supported by the NSF grant DMS 0618679 and the DOE grant DE-FG02-04ER25617. The third author was partially supported by the NSF grants DMS 0620402 and DMS 0813901 and the DOE grant DE-FG02-04ER25618.

References

[1] Y. Achdou, Y. Maday, O. Widlund, Iterative substructuring preconditioners for mortar element methods in two dimensions, SIAM J. Numer. Anal. 36 (2) (1999) 551–580.
[2] T. Arbogast, L.C. Cowsar, M.F. Wheeler, I. Yotov, Mixed finite element methods on nonmatching grid blocks, SIAM J. Numer. Anal. 37 (2000) 1295–1315.
[3] T. Arbogast, G. Pencheva, M.F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, Multiscale Model. Simul. 6 (1) (2007) 319–346.
[4] B.V. Asokan, N. Zabaras, A stochastic variational multiscale method for diffusion in heterogeneous random media, J. Comput. Phys. 218 (2) (2006) 654–676.
[5] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM J. Numer. Anal. 45 (3) (2007) 1005–1034.
[6] I. Babuska, R. Tempone, G.E. Zouraris, Galerkin finite element approximations of stochastic differential equations, SIAM J. Numer. Anal. 42 (2) (2004) 800–825.
[7] P.E. Bjorstad, O.B. Widlund, Iterative methods for the solution of elliptic problems on regions partitioned into substructures, SIAM J. Numer. Anal. 23 (6) (1986) 1097–1120.
[8] J.H. Bramble, J.E. Pasciak, A.H. Schatz, The construction of preconditioners for elliptic problems by substructuring. I, Math. Comput. 47 (175) (1986) 103–134.
[9] F. Brezzi, J. Douglas Jr., R. Durán, M. Fortin, Mixed finite elements for second order elliptic problems in three variables, Numer. Math. 51 (2) (1987) 237–250.
[10] F. Brezzi, J. Douglas Jr., M. Fortin, L.D. Marini, Efficient rectangular mixed finite elements in two and three space variables, RAIRO Modél. Math. Anal. Numér. 21 (4) (1987) 581–604.
[11] F. Brezzi, J. Douglas Jr., L.D. Marini, Two families of mixed finite elements for second order elliptic problems, Numer. Math. 47 (2) (1985) 217–235.
[12] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag, New York, 1991.
[13] M. Chen, D. Zhang, A. Keller, Z. Lu, Stochastic analysis of two phase flow in heterogeneous media by combining Karhunen–Loéve expansion and perturbation method, Water Resour. Res. 41 (1) (2005) W01006. 1–14pp..
[14] L.C. Cowsar, J. Mandel, M.F. Wheeler, Balancing domain decomposition for mixed finite elements, Math. Comput. 64 (1995) 989–1015.
[15] M.K. Deb, I.M. Babuška, J.T. Oden, Solution of stochastic partial differential equations using Galerkin finite element techniques, Comput. Meth. Appl. Mech. Engrg. 190 (48) (2001) 6359–6372.
[16] P. Dostert, Y. Efendiev, T.Y. Hou, Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification, Comput. Meth. Appl. Mech. Engrg. 197 (43–44) (2008) 3445–3455 (Stochastic Modeling of Multiscale and Multiphysics Problems).
[17] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation methods for stochastic natural convection problems, J. Comput. Phys. 225 (2007) 652–685.
[18] B. Ganapathysubramanian, N. Zabaras, A stochastic multiscale framework for modeling flow through random heterogeneous porous media, J. Comput. Phys. 228 (2) (2009) 591–618.
[19] B. Ganis, H. Klie, M.F. Wheeler, T. Wildey, I. Yotov, D. Zhang, Stochastic collocation and mixed finite elements for flow in porous media, Comput. Meth. Appl. Mech. Engrg. 197 (43–44) (2008) 3547–3559 (Stochastic Modeling of Multiscale and Multiphysics Problems).
[20] B. Ganis, M.F. Wheeler, T. Wildey, I. Yotov, Computational efficiency of the multiscale mortar mixed finite element approach to domain decomposition, in preparation.
[21] B. Ganis, I. Yotov, Implementation of a mortar mixed finite element method using a multiscale flux basis, Comput. Meth. Appl. Mech. Engrg. 198 (2009) 3989–3998.
[22] B. Ganis, I. Yotov, M. Zhong, A stochastic mortar mixed finite element method for flow in porous media with multiple rock types, Technical Report TR-MATH 10-06, Department of Mathematics, University of Pittsburgh, 2010.
[23] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, 1991.
[24] R. Glowinski, M.F. Wheeler, Domain decomposition and mixed finite element methods for elliptic problems, in: R. Glowinski, G.H. Golub, G.A. Meurant, J. Periaux (Eds.), First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988, pp. 144–172.
[25] C. Jin, X. Cai, C. Li, Parallel domain decomposition methods for stochastic elliptic equations, SIAM J. Sci. Comput. 29 (5) (2008) 2096–2114.
[26] H. Li, D. Zhang, Probabilistic collocation method for flow in porous media: comparisons with other stochastic methods, Water Resour. Res. 43 (2007) W09409, doi:10.1029/2006WR005673.
[27] J. Mandel, Balancing domain decomposition, Commun. Numer. Meth. Engrg. 9 (3) (1993) 233–241.
[28] J. Mandel, M. Vidrascu, P. Le Tallec, Neumann–Neumann domain decomposition algorithms for solving 2D elliptic problems with nonmatching grids, SIAM J. Numer. Anal. 35 (35) (1998) 836–867.
[29] V. Narayanan, N. Zabaras, Variational multiscale stabilized FEM formulations for transport equations: stochastic advection-diffusion and incompressible stochastic Navier–Stokes equations, J. Comput. Phys. 202 (1) (2005) 94–133.
[30] J.-C. Nédélec, Mixed finite elements in R^3, Numer. Math. 35 (3) (1980) 315–341.
[31] F. Nobile, R. Tempone, C.G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, SIAM J. Numer. Anal. 46 (5) (2008) 2411–2442.
[32] F. Nobile, R. Tempone, C.G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM J. Numer. Anal. 46 (5) (2008) 2309–2345.
[33] B. Oksendal, Stochastic Differential Equations: An Introduction with Applications, Springer-Verlag, Berlin, 1998.
[34] G. Pencheva, M. Vohralik, M.F. Wheeler, T. Wildey, A posteriori error control and adaptivity for multiscale, multinumerics, and mortar coupling, Technical Report 10-15, ICES, The University of Texas at Austin, 2010.
[35] G. Pencheva, I. Yotov, Balancing domain decomposition for mortar mixed finite element methods on non-matching grids, Numer. Linear Algebr. Appl. 10 (2003) 159–180.
[36] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, UK, 2005.
[37] P.-A. Raviart, J.M. Thomas, A mixed finite element method for 2nd order elliptic problems, in: Mathematical Aspects of Finite Element Methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975), Lecture Notes in Mathematics, vol. 606, Springer, Berlin, 1977.
[38] C. Schwab, R. Todor, Karhunen–Loève approximation of random fields by generalized fast multipole methods, J. Comput. Phys. 217 (2006) 100–122.
[39] B.F. Smith, P. Bjorstad, W. Gropp, Domain Decomposition, Cambridge University Press, UK, 1996.
[40] R.A. Todor, C. Schwab, Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients, IMA J. Numer. Anal. 27 (2007) 232–261.

[41] X. Wan, G.E. Karniadakis, Beyond Weiner–Askey expansions: handling arbitrary pdfs, J. Sci. Comput. 27 (1–3) (2006) 455–464.
[42] M. Wheeler, I. Yotov, A posteriori error estimates for the mortar mixed finite element method, SIAM J. Numer. Anal. 43 (3) (2005) 1021–1042.
[43] M.F. Wheeler, I. Yotov, Multigrid on the interface for mortar mixed finite element methods for elliptic problems, Comput. Meth. Appl. Mech. Engrg. 184 (2000) 287–302.
[44] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2) (2007) 293–309.
[45] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.
[46] X.F. Xu, A multiscale stochastic finite element method on elliptic problems involving uncertainties, Comput. Meth. Appl. Mech. Engrg. 196 (25–28) (2007) 2723–2736.
[47] I. Yotov, Mixed finite element methods for flow in porous media, PhD Thesis, Rice University, Houston, TX, 1996. Also TR96-09, Department of Computational and Applied Mathematics, Rice University and TICAM report 96-23, University of Texas at Austin.
[48] D. Zhang, Stochastic Methods for Flow in Porous Media: Coping with Uncertainties, Academic Press, San Diego, CA, 2002.
[49] D. Zhang, Z. Lu, An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions, J. Comput. Phys. 194 (2) (2004) 773–794.
[50] K. Zhang, R. Zhang, Y. Yin, S. Yu, Domain decomposition methods for linear and semilinear elliptic stochastic partial differential equations, Appl. Math. Comput. 195 (2) (2008) 630–640.