# A multiscale preconditioner for stochastic mortar mixed finite elements

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

[a] *Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260*

[b] *The Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, Austin, TX 78712*

**Abstract**

The aim of this paper is to introduce a new approach to efficiently perform uncertainty quantification for flow in porous media through stochastic modeling. The governing equations are based on Darcy's law with stochastic permeability. 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 basis is computed for a single permeability that captures the main characteristics of the porous media, called a *training permeability*, and 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.

*Key words:* stochastic collocation, mixed finite element, stochastic partial differential equations, flow in porous media, interface preconditioner

## 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 the last years [12,42]. 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 [4,40,39]. 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 [5,37], making it more efficient than MCS. Choices of collocation points include tensor product of zeros of orthogonal polynomials [4,40], sparse grid approximations [15,27,36,40], and probabilistic collocation [22].

A tremendous amount of research over the past thirty years has been devoted to the efficient parallel solution of deterministic subsurface flow problems. Nonoverlapping 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 multiscale method [3] with the subdomain problems serving as the fine scale and the interface problem representing the coarse scale. This interpretation allows for *a posteriori* error estimates and adaptive mesh refinement for both the subdomain and mortar scales. The coarse scale interface problem is usually solved via an iterative method such as the conjugate gradient or GMRES [21], which relies on the availability of an efficient and robust preconditioner, e.g., Neumann-Neumann [24], balancing [23,13,29], or multigrid [38].

Recently, an alternative algorithm, based on the multiscale interpretation, has been developed for the mortar multiscale method [19]. In this approach, a set of multiscale basis functions are computed for the coarse scale interface problem, which is then solved efficiently in parallel. The cost of computing a multiscale basis function is solving a subdomain problem. The number of multiscale basis functions per subdomain is equal to the number of coarse scale mortar degrees of freedom for this subdomain. In [19], this was shown to outperform the original algorithm with the balancing preconditioner in certain scenarios. The multiscale basis approach has been applied to stochastic problems in [18]. There, a multiscale 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 that alleviates the need to recompute the multiscale basis for each realization. A multiscale mortar basis is computed for a single permeability that captures the main characteristics of the porous media, called a *training permeability.* The resulting training operator is used as a preconditioner for each stochastic realization. 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. The cost of 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.

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

## 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 $\Omega$ be a stochastic event space with probability measure $P$. The Darcy velocity $\mathbf{u}$ and the pressure $p$ satisfy $P$-almost everywhere in $\Omega$

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

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 $\Omega$, 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 $\lambda_i$ 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, if $Y'$ is given by a Gaussian process, $\xi_i$ are mutually uncorrelated random variables with zero mean and unit variance [20].

At this point, the KL expansion is truncated after $N$ terms, which is reasonable to do as typically the $\lambda_i$ decay rapidly [43]. 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 $\Omega_i = \xi_i(\Omega)$ make up a finite dimensional vector space $\Omega^N = \prod_{i=1}^N \Omega_i \subset \mathbb{R}^N$. If $\rho_i$ corresponds to the p.d.f. of each $\xi_i$, 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

[43], in which $\lambda_i$ and $f_i(\mathbf{x})$ can be found analytically.

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

Here $\sigma_Y$ and $\eta_i$ denote variance and correlation length in the $i$-th 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 equation (2.3). Efficient methods for numerically computing the KL expansion are reported in [34].

### *2.2 Variational Formulation*

Appealing to the Doob-Dynkn Lemma [28], the p.d.f. for the permeability $K$ carries through to the solution of (2.1), so that $(\mathbf{u},p)$ has the form

$$\mathbf{u}(\mathbf{x},\omega) = \mathbf{u}(\mathbf{x},\xi_1(\omega),\dots,\xi_N(\omega)) = \mathbf{u}(\mathbf{x},y_1,\dots,y_N) \text{ and}$$

$$p(\mathbf{x},\omega) = p(\mathbf{x},\xi_1(\omega),\dots,\xi_N(\omega)) = p(\mathbf{x},y_1,\dots,y_N).$$

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 $\Gamma_{i,j} = \partial D_i \cap \partial D_j$, $\Gamma = \cup_{1\le i<j\le n}\Gamma_{i,j}$, and $\Gamma_i = \partial D_i \cap \Gamma = \partial D_i \backslash \partial D$ denote interior block interfaces. Let

$$\mathbf{V}_i = H(\text{div}; D_i), \quad \mathbf{V} = \bigoplus_{i=1}^n \mathbf{V}_i,$$
$$W_i = L^2(D_i), \quad W = \bigoplus_{i=1}^n W_i = L^2(D),$$

$$M = \{\mu \in H^{1/2}(\Gamma) \ : \ \mu|_{\Gamma_i} \in (\mathbf{V}_i\cdot\nu_i)^*,\, i=1,\dots,n\},$$

where where $\nu_i$ is the outer unit normal to $\partial D_i$ 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 [2,16,18], a weak form of (2.1) seeks $\mathbf{u}(\mathbf{x},\omega) \in \mathbf{V}\otimes L^2(\Omega^N)$, $p(\mathbf{x},\omega) \in W\otimes L^2(\Omega^N)$ and $\lambda(\mathbf{x},\omega) \in$

$M \otimes L^2(\Omega^N)$ such that, for each $i$,

$$\int_{\Omega^N} (K^{-1}\mathbf{u}, \mathbf{v})_{D_i}\ \rho(\mathbf{y})\ d\mathbf{y} = \int_{\Omega^N} \Big( (p, \nabla \cdot \mathbf{v})_{D_i} \\ - \langle \lambda, \mathbf{v} \cdot \nu_i \rangle_{\Gamma_i} - \langle g, \mathbf{v} \cdot \nu_i \rangle_{\partial D_i \setminus \Gamma} \Big)\ \rho(\mathbf{y})\ d\mathbf{y}, \quad (2.6)$$

$$\int_{\Omega^N} (\nabla \cdot \mathbf{u}, w)_{D_i}\ \rho(\mathbf{y})\ d\mathbf{y} = \int_{\Omega^N} (f, w)_{D_i}\ \rho(\mathbf{y})\ d\mathbf{y}, \quad (2.7)$$

$$\sum_{i=1}^{n} \int_{\Omega^N} \langle \mathbf{u} \cdot \nu_i, \mu \rangle_{\Gamma_i}\ \rho(\mathbf{y})\ d\mathbf{y} = 0, \quad (2.8)$$

for all $\mathbf{v} \in \mathbf{V}_i \otimes L^2(\Omega^N)$, $w \in W_i \otimes L^2(\Omega^N)$, and $\mu \in M \otimes L^2(\Omega^N)$. Note that $\lambda$ is the pressure on the block interfaces $\Gamma$ and that (2.8) enforces weak continuity of $\mathbf{u} \cdot \nu$ on each $\Gamma_{i,j}$.

## 3 Nonintrusive Stochastic Methods

After expressing the log permeability as a truncated KL expansion, the problem has now been reformulated in the finite dimensional space $D \otimes \Omega^N \in \mathbb{R}^{d+N}$. At this point, there are several ways in which to discretize the problem. The Stochastic Finite Element Method (SFEM) [14] 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 $\Omega^N$ 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, $O(1/\sqrt{\tilde{M}})$, where $\tilde{M}$ is the number of realizations.

The Stochastic Collocation Method improves upon the Monte Carlo approach 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 [16,4,40,39].

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 (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, $\tilde{M} = k^N$. To cope with this problem, more advanced collocation techniques are possible such as the so called probabilistic collocation method (see *e.g.* [22]) and the Smolyak sparse grids (see *e.g.* [39,27,26]) but will not be considered in this paper.

Regardless of the choice of nonintrusive stochastic method, we assume henceforth that the goal is to find, for $1 \le m \le \tilde{M}$, $\mathbf{u}^{\{m\}} \in \mathbf{V}$, $p^{\{m\}} \in W$ and $\lambda^{\{m\}} \in M$ such that for each $i$

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

for all $\mathbf{v} \in \mathbf{V}_i$, $w \in W_i$, and $\mu \in M$, where $\{\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_{\tilde{M}}\}$ are the chosen sample points in $\Omega^N$, and

$$K^{\{m\}} = K(\mathbf{x}, \mathbf{y}_m).$$

## 4 The finite element approximation

Let $\mathcal{T}_{h,i}$ be a conforming quasi-uniform affine finite element partition of $D_i$, $1 \le i \le n$, 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 $h = \max_{1 \le i \le n} h_i$ 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 [8,9,25,10,31]), 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 $\Gamma$.

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 $H = \max_{1\le i,j\le n} H_{i,j}$. Define $\mathcal{T}^{\Gamma,H} = \cup_{1\le i<j\le n}\mathcal{T}_{H,i,j}$. 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,

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

be the mortar finite element space on $\Gamma$. 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*

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

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 $h \le H \le 1$. Condition (4.2) is not very restrictive, and is easily satisfied in practice (see, e.g., [29,41]). In the following, we treat any function $\mu \in M_H$ as extended by zero on $\partial D$.

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

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

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 $M_H$.

## 5 A multiscale domain decomposition formulation

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

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

where the pair $(\mathbf{u}_h^{*,\{m\}}(\lambda), p_h^{*,\{m\}}(\lambda)) \in \mathbf{V}_h \times W_h$ solves

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

for each $1 \le i \le n$. 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 $(\bar{\mathbf{u}}_h^{\{m\}}, \bar{p}_h^{\{m\}}) \in \mathbf{V}_h \times W_h$ solves, for $1 \le i \le n$,

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

It is straightforward to show (see [2,21]) that the solution of

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

generates a solution of (4.3) via

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

The following result has been shown in [2].

**Lemma 5.1** *For $1 \le m \le \tilde{M}$, 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,*

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

For each $1 \le m \le \tilde{M}$ we define a real $N_H \times N_H$ matrix $\mathcal{D}_H^{\{m\}}$ satisfying

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

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

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

Note that the interface problem (5.3) can be written as

$$\mathcal{D}_H^{\{m\}}\boldsymbol{\lambda}_H^{\{m\}} = \boldsymbol{g}_H^{\{m\}}.$$

The operator $\mathcal{D}_H^{\{m\}}$ is the mortar version of the Steklov-Poincaré operator [30].

We assume that there exist positive constants $\hat{c}^{\{m\}}$, $\hat{C}^{\{m\}}$, and $\alpha_i$ such that

$$\hat{c}^{\{m\}}\alpha_i\xi^T\xi \le \xi^T K^{\{m\}}(\mathbf{x})\xi \le \hat{C}^{\{m\}}\alpha_i\xi^T\xi, \quad \forall \xi \in \mathbb{R}^d, \ \forall \mathbf{x} \in D_i, \ i = 1, \ldots, n. \tag{5.6}$$

It is shown in [41], see also [13,29], that there exist positive constants $c^{\{m\}}$ and $C^{\{m\}}$ such that

$$c^{\{m\}}\sum_{i=1}^n \alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i} \le \left[\mathcal{D}_H^{\{m\}}\boldsymbol{\mu}, \boldsymbol{\mu}\right] \le C^{\{m\}}\sum_{i=1}^n \alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i}, \tag{5.7}$$

for all $\mu \in M_H$, where $\mathcal{I}^{\partial D_i}$ is a continuous piecewise linear interpolant defined in [13,41,29]. 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 $\alpha_i$ is given explicitly.

## 6 A multiscale preconditioner

In the original implementation of the mortar mixed finite element method [41,2], a substructuring domain decomposition algorithm based on the algorithm of Glowinski and Wheeler [21] is used to solve each linear system of equations resulting from (4.3) using the conjugate gradient (CG) algorithm. In that approach, the action of the interface operator requires the solution of subdomain problems for each iteration. In [3], the mortar mixed finite element method was shown to be equivalent to a nonstandard multiscale method with the subdomain problems corresponding to the fine scale and the mortar interface problem corresponding to the coarse scale. This relationship has been explored in [19], where a mortar multiscale flux basis is precomputed. The action of the interface is then computed by taking a linear combination of the multiscale basis functions, which is much faster than solving subdomain problems.

Following [19], 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 mortar multiscale flux basis is computed by calculating the flux response for each mortar basis function. This process is summarized in Algorithm 1.

**Algorithm 1** Construction of the Multiscale Basis

```
for $i = 1, 2, \dots, n$ do
  for $k = 1, 2, , \dots, N_{H,i}$ 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 problem
        $$\begin{cases} ((K^{\{m\}})^{-1}\mathbf{u}_h^{*,\{m\}}(\gamma_i^{\{k\}}), \mathbf{v})_{D_i} = (p_h^{*,\{m\}}(\gamma_i^{\{k\}}), \nabla\cdot\mathbf{v})_{D_i} - \left\langle \gamma_i^{\{k\}}, \mathbf{v}\cdot\nu_i \right\rangle_{\Gamma_i}, \\ (\nabla\cdot\mathbf{u}_h^{*,\{m\}}(\gamma_i^{\{k\}}), w)_{D_i} = 0. \end{cases} \tag{6.1}$$
    (c) Project the flux into the mortar space,
        $$\psi_{H,i}^{\{m\},\{k\}} = -\mathcal{Q}_{h,i}^T\mathbf{u}_h^{*,\{m\}}(\gamma_i^{\{k\}})\cdot\nu_i.$$
    (d) Store the multiscale basis function $\psi_{H,i}^{\{m\},\{k\}}$.
  end for
end for
```

In the above $\mathcal{Q}_{h,i}^T : \mathbf{V}_{h,i}\cdot\nu_i \to M_{H,i}$ is the $L^2$-orthogonal projection from the normal trace of the velocity space into the mortar space.

Note that the number of solves (6.1) depends only on the number of mortar degrees of freedom associated with a particular subdomain and may be different for each subdomain.

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

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

where, by abuse of notation, $\mathcal{D}_{H,i}^{\{m\}}$ is the operator corresponding to the matrix with the same name introduced earlier. In [19], 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 6.1** *This approach is closely related (but more general) than the substructuring methods developed in e.g., [7,6,1]. 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 [35,30].*

In [18], the multiscale basis method is combined with stochastic collocation to model flow in non-stationary random porous media. In that approach, however, Algorithm 1 must be repeated for each $1 \leq m \leq \tilde{M}$. To avoid recomputing the multiscale flux basis for each realization, we only implement Algorithm 1 for a carefully chosen *training operator*. This multiscale flux basis is then used to compute a preconditioner for each of the realizations.

Let $\overline{K}$ denote a uniformly positive and bounded *training permeability*. We use Algorithm 1 to compute a multiscale flux basis for the interface *training operator* $\overline{\mathcal{D}}_H$. Then, for each $1 \leq m \leq \tilde{M}$, we solve the preconditioned system

$$\overline{\mathcal{D}}_H^{-1} \mathcal{D}_H^{\{m\}} \boldsymbol{\lambda}_H^{\{m\}} = \overline{\mathcal{D}}_H^{-1} \boldsymbol{g}_H^{\{m\}} \tag{6.2}$$

using an iterative algorithm such as CG or GMRES. Note that in this approach, the action of $\mathcal{D}_H^{\{m\}}$ is computed by solving subdomain problems with the true permeability $K_m$, while the action of $\overline{\mathcal{D}}_H^{-1}$ is computed using the multiscale flux basis.

The preconditioner $\overline{\mathcal{D}}_H^{-1}$ is not explicitly constructed. Instead, we use an iterative algorithm such as CG to compute the action of the preconditioner as needed. On each CG iteration, the action of the interface operator $\overline{\mathcal{D}}_H$ is computed using the linear combination of multiscale basis functions

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

thus avoiding the need to solve any additional subdomain problems.

**Remark 6.2** *In general, the preconditioned system (6.2) is not symmetric, so GMRES is usually used. In fact, since the action of the preconditioner is computed another iterative method, we should use an algorithm that allows different preconditioners, e.g., FGMRES [33], to solve (6.2). This could substantially decrease the cost in applying the preconditioner and will be investigated in a future paper.*

The following result on the preconditioning of symmetric positive definite operators can be found in [30] Theorem 4.1.5.

**Theorem 6.3** *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*

$$C_1 \left[\mathcal{D}_1 \boldsymbol{\mu}, \boldsymbol{\mu}\right] \leq \left[\mathcal{D}_2 \boldsymbol{\mu}, \boldsymbol{\mu}\right] \leq C_2 \left[\mathcal{D}_1 \boldsymbol{\mu}, \boldsymbol{\mu}\right],$$

*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*

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

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

$$cond\left(\mathcal{D}_1^{-1}\mathcal{D}_2\right) \le \frac{C_2}{C_1}.$$

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

**Theorem 6.4** *Assume that (4.2) holds and that the training operator* $\overline{\mathcal{D}}_H$ *satisfies a bound similar to (5.7),*

$$\overline{c}\sum_{i=1}^{n}\alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i} \le \left[\overline{\mathcal{D}}_H\boldsymbol{\mu},\boldsymbol{\mu}\right] \le \overline{C}\sum_{i=1}^{n}\alpha_i|\mathcal{I}^{\partial D_i}\mathcal{Q}_{h,i}\mu|_{1/2,\partial D_i} \qquad (6.3)$$

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

$$cond\left(\overline{\mathcal{D}}_H^{-1}\mathcal{D}_H^{\{m\}}\right) \le \frac{\overline{C}C^{\{m\}}}{\overline{c}c^{\{m\}}}, \qquad (6.4)$$

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

**Proof.** From (5.7) and (6.3) we easily derive

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

The result (6.4) follows from Theorem 6.3. □

**Remark 6.5** *Although the condition number (6.4) does not depend on h or H, it does depend through (6.3) and (5.7) 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 doesn't 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 [17], 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 §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 §6. All results are computed in 2D using the lowest order Raviart-Thomas $\mathrm{RT}_0$ spaces [32,11] 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 (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 $4 \times 4$ 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.

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

Table 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.

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

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

Table 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.

Finally, we set $h = 1/128$ and fix the number of subdomains at 16 ($4 \times 4$). 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.

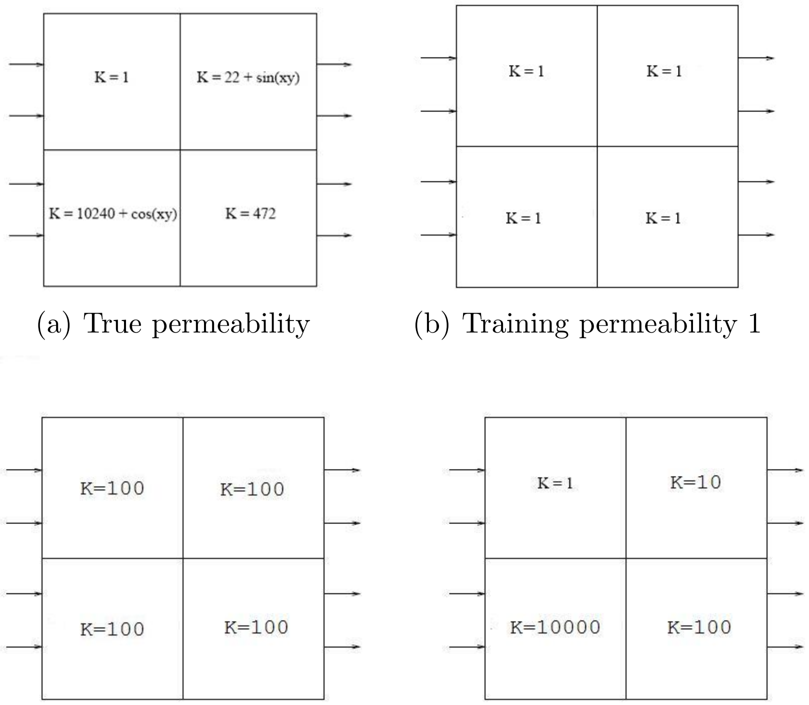
### *7.2 Choice of training permeability*

Define $D = [0,1] \times [0,1]$ and consider problem (2.1) with a deterministic cefficient $K(\mathbf{x})$ as shown in Figure 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 Figure 1, we show the three different *training permeabilities* for this problem. 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

| $h$ | 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 |

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.

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 Figure 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 Figure 2(b).



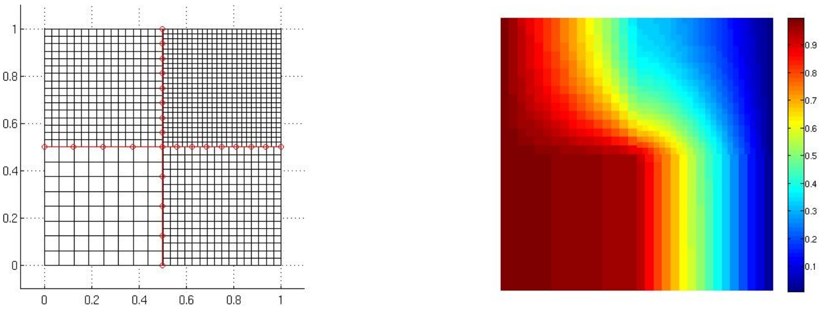
(a) True permeability

(b) Training permeability 1

(c) Training permeability 2

(d) Training permeability 3

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



(a) Subdomain and mortar mesh.

(b) Calculated pressure

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

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

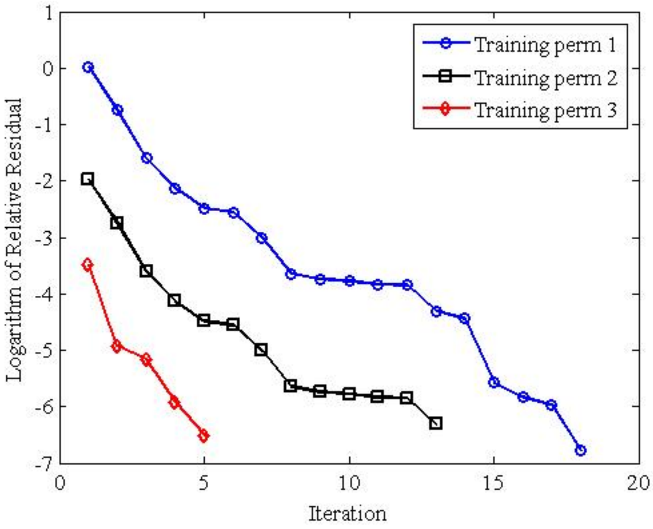
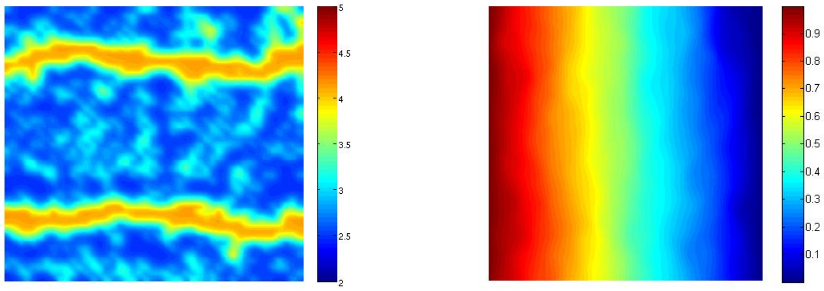


Fig. 3. Norm of the relative residual using the multiscale basis preconditioner corresponding to each of the *training permeabilities* for Example 7.2.

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 Uncertainty quantification*

Define $D = [0, 1] \times [0, 1]$ and consider problem (2.1) with stochastic permeability $K(\mathbf{x}, \omega)$ that has an expected value shown in Figure 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 $5 \times 5$ pattern, each with $h = 1/360$ giving 5184 equal sized elements per subdomain. The total number of elements in the mesh is 129600. We choose continuous quadratic mortars and set $H \approx \sqrt{h}$, giving 17 mortar degrees of freedom per subdomain boundary. 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 Figure 4(b), we plot the pressure field corresponding to the mean of the permeability.



(a) Mean of the permeability field.

(b) Calculated pressure corresponding to the mean permeability.

Fig. 4. Mean permeability and corresponding pressure solution for Example 7.3.

Our goal is to compare the performance of the following strategies for the solution of the series of interface problems:

- Method 1: standard iterative approach without preconditioning,
- Method 2: multiscale approach with the multiscale basis recomputed for each realization,
- Method 3: multiscale basis preconditioner with the multiscale basis computed using the mean permeability.

In Method 1, a subdomain solve is required on each outer interface iteration. In Method 2, instead of solving subdomain problems on each outer 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. In Method 3, the outer iterative procedure requires one subdomain solve per iteration. The preconditioner uses the multiscale basis for the *training permeability*. Each inner preconditioning iteration involves a linear combination of the *training* multiscale basis.

The performance of the multiscale basis preconditioner for each permeability realization depends on the perturbation from the mean permeability as determined by the assumed variance $\sigma_Y$ and the number of terms in the truncated Karhunen-Loéve expansion. We assume correlation lengths $\eta_1 = 0.2$ and $\eta_2 = 0.125$ in the $x_i-$directions. All of the following numerical experiments use GMRES for the outer iterations with a tolerance of $10^{-6}$. The action of the preconditioner is computed using a block Jacobi preconditioned GMRES algorithm with a coarse scale correction (see [1,17]) with a tolerance of $10^{-8}$.

In the tables below we report some statistics for the number of solves per subdomain: the total number of solves for all realizations, the average number of solves per realization, as well as the minimum and maximum number of solves across the realizations. In the numbers for Method 3 we do not report the number of solves needed to construct the preconditioner, which in this case is 68. Therefore, for a fair comparison, the number 68 should be added to the total number of solves for Method 3. In all cases, the resulting number is still significantly smaller than those for Method 1 and Method 2. Furthermore, the relative cost of the preconditioner decreases when the number of realizations is increased.

In the first experiment we take 4 terms in the KL expansion and 16 collocation points (2 per stochastic dimension) corresponding to the roots of the Hermite polynomials for second order stochastic collocation. In Table 4, we give the results for $\sigma_Y = 0.25$, i.e., the perturbation of the permeability field in each realization is relatively small.

For Method 1 and Method 3, the number of solves equals the number of outer interface iterations. The multiscale basis preconditioner performs very well and reduces significantly the number of outer iterations. For Method 2, the number of solves is the cost for computing the multiscale basis for each realization; hence the number of solves does not depend on the number of outer iterations and stays constant over all realizations. The multiscale basis preconditioner outperforms the other methods by factors of approximately 16 and 40. In Tables 5 and 6, we give the results for $\sigma_Y = 1$ and $\sigma_Y = 10$. As we increase $\sigma_Y$ the average performance of the multiscale basis preconditioner degrades slightly, but it is still significantly more efficient than the other two

| Method | Total | Average | Minimum | Maximum |
|---|---|---|---|---|
| 1 | 2825 | 176.6 | 165 | 186 |
| 2 | 1088 | 68 | 68 | 68 |
| 3 | 64 | 4 | 4 | 4 |

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.

methods.

| Method | Total | Average | Minimum | Maximum |
|---|---|---|---|---|
| 1 | 2960 | 185 | 169 | 195 |
| 2 | 1088 | 68 | 68 | 68 |
| 3 | 84 | 5.25 | 4 | 6 |

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 |
|---|---|---|---|---|
| 1 | 3827 | 239.2 | 195 | 285 |
| 2 | 1088 | 68 | 68 | 68 |
| 3 | 216 | 13.3 | 7 | 19 |

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.

In the second set of experiments, we set $\sigma_Y = 1.0$ and increase the number of terms in the KL expansion. In Figure 5, we plot four random realizations with 100 terms in the KL expansion. First, we compare the number of subdomain solves for each of the three methods using 9 terms in the KL expansion. As

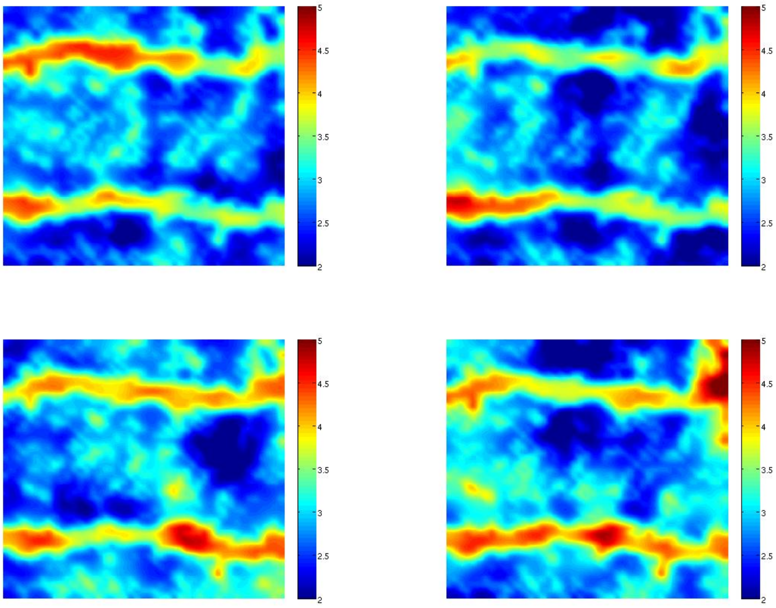


Fig. 5. Four stochastic realizations for Example 7.2 with 100 terms in the KL expansion.

in the previous cases, two stochastic collocation points were chosen in each stochastic dimension, giving 512 realizations. In Table 7, we see that reusing the multiscale basis as a preconditioner results in an order of magnitude fewer solves per subdomains than recomputing the multiscale basis for each realization, and in even larger savings compared to the standard iterative approach. Note also that the cost of constructing the preconditioner, 68 solves per subdomain, is less significant in this case, compared to the previous cases with 16 realizations.

| Method | Total | Average | Minimum | Maximum |
|---|---|---|---|---|
| 1 | 98544 | 192.5 | 160 | 225 |
| 2 | 34816 | 68 | 68 | 68 |
| 3 | 3463 | 6.8 | 5 | 10 |

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.

Next, we increase the number of terms in the KL expansion to 100. Since the tensor product stochastic collocation suffers from the *curse of dimensionality*, requiring $2^{100}$ realizations, we use a Monte Carlo sampling technique to generate 1000 realizations. The results are 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 three methods. We emphasize that while decreasing the subdomain mesh size, increasing the number of subdomains, or refining the mortar mesh would increase the number of solves per subdomain for methods 1 and 2, the number of solves per subdomain for method 3 would remain fixed.

| Method | Total | Average | Minimum | Maximum |
|---|---|---|---|---|
| 1 | 203587 | 203.6 | 144 | 283 |
| 2 | 68000 | 68 | 68 | 68 |
| 3 | 11262 | 11.3 | 6 | 33 |

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.

In summary, the numerical results confirm that, for a suitable *training permeability*, 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 very robust with respect to the variance of the stochastic permeability and the dimension of the stochastic space.

**Acknowledgments.** 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] Yves Achdou, Yvon Maday, and Olof Widlund. Iterative substructuring preconditioners for mortar element methods in two dimensions. *SIAM J. Numer. Anal.*, 36(2):551–580, 1999.

[2] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov. Mixed finite element methods on nonmatching grid blocks. *SIAM J. Numer. Anal.*, 37:1295–1315, 2000.

[3] Todd Arbogast, Gergina Pencheva, Mary F. Wheeler, and Ivan Yotov. A multiscale mortar mixed finite element method. *Multiscale Model. Simul.*, 6(1):319–346, 2007.

[4] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034 (electronic), 2007.

[5] I. Babuska, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic differential equations. *SIAM J. Numer. Anal.*, 42(2):800–825, 2004.

[6] Petter E Bjorstad and Olof B Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1097–1120, 1986.

[7] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring. i. *Math. Comput.*, 47(175):103–134, 1986.

[8] Franco Brezzi, Jim Douglas, Jr., Ricardo Durán, and Michel Fortin. Mixed finite elements for second order elliptic problems in three variables. *Numer. Math.*, 51(2):237–250, 1987.

[9] Franco Brezzi, Jim Douglas, Jr., Michel Fortin, and L. Donatella Marini. Efficient rectangular mixed finite elements in two and three space variables. *RAIRO Modél. Math. Anal. Numér.*, 21(4):581–604, 1987.

[10] Franco Brezzi, Jim Douglas, Jr., and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numer. Math.*, 47(2):217–235, 1985.

[11] Franco Brezzi and Michel Fortin. *Mixed and hybrid finite element methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991.

[12] M. Chen, D. Zhang, A. Keller, and Z. Lu. Stochastic analysis of two phase flow in heterogeneous media by combining Karhunen-Loeve expansion and perturbation method. *Water Resour. Res.*, 41(1):W01006(1–14), 2005.

[13] L. C. Cowsar, J. Mandel, and M. F. Wheeler. Balancing domain decomposition for mixed finite elements. *Math. Comp.*, 64:989–1015, 1995.

[14] M.K. Deb, I.M. Babuška, and J.T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Comput. Methods Appl. Mech. Engrg*, 190(48):6359–6372, 2001.

[15] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation methods for stochastic natural convection problems. *Journal of Computational Physics*, 225:652–685, 2007.

[16] Benjamin Ganis, Hector Klie, Mary F. Wheeler, Tim Wildey, Ivan Yotov, and Dongxiao Zhang. Stochastic collocation and mixed finite elements for flow in porous media. *Comput. Methods Appl. Mech. Engrg*, 197(43-44):3547 – 3559, 2008. Stochastic Modeling of Multiscale and Multiphysics Problems.

[17] Benjamin Ganis, Mary F. Wheeler, Tim Wildey, and Ivan Yotov. Computational efficiency of the multiscale mortar mixed finite element approach to domain decomposition. In preparation, 2009.

[18] Benjamin Ganis and Ivan Yotov. A stochastic mortar mixed finite element method for flow in porous media with multiple rock types. In preparation.

[19] Benjamin Ganis and Ivan Yotov. Implementation of a mortar mixed finite element method using a multiscale flux basis. Technical Report TR-MATH 09-12, Department of Mathematics, University of Pittsburgh, 2009.

[20] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach.* Springer-Verlag, New York, 1991.

[21] R. Glowinski and M. F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 144–172. SIAM, 1988. Philadelphia.

[22] H. Li and D. Zhang. Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resour. Res.*, 43:W09409, doi:10.1029/2006WR005673, 2007.

[23] Jan Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[24] Jan Mandel, Marina Vidrascu, and Patrick Le Tallec. Neumann-Neumann domain decomposition algorithms for solving 2D elliptic problems with nonmatching grids. *SIAM J. Numer. Anal*, 35(35):836–867, 1998.

[25] J.-C. Nédélec. Mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 35(3):315–341, 1980.

[26] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[27] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[28] B. Oksendal. *Stochastic Differential Equations: An Introduction with Applications.* Springer-Verlag, Berlin, 1998.

[29] G. Pencheva and I. Yotov. Balancing domain decomposition for mortar mixed finite element methods on non-matching grids. *Numer. Linear Algebra Appl.*, 10:159–180, 2003.

[30] Alfio Quarteroni and Alberto Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, UK, 2005.

[31] P.-A. Raviart and 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)*, pages 292–315. Lecture Notes in Math., Vol. 606. Springer, Berlin, 1977.

[32] R. A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics*, volume 606, pages 292–315. Springer-Verlag, New York, 1977.

[33] Youcef Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.

[34] C. Schwab and R. Todor. Karhunen–Loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217:100–122, 2006.

[35] B. F. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition.* Cambridge University Press, UK, 1996.

[36] R. A. Todor and C. Schwab. Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients. *IMA Journal of Numerical Analysis*, 27:232–261, 2007.

[37] X. Wan and G.E. Karniadakis. Beyond Weiner-Askey expansions: Handling arbitrary pdfs. *J. of Scientific Computing*, 27(1-3):455–464, 2006.

[38] Mary F. Wheeler and Ivan Yotov. Multigrid on the interface for mortar mixed finite element methods for elliptic problems. *Comput. Methods Appl. Mech. Engrg*, 184:287–302, 2000.

[39] D. Xiu. Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys.*, 2(2):293–309, 2007.

[40] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139 (electronic), 2005.

[41] 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.

[42] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties.* Academic Press, San Diego, Calif., 2002.

[43] D. Zhang and Z. Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions. *Journal of Computational Physics*, 194(2):773–794, 2004.