SIAM J. SCI. COMPUT.
Vol. 33, No. 3, pp. 1439–1474
© 2011 Society for Industrial and Applied Mathematics

A STOCHASTIC MORTAR MIXED FINITE ELEMENT METHOD FOR FLOW IN POROUS MEDIA WITH MULTIPLE ROCK TYPES*

BENJAMIN GANIS†, IVAN YOTOV‡, AND MING ZHONG‡

Abstract. This paper presents an efficient multiscale stochastic framework for uncertainty quantification in modeling of flow through porous media with multiple rock types. The governing equations are based on Darcy's law with nonstationary stochastic permeability represented as a sum of local Karhunen–Loève expansions. The approximation uses stochastic collocation on either a tensor product or a sparse grid, coupled with a domain decomposition algorithm known as the multiscale mortar mixed finite element method. The latter method requires solving a coarse scale mortar interface problem via an iterative procedure. The traditional implementation requires the solution of local fine scale linear systems on each iteration. We employ a recently developed modification of this method that precomputes a multiscale flux basis to avoid the need for subdomain solves on each iteration. In the stochastic setting, the basis is further reused over multiple realizations, leading to collocation algorithms that are more efficient than the traditional implementation by orders of magnitude. Error analysis and numerical experiments are presented.

Key words. uncertainty quantification, stochastic collocation, multiscale basis, mortar finite element, mixed finite element, porous media flow, Smolyak sparse grid

AMS subject classifications. 65C30, 65N30, 65N55, 76S05

DOI. 10.1137/100790689

1. Introduction.

Accurately predicting physical phenomena often involves incorporating uncertainties into a model's input, due to both natural randomness and incomplete knowledge of various physical properties, and then following those uncertainties into the model's output. In this paper we simulate single-phase flow through porous media, by modeling the permeability as a spatially random function. As a result, the equations governing the flow are stochastic. The goal is uncertainty quantification (UQ) via the computation of the expectation and variance of the stochastic solution. The expectation gives mean solution and the variance gives measure of uncertainty. Both of these quantities are of interest in stochastic porous media simulation, and higher order moments can also be approximated with our approach. To compute these statistical moments, we employ the stochastic collocation method [8, 39, 28, 19] coupled with the multiscale mortar mixed finite element method (MMMFEM) [6] implemented with a multiscale flux basis [20].

Stochastic modeling methods can be classified into three groups: (1) sampling methods [16], (2) moment or perturbation methods [41], and (3) nonperturbative methods based on either polynomial chaos expansions [40] or stochastic finite elements [14, 21]. A brief survey of these methods can be found in [34], where an extensive reference list is given. In this order, these methods range from being nonintrusive to very intrusive in terms of modifications to the deterministic model. The stochastic

*Submitted to the journal's Computational Methods in Science and Engineering section March 31, 2010; accepted for publication (in revised form) May 9, 2011; published electronically June 21, 2011. This work was partially supported by the NSF grants DMS 0620402 and DMS 0813901 and the DOE grant DE-FG02-04ER25618.

http://www.siam.org/journals/sisc/33-3/79068.html

†Center for Subsurface Modeling, The Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, Austin, TX 78712 (bganis@ices.utexas.edu).

‡Department of Mathematics, University of Pittsburgh, 301 Thackeray Hall, Pittsburgh, PA 15260 (yotov@math.pitt.edu, miz17@pitt.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

collocation method is a member of the first category along with the well-known Monte Carlo (MC) method [16]. Whereas MC simulations require generating a large number of realizations at random points in the stochastic event space, the stochastic collocation method instead performs realizations at specifically chosen collocation points. This technique obtains better accuracy than MC with fewer realizations. While the nonsampling methods such as moment or perturbation and stochastic finite elements are known to be highly accurate, in practice they are only suitable for systems with relatively small dimensions of random inputs. Their intrusive character complicates implementation, and the resulting large coupled systems may be difficult to parallelize. Conversely, sampling methods generate systems of the same size as their deterministic equivalents that are completely decoupled from each other and hence very easy to parallelize.

In our model, the mean removed log permeability function is parameterized using independent identically distributed random variables in a truncated Karhunen–Loève (KL) expansion. The eigenvalues and eigenfunctions forming this series are computed from a given covariance relationship in which statistical properties such as variance and correlation lengths are assumed to be experimentally determined. Using KL expansion for stochastic permeability is a common approach as seen in [41, 42, 38, 27].

This work builds upon the framework for stochastic collocation and mixed finite elements that was developed in [19]. There, the porous media were assumed to be stationary, meaning that the statistical properties of the permeability were assumed to be constant throughout the domain. In this work we follow [27] (see also [38] for a related perturbation-based approach) in extending this framework to allow nonstationary porous media with different covariance functions for different parts of the domain. These statistically independent zones are used to represent multiple rock types, motivated by geologic features such as stratification. We shall refer to these zones as *KL regions*. In this framework for nonstationary porous media, the covariance between any two points within a single KL region depends on their distance only, but the covariance between any two points that lie in different KL regions is zero; i.e., they are uncorrelated.

In porous media problems, resolving fine scale accuracy is oftentimes computationally infeasible, necessitating multiscale approximations, such as the variational multiscale method [25, 4] and multiscale finite elements [24, 11, 2]. Both have been applied to stochastic problems in [7, 18] and [15, 1], respectively.

This paper employs for each stochastic realization the MMMFEM [6], with the recently proposed multiscale flux basis implementation [20]. As a mixed method, it provides accurate approximation of both pressure and velocity and elementwise conservation of mass, which are advantageous properties for porous media flow. The MMMFEM uses nonoverlapping domain decomposition to break up the physical domain into *subdomains* controlled by separate computer processors, giving a natural parallelization within a fixed realization, thereby enabling UQ for very large problems.[1] Within each subdomain, there is a fine scale discretization that may be spatially nonconforming to its neighboring subdomains. On subdomain interfaces, a coarse scale mortar discretization is used to impose weak continuity of the discrete normal velocities. Using these varying scales, the global fine scale problem is reduced

[1] It should be noted that one of the benefits of nonintrusive UQ techniques is the "embarassingly parallel" nature of stochastic sampling. It is entirely possible to compute several simultaneous realizations in parallel, while also utilizing the parallelization in the MMMFEM's domain decomposition, but this is not considered in this work.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

to a coarse scale interface problem and solved in parallel using an iterative method. We present error analysis for the stochastic multiscale approximation of the pressure and the velocity. We refer the reader to [26] for work on overlapping Schwarz domain decomposition methods for stochastic partial differential equations.

Notice that the physical domain has two decompositions: KL regions for the statistical representation of the nonstationary random permeability, and subdomains for the domain decomposition of the MMMFEM. The former is a physical decomposition depending on geologic structure, and the latter is a computational decomposition depending on available computing resources. We specifically choose to have the subdomains conform to the KL regions so that each processor only deals with a single KL expansion, which leads to a more efficient algorithm. We retain proper scalability because within each KL region, we allow multiple subdomains. Hence, the number of KL regions N_Ω is less than or equal to the number of subdomains N_D, and each KL region can be expressed as a union of one or more disjoint subdomains. This approach allows for utilizing more processors than the physically dependent number of KL regions.

In a deterministic setting, the traditional implementation of the interface iteration in the MMMFEM requires solving one Dirichlet-to-Neumann problem on each subdomain (a linear system) for each interface iteration. Solving these subdomain problems is the dominant computational cost of the MMMFEM, and therefore this cost worsens with the condition number of the problem. In [20], a new approach was proposed called multiscale flux basis implementation, in which one subdomain problem is solved for each mortar degree of freedom before the interface iteration begins. The solutions to these problems form a basis of flux responses containing all the necessary information to solve the subdomain problem. The computational cost in forming the basis is a fixed and controllable quantity, and therefore does not worsen with the condition number of the problem. Linear combinations of multiscale basis functions are used during the interface iteration so that no additional subdomain problems are required, except for one or more additional solves to recover the local fine scale information at the completion of the iteration. Therefore the multiscale flux basis implementation is more efficient in cases where the number of interface iterations strictly exceeds the number of mortar degrees of freedom per subdomain. This gain in computational efficiency increases with the number of subdomains.

In this paper we propose possible ways that extend the concept of a multiscale flux basis to the stochastic flow problem, where the permeability is a nonstationary random field. To this end, we investigate three algorithms that combine stochastic collocation and the MMMFEM with varying degrees of the multiscale flux basis implementation. The first collocation algorithm uses the MMMFEM with its traditional implementation, requiring solving one subdomain problem per interface iteration, on every stochastic realization. The second collocation algorithm forms a *deterministic* multiscale basis to solve the MMMFEM on each stochastic realization. These bases are then discarded and then recomputed with new permeability data for each subsequent realization. The third collocation algorithm forms a full *stochastic* multiscale basis across all local realizations, containing all the necessary information to perform the collocation before it begins. With extra "bookkeeping" in the nonstationary case, we can take advantage of the repeated local structure of the permeability realizations in both tensor and sparse stochastic collocation. In particular, the multiscale flux basis for a fixed subdomain and a fixed realization is reused a number of times during the stochastic collocation process for all cases with the same local permeability realization. This increases substantially the gain in computational efficiency from the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 1
Constants used throughout this paper.

Constant	Represents number of...
N_D	Subdomains
N_Ω	KL regions
N_{term}	Total stochastic dimensions
$N_{\text{term}}(i)$	Stochastic dimensions in KL region i
$N_{\text{coll}}(i,j)$	1-D tensor product collocation points in dimension j of KL region i
$N_{\text{real}}(i)$	Local permeability realizations in KL region i
N_{real}	Global permeability realizations
$N_{\text{dof}}(i)$	Mortar degrees of freedom on subdomain i
$N_{\text{iter}}(i)$	Conjugate gradient iterations for global collocation index i

multiscale flux basis. We refer the reader to [37] for the use of the multiscale flux basis as a preconditioner, which provides a different approach for its reuse over many stochastic realizations and can lead to even larger speedup.

The resulting collocation algorithms are more computationally efficient than the traditional implementation by orders of magnitude. By limiting the number of subdomain solves via the computation of deterministic or stochastic multiscale bases, we demonstrate that we can lessen the burden of the curse of dimensionality in the stochastic collocation method. We present a number of computational experiments that confirm the above statement. Some of the examples show how a posteriori error estimation and adaptivity for the MMMFEM can be employed in stochastic multiscale simulations. We also present numerical convergence studies that confirm the theoretical a priori error estimates.

1.1. Notation. Let $D \subset \mathbb{R}^d$ ($d = 2$ or 3) denote the physical domain. It is bounded, with Lipschitz boundary ∂D and outer unit normal $\mathbf{n}$. Let Ω denote the stochastic event space with probability measure P. The expectation and variance of a random variable $\xi(\omega) : \Omega \to \mathbb{R}$ with a probability density function (PDF) $\rho(y)$ are denoted by

$$E[\xi] = \int_\Omega \xi(\omega) dP(\omega) = \int_{\mathbb{R}} y\rho(y) dy \quad \text{and} \quad \text{var}[\xi] = E[\xi^2] - (E[\xi])^2. \tag{1}$$

In the following, C denotes a generic positive constant independent of the discretization parameters h and H. For a domain $G \subset \mathbb{R}^d$, the $L^2(G)$ inner product and norm for scalar and vector valued functions are denoted $(\cdot,\cdot)_G$ and $\|\cdot\|_G$, respectively. We omit G in the subscript if $G = D$. The norm in the Sobolev space $H^s(G)$ will be denoted by $\|\cdot\|_{s,G}$. For a section of the domain or element boundary $S \subset \mathbb{R}^{d-1}$ we write $\langle\cdot,\cdot\rangle_S$ and $\|\cdot\|_S$ for the $L^2(S)$ inner product (or duality pairing) and norm, respectively. Dual spaces are denoted by $(\cdot)^*$. Constants that are frequently used throughout this paper are given in Table 1.

The rest of the paper is organized as follows. In section 2 we present the stochastic model problem and its domain decomposition variational formulations. The stochastic multiscale discretization based on stochastic collocation and the MMMFEM is described in section 3. The error analysis of the method is given in section 4. In section 5 we discuss three different algorithms that can be used to solve the fully discrete problem. A number of computational examples are presented in section 6.

2. Model problem. We consider Darcy's law for steady-state, single-phase, incompressible flow through a saturated porous medium in physical domain D. Let $\partial D = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. Let the permeability K be a stochastic function

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

defined on $D \times \Omega$. The Darcy velocity $\mathbf{u}$ and the pressure p are stochastic functions that satisfy P-almost everywhere $\omega \in \Omega$,

$$\nabla \cdot \mathbf{u} = f, \qquad \text{in } D, \tag{2a}$$

$$\mathbf{u} = -K(\mathbf{x}, \omega)\nabla p, \qquad \text{in } D, \tag{2b}$$

$$p = g_D, \qquad \text{on } \Gamma_D, \tag{2c}$$

$$\mathbf{u} \cdot \mathbf{n} = g_N, \qquad \text{on } \Gamma_N. \tag{2d}$$

We assume that $f(\mathbf{x}) \in L^2(D)$, $g_D(\mathbf{x}) \in H^{1/2}(\Gamma_D)$, and $g_N(\mathbf{x}) \in L^2(\Gamma_N)$ are deterministic functions. The permeability $K(\mathbf{x}, \omega)$ is either a scalar or diagonal 2-tensor, which, for P-almost every $\omega \in \Omega$, is uniformly positive definite with components in $L^\infty(D)$.

In order to guarantee positive permeability almost surely in Ω, we consider its logarithm $Y = \ln(K)$. Let the mean removed log permeability be denoted by Y', so that

$$Y(\mathbf{x}, \omega) = E[Y](\mathbf{x}) + Y'(\mathbf{x}, \omega).$$

Following [27], let D be a union of disjoint KL regions, $\overline{D} = \cup_{i=1}^{N_\Omega} \overline{D}_{KL}^{(i)}$. Strictly within each KL region, the porous medium is statistically stationary, meaning covariance between any two points depends only on their distance and not on their location. The covariance between any two points from different regions is zero. Therefore the medium is globally nonstationary. As a result the probability space Ω is a product of N_Ω spaces $\Omega^{(i)}$. For each event $\omega \in \Omega$,

$$\omega = (\omega^{(1)}, \ldots, \omega^{(N_\Omega)}) \quad \text{and} \quad Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} Y^{(i)}(\mathbf{x}, \omega^{(i)}),$$

where $Y^{(i)}(\mathbf{x}, \omega^{(i)})$ has physical support in $D_{KL}^{(i)}$.

2.1. Karhunen–Loève (KL) expansion. Each $Y^{(i)}$ is assumed to be colored noise, for which we are given a covariance function. These are symmetric and positive definite, so they can be decomposed into series expansions

$$C_{Y^{(i)}}(\mathbf{x}, \bar{\mathbf{x}}) = E[Y^{(i)}(\mathbf{x}, \omega^{(i)}) Y^{(i)}(\bar{\mathbf{x}}, \omega^{(i)})] = \sum_{j=1}^{\infty} \lambda_j^{(i)} f_j^{(i)}(\mathbf{x}) f_j^{(i)}(\bar{\mathbf{x}}).$$

The eigenvalues $\lambda_j^{(i)}$ and eigenfunctions $f_j^{(i)}$ are computed by solving the Fredholm integral equations

$$\int_{D_{KL}^{(i)}} C_{Y^{(i)}}(\mathbf{x}, \bar{\mathbf{x}}) f_j^{(i)}(\mathbf{x}) d\mathbf{x} = \lambda_j^{(i)} f_j^{(i)}(\bar{\mathbf{x}}). \tag{3}$$

Since the $C_{Y^{(i)}}$ are symmetric and positive definite, the eigenfunctions are mutually orthogonal and form a complete spanning set. Using these facts, the Karhunen–Loève expansion for the log permeability can be exactly written as

$$Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} \sum_{j=1}^{\infty} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}), \tag{4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where the eigenfunctions $f_j^{(i)}(\mathbf{x})$ computed in (3) have been extended by zero outside of $D_{KL}^{(i)}$ and $\xi_j^{(i)} : \Omega_i \to \mathbb{R}$ are independent identically distributed random variables [21]. In our work we assume $Y^{(i)}$ are Gaussian processes, so each $\xi_j^{(i)}$ is a normal random variable with zero mean and unit variance, having PDF $\rho_j^{(i)}(y) = 1/\sqrt{2\pi}\exp[-y^2/2]$.

As is typically done at this point, we commit a modeling error that replaces the stochastic problem by a higher dimensional deterministic approximation. This is feasible to do as the eigenvalues $\lambda_j^{(i)}$ typically decay rapidly [42]. It is beyond the scope of this paper to address the modeling error associated with truncating the KL expansion. Some work has been done to quantify the modeling error [28] and it can be reduced a posteriori [10].

Assumption 2.1 (finite dimensional noise assumption). Each KL expansion $Y^{(i)}$ is truncated after $N_{\text{term}}(i)$ terms, which allows us to approximate (4) by

$$Y'(\mathbf{x},\omega) \approx \sum_{i=1}^{N_\Omega} \sum_{j=1}^{N_{\text{term}}(i)} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}). \tag{5}$$

Globally, this means that we have $N_{\text{term}} = \sum N_{\text{term}}(i)$ terms in Y'. A low number of terms leads to a smooth permeability in a KL region. Therefore to model very heterogeneous noise in a KL region, $N_{\text{term}}(i)$ should be increased. The images of the random variables $\mathbb{S}_j^{(i)} = \xi_j^{(i)}(\Omega^{(i)})$ make up the finite dimensional vector spaces

$$\mathbb{S}^{(i)} = \prod_{j=1}^{N_{\text{term}}(i)} \mathbb{S}_j^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}(i)} \quad \text{and} \quad \mathbb{S} = \prod_{i=1}^{N_\Omega} \mathbb{S}^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}},$$

which are local to each KL region and global, respectively.

To simplify notation, we shall introduce a function κ that provides a natural ordering for the global number of stochastic dimensions. Let the jth stochastic parameter of the ith KL region have a global index in $\{1,\dots,N_{\text{term}}\}$ by the function

$$\kappa(i,j) = \begin{cases} j, & \text{if } i = 1, \\ j + \displaystyle\sum_{k=1}^{i-1} N_{\text{term}}(k), & \text{if } i > 1. \end{cases}$$

For example, the random vector $\xi = (\xi_j^{(i)})_{1\le\kappa(i,j)\le N_{\text{term}}} = (\xi_j^{(i)})_\kappa$ is by definition equal to

$$\left(\underbrace{\xi_1^{(1)},\dots,\xi_{N_{\text{term}}(1)}^{(1)}}_{\text{KL region 1}}, \underbrace{\xi_1^{(2)},\dots,\xi_{N_{\text{term}}(2)}^{(2)}}_{\text{KL region 2}}, \ \dots, \ \xi_j^{(i)}, \dots, \underbrace{\xi_1^{(N_\Omega)},\dots,\xi_{N_{\text{term}}(N_\Omega)}^{(N_\Omega)}}_{\text{KL region } N_\Omega} \right).$$

If $\rho_j^{(i)}$ is the PDF of each $\xi_j^{(i)}$, then joint PDF for ξ is defined to be $\rho = \prod_i \prod_j \rho_j^{(i)}$. Then we can write $Y(\mathbf{x},\omega) \approx Y(\mathbf{x},\mathbf{y})$, where $\mathbf{y} = (\xi_j^{(i)}(\omega^{(i)}))_\kappa$.

For the remainder of this paper, we abuse notation by replacing $K(\mathbf{x},\omega)$ with its finite dimensional spectral approximation $K(\mathbf{x},\mathbf{y})$ given by (5). We also identify

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

each stochastic subspace $\Omega^{(i)}$ with its parameterization $\mathbb{S}^{(i)}$. Therefore the modeling error between the true stochastic solution and its finite dimensional approximation $\|\mathbf{u}(\mathbf{x},\omega) - \mathbf{u}(\mathbf{x},\mathbf{y})\|$ is neglected.

2.2. Domain decomposition. We use the domain decomposition approach described in [22] to restrict the model problem into nonoverlapping subdomains D_i, $i = 1,\dots,N_D$, $\overline{D} = \bigcup_{i=1}^{N_D} \overline{D_i}$, and $D_i \cap D_j = \emptyset$ for $i \neq j$. They may be spatially nonconforming, but conform to the KL regions. Denote the interface between subdomains D_i and D_j by $\Gamma_{i,j} = \partial D_i \cap \partial D_j$, the union of all interfaces that touch subdomain D_i by $\Gamma_i = \partial D_i \setminus \partial D$, and the union of all interfaces by $\Gamma = \bigcup_{i\neq j} \Gamma_{i,j}$. The domain decomposition can be viewed as a coarse grid on D. Note that subdomains may be different from KL regions. We assume that each KL region is a union of subdomains.

System (2) holds within each subdomain D_i, but additionally the pressure and the normal velocity components must remain physically continuous across the interfaces. Equivalently, we seek $(\mathbf{u}_i, p_i)$ such that for $i = 1,\dots,N_D$ and for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\nabla \cdot \mathbf{u}_i = f \qquad \text{in } D_i, \tag{6a}$$

$$\mathbf{u}_i = -K(\mathbf{x},\mathbf{y})\nabla p_i \qquad \text{in } D_i, \tag{6b}$$

$$p_i = g_D \qquad \text{on } \partial D_i \cap \Gamma_D, \tag{6c}$$

$$\mathbf{u}_i \cdot \mathbf{n} = g_N \qquad \text{on } \partial D_i \cap \Gamma_N, \tag{6d}$$

$$p_i = p_j \qquad \text{on } \Gamma_{i,j}, \quad i \neq j, \tag{6e}$$

$$\mathbf{u}_i \cdot \mathbf{n}_i + \mathbf{u}_j \cdot \mathbf{n}_j = 0 \qquad \text{on } \Gamma_{i,j}, \quad i \neq j, \tag{6f}$$

where $\mathbf{n}_i$ is the outer unit normal to ∂D_i.

2.3. Variational formulation. In the physical dimensions, define the space $\mathbf{V}_i(D_i) = H(div; D_i) = \{\mathbf{v} \in (L^2(D_i))^d \mid \nabla \cdot \mathbf{v} \in L^2(D_i)\}$. Then the deterministic spaces for $i = 1,\dots,N_D$ are

$$W_i(D_i) = L^2(D_i), \qquad \mathbf{V}_i^{\gamma}(D_i) = \{\mathbf{v} \in H(\mathrm{div}; D_i) \mid \mathbf{v}\cdot\mathbf{n} = \gamma \text{ on } \partial D_i \cap \Gamma_N\},$$

$$\text{and globally:} \qquad W(D) = \bigoplus_{i=1}^{N_D} W_i(D_i), \qquad \mathbf{V}^{\gamma}(D) = \bigoplus_{i=1}^{N_D} \mathbf{V}_i^{\gamma}(D_i),$$

where $\gamma \in L^2(\Gamma_N)$.[2] The global velocity space $\mathbf{V}^{\gamma}(D)$ is not continuous in the normal direction across subdomain interfaces Γ, so it is not a subset of $H(\mathrm{div}; D)$. To account for this, we introduce a Lagrange multiplier space that has a physical meaning of pressure and is used to weakly impose continuity of the normal velocities:

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

In the above, $(\cdot)^*$ denotes the dual space. Since our goal is to compute statistical moments, we define the space

$$L^2_{\rho}(\mathbb{S}) = \left\{ \mathbf{v} : \mathbb{S} \to \mathbb{R}^d \mid \left(\int_{\mathbb{S}} \|\mathbf{v}(\mathbf{y})\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} < \infty \right\},$$

[2] Note that the condition $\mathbf{v}\cdot\mathbf{n} = \gamma$ requires slightly higher regularity than the usual for normal traces of functions in $H(\mathrm{div}; D)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

and take its tensor product with the aforementioned deterministic spaces to form the stochastic spaces

$$W(D,\mathbb{S}) = W(D)\otimes L^2_\rho(\mathbb{S}), \quad \mathbf{V}^\gamma(D,\mathbb{S}) = \mathbf{V}^\gamma(D)\otimes L^2_\rho(\mathbb{S}), \quad M(\Gamma,\mathbb{S}) = M(\Gamma)\otimes L^2_\rho(\mathbb{S}).$$

Whenever the explicit dependence in parentheses is omitted, it is implied that we mean the stochastic spaces, e.g., $W = W(D,\mathbb{S})$. We equip the stochastic pressure and velocity spaces with the mean norms

$$\begin{aligned}
\|\mathbf{v}\|^2_{V_i} &= \int_{\mathbb{S}} \left(\int_{D_i} \left(\mathbf{v}\cdot\mathbf{v} + (\nabla\cdot\mathbf{v})^2 \right) d\mathbf{x} \right) \rho(\mathbf{y}) d\mathbf{y} = E\left[\|\mathbf{v}\|^2_{H(\mathrm{div};D_i)} \right], \\
\|\mathbf{v}\|^2_{V} &= \sum_{i=1}^{N_D} \|\mathbf{v}\|^2_{V_i}, \\
\|w\|^2_{W} &= \int_{\mathbb{S}} \left(\int_{D} w^2 d\mathbf{x} \right) \rho(\mathbf{y}) d\mathbf{y} = E\left[\|w\|^2 \right].
\end{aligned}$$

Multiplication of system (6a) by appropriate test functions and integration by parts gives the following stochastic dual mixed variational formulation: Find $\mathbf{u} \in \mathbf{V}^{g_N}$, $p \in W$, and $\lambda \in M$ such that for $i = 1, \ldots, N_D$,

$$\begin{aligned}
\int_{\mathbb{S}} (K^{-1}\mathbf{u},\mathbf{v})_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \Big[& (p, \nabla\cdot\mathbf{v})_{D_i} - \langle \mathbf{v}\cdot\mathbf{n}_i, \lambda \rangle_{\Gamma_i} \\
& - \langle \mathbf{v}\cdot\mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \forall \mathbf{v} \in \mathbf{V}^0_i, \qquad \text{(7a)} \\
\int_{\mathbb{S}} (\nabla\cdot\mathbf{u}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} (f, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} \qquad & \forall w \in W_i, \qquad \text{(7b)} \\
\int_{\mathbb{S}} \sum_{i=1}^{N_D} \langle \mathbf{u}_i\cdot\mathbf{n}_i, \mu \rangle_{\Gamma_i} \rho(\mathbf{y}) d\mathbf{y} = 0 \qquad & \forall \mu \in M. \qquad \text{(7c)}
\end{aligned}$$

The extra condition (7c) enforces weakly the flux continuity lost across the interfaces in the domain decomposition.

3. Discretization. We begin with a semidiscrete approximation to the weak solution $(\mathbf{u}, p, \lambda)$ of the stochastic variational formulation (7), based on the MMMFEM in the physical dimensions. This is a multiscale approach that combines a local fine scale discretization within each subdomain with a global coarse scale discretization across subdomain interfaces. We then employ the stochastic collocation method, using a tensor product or sparse grid Gauss–Hermite quadrature rule in the additional stochastic dimensions, to form the fully discrete solution. This nonintrusive approach decouples the $(d + N_{\mathrm{term}})$-dimensional stochastic problem into a sequence of independent d-dimensional deterministic problems, which are realizations in stochastic space and function evaluations in the quadrature rule.

3.1. Finite element approximation. Each subdomain D_i is partitioned into a local d-dimensional quasi-uniform affine mesh $\mathcal{T}_{h,i}$. The faces (or edges) of these meshes are spatially conforming within each subdomain, but are allowed to be nonconforming along subdomain interfaces. Let the maximal element diameter of $\mathcal{T}_{h,i}$ be h_i, and let the global characteristic fine scale diameter be $h = \max_{i=1}^{N_D} h_i$. Denote the global fine mesh by $\mathcal{T}_h = \bigcup_{i=1}^{N_D} \mathcal{T}_{h,i}$. Let $\mathbf{V}_{h,i}(D_i) \times W_{h,i}(D_i) \subset \mathbf{V}_i(D_i) \times W_i(D_i)$ be a mixed finite element space on the mesh $\mathcal{T}_{h,i}$ such that $\mathbf{V}_{h,i}(D)$ contains piecewise

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

polynomials of degree k and $W_{h,i}(D)$ contains piecewise polynomials of degree l. Examples of mixed finite element spaces can be found in [9]. The numerical tests in this paper use the lowest order Raviart–Thomas space [31] on rectangular elements in two dimensions (2-D) and brick elements in three dimensions (3-D). Globally, the discrete pressure and velocity spaces for this method are $W_h(D) = \bigoplus_{i=1}^{N_D} W_{h,i}(D_i)$ and $\mathbf{V}_h(D) = \bigoplus_{i=1}^{N_D} \mathbf{V}_{h,i}(D_i)$. We further define $\mathbf{V}_h^\gamma(D) = \{\mathbf{v} \in \mathbf{V}_h(D) \,|\, \mathbf{v}\cdot\mathbf{n} = Q_h\gamma \text{ on } \Gamma_N\}$, where Q_h is the L^2-projection operator onto the normal trace of the velocity space; see (10).

Each interface $\Gamma_{i,j}$ is partitioned into a coarse $(d-1)$-dimensional quasi-uniform affine mesh denoted $\mathcal{T}_{H,i,j}$. On this mesh we define the mortar space that weakly enforces continuity of normal fluxes for the discrete velocities across the nonmatching grids. Let the maximal element diameter of this coarse mesh be $H_{i,j}$, and let the global characteristic coarse scale diameter be $H = \max_{1\le i<j\le N_D} H_{i,j}$. Denote the global coarse mesh by $\mathcal{T}_H = \bigcup_{1\le i<j\le N_D} \mathcal{T}_{H,i,j}$. Let $M_{H,i,j}(\Gamma_{i,j}) \subset L^2(\Gamma_{i,j})$ be the mortar space containing continuous or discontinuous piecewise polynomials of degree r,s where $r \ge k+1$. Globally, the mortar space for this method is $M_H(\Gamma) = \bigoplus_{1\le i<j\le N_D} M_{H,i,j}(\Gamma_{i,j})$. Notice that this is a nonconforming approximation, as $M_H(\Gamma) \not\subseteq M(\Gamma)$.

Under these finite dimensional subspaces, the semidiscrete stochastic multiscale mortar mixed finite element approximation of (7) is to find $\mathbf{u}_h : \mathbb{S} \to \mathbf{V}_h^{g_N}(D)$, $p_h : \mathbb{S} \to W_h(D)$, and $\lambda_H : \mathbb{S} \to M_H(\Gamma)$ such that for $i = 1, \dots, N_D$ and ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$(K^{-1}\mathbf{u}_h, \mathbf{v})_{D_i} = (p_h, \nabla\cdot\mathbf{v})_{D_i} - \langle \mathbf{v}\cdot\mathbf{n}_i, \lambda_H\rangle_{\Gamma_i}$$

$$- \langle \mathbf{v}\cdot\mathbf{n}_i, g_D\rangle_{\partial D_i\cap\Gamma_D} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0(D_i), \tag{8a}$$

$$(\nabla\cdot\mathbf{u}_h, w)_{D_i} = (f, w)_{D_i} \qquad \forall w \in W_{h,i}(D_i), \tag{8b}$$

$$\sum_{i=1}^{N_D} \langle \mathbf{u}_{h,i}\cdot\mathbf{n}_i, \mu\rangle_{\Gamma_i} = 0 \qquad \forall \mu \in M_H(\Gamma). \tag{8c}$$

In this formulation the pressure continuity (6e) is modeled via the mortar pressure function λ_H, while the flux continuity (6f) is imposed weakly on the coarse scale via (8c). For the above method to be well-posed, the two scales must be chosen such that the mortar space is not too rich compared to the normal traces of the subdomain velocity spaces.

Assumption 3.1. Assume there exists a constant C independent of h and H such that

$$\|\mu\|_{\Gamma_{i,j}} \le C(\|\mathcal{Q}_{h,i}\mu\|_{\Gamma_{i,j}} + \|\mathcal{Q}_{h,j}\mu\|_{\Gamma_{i,j}}) \quad \forall \mu \in M_H(\Gamma), \quad 1 \le i < j \le N_D, \tag{9}$$

where $\mathcal{Q}_{h,i} : L^2(\Gamma_i) \to \mathbf{V}_{h,i}\cdot\mathbf{n}_i|_{\Gamma_i}$ is the L^2-projection operator onto the normal trace of the velocity space on subdomain i, i.e., for any $\phi \in L^2(\Gamma_i)$,

$$\langle \phi - \mathcal{Q}_{h,i}\phi, \mathbf{v}\cdot\mathbf{n}_i\rangle_{\Gamma_i} = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}(D_i). \tag{10}$$

This assumption is easily satisfied in practice and controls the mortar degrees of freedom by the degrees of freedom on the traces of subdomain velocities. This is even less restrictive in the multiscale case when the mortar grid is coarser than the subdomain grids. A more specific characterization of grids that satisfy this assumption is given in [29, 6].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

3.2. Stochastic collocation. Let m (or $\mathbf{m}$) be a multi-index indicating the desired polynomial degree of accuracy in the stochastic dimensions. The stochastic collocation method approximates the semidiscrete solution $(\mathbf{u}_h, p_h, \lambda_H)$ by an interpolant $\mathcal{I}_m$ in the stochastic dimensions. It is uniquely formed on a set of N_{real} stochastic points $\{\mathbf{y}_k\}$ that form a Haar set in $\mathbb{S}$, where N_{real} is a function of m. More precisely the fully discrete solution is

$$\mathbf{u}_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m \mathbf{u}_h(\mathbf{x},\mathbf{y}), \quad p_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m p_h(\mathbf{x},\mathbf{y}), \quad \lambda_{H,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m \lambda_H(\mathbf{x},\mathbf{y}).$$

Let $\{L_m^{\{k\}}(\mathbf{y})\}$ be the Lagrange basis satisfying $\{L_m^{\{k\}}(\mathbf{y}_j)\} = \delta_{kj}$. Then the fully discrete solution has the Lagrange representation

$$(\mathbf{u}_{h,m}, p_{h,m}, \lambda_{H,m})(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{N_{\text{real}}} (\mathbf{u}_h^{\{k\}}, p_h^{\{k\}}, \lambda_H^{\{k\}})(\mathbf{x}) L_m^{\{k\}}(\mathbf{y}),$$

where $(\mathbf{u}_h^{\{k\}}, p_h^{\{k\}}, \lambda_H^{\{k\}})$ is the evaluation of semidiscrete solution $(\mathbf{u}_h, p_h, \lambda_H)$ at the point in stochastic space $\mathbf{y}_k$. In other words, for each permeability realization $K^{\{k\}}(\mathbf{x}) = K(\mathbf{x},\mathbf{y}_k)$, $k = 1,\dots,N_{\text{real}}$, we solve the deterministic problem: find $\mathbf{u}_h^{\{k\}} \in \mathbf{V}_h^{g_N}(D)$, $p_h^{\{k\}} \in W_h(D)$, and $\lambda_H^{\{k\}} \in M_H(\Gamma)$ such that for $i = 1,\dots,N_D$,

$$\begin{aligned}
((K^{\{k\}})^{-1}\mathbf{u}_h^{\{k\}}, \mathbf{v})_{D_i} &= (p_h^{\{k\}}, \nabla\cdot\mathbf{v})_{D_i} - \langle \mathbf{v}\cdot\mathbf{n}_i, \lambda_H^{\{k\}}\rangle_{\Gamma_i} \\
&\quad - \langle \mathbf{v}\cdot\mathbf{n}_i, g_D\rangle_{\partial D_i\cap\Gamma_D} && \forall \mathbf{v} \in \mathbf{V}_{h,i}^0(D_i), \qquad \text{(11a)}\\
(\nabla\cdot\mathbf{u}_h^{\{k\}}, w)_{D_i} &= (f,w)_{D_i} && \forall w \in W_{h,i}(D_i), \qquad \text{(11b)}\\
\sum_{i=1}^{N_D} \langle \mathbf{u}_{h,i}^{\{k\}}\cdot\mathbf{n}_i, \mu\rangle_{\Gamma_i} &= 0 && \forall \mu \in M_H(\Gamma). \qquad \text{(11c)}
\end{aligned}$$

The Lagrange representation of the fully discrete solution is plugged into the expectation integral (1) to form a quadrature rule. For example, the pressure expectation is computed by

$$\begin{aligned}
E[p_{h,m}](\mathbf{x}) &= \int_{\mathbb{S}} p_{h,m}(\mathbf{x},\mathbf{y})\rho(\mathbf{y})d\mathbf{y} = \int_{\mathbb{S}} \sum_{k=1}^{N_{\text{real}}} p_h^{\{k\}}(\mathbf{x}) L_m^{\{k\}}(\mathbf{y})\rho(\mathbf{y})d\mathbf{y} \\
&= \sum_{k=1}^{N_{\text{real}}} w_m^{\{k\}} p_h^{\{k\}}(\mathbf{x}),
\end{aligned}$$

where the weights are given by $w_m^{\{k\}} = \int_{\mathbb{S}} L_m^{\{k\}}(\mathbf{y})\rho(\mathbf{y})d\mathbf{y}$.

The choice of collocation points $\{\mathbf{y}_k\}$, i.e., the type of quadrature rule, produces different types of stochastic collocation methods. This paper considers two types of grids: tensor product and sparse grids. Both types of grids are constructed from one-dimensional rules, where the points in dimension $\mathbb{S}_j^{(i)}$ are the zeros of orthogonal polynomials with respect to the $L_\rho^2(\mathbb{S}_j^{(i)})$–inner product. Since we are using Gaussian random variables, we choose the zeros of the "probabilist" $N(0,1)$ Hermite polynomials

$$H_m(y) = m! \sum_{k=0}^{[m/2]} (-1)^k \frac{(2y)^{m-2k}}{k!(m-2k)!}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Denote the sets of one-dimensional weights and abscissae for $H_m(y)$ by

$$\mathcal{W}(m) = \{w_m^1, \ldots, w_m^m\} \quad \text{and} \quad \mathcal{H}(m) = \{h_m^1, \ldots, h_m^m\},$$

and notice that when $m = 2k-1$ is odd, the point h_m^k is the origin. These weights and abscissae can easily be computed with a symbolic manipulation software package. Alternatively, one may convert a table of rules for the "physicist" $N(0, 1/2)$ Hermite polynomials listed in [3] by dividing the weights by a factor of $\sqrt{\pi}$ and multiplying the abscissae by a factor of $\sqrt{2}$.

3.3. Collocation on tensor product grids. In tensor product collocation, the polynomial accuracy is prescribed in terms of *component* degree, i.e., independently in each stochastic dimension. This allows for very easy construction of anisotropic rules, accurate to different polynomial degrees in different stochastic dimensions. Unfortunately, the number of points in tensor product rules grow exponentially with both the polynomial accuracy and the number of dimensions. This is commonly referred to as the "curse of dimensionality." Therefore, this inherently limits their usage to problems with a relatively low number of stochastic dimensions, i.e., about a dozen or less.

If we choose $N_{\text{coll}}(i,j)$ collocation points in stochastic dimension j of KL region i, then $\mathbf{m} = (N_{\text{coll}}(i,j))_\kappa$ is the N_{term}-dimensional multi-index indicating the desired component degree of the interpolant in the stochastic space $\mathbb{S}$. The corresponding anisotropic tensor product Gauss–Hermite interpolant in N_{term} dimensions is defined by

$$\begin{aligned}\mathcal{I}_{\mathbf{m}}^{\text{TG}} f(\mathbf{y}) &= (\mathcal{I}_{\mathbf{m}(1)} \otimes \cdots \otimes \mathcal{I}_{\mathbf{m}(N_{\text{term}})}) f(\mathbf{y}) \\ &= \sum_{k_1=1}^{\mathbf{m}(1)} \cdots \sum_{k_{N_{\text{term}}}=1}^{\mathbf{m}(N_{\text{term}})} f(h_{\mathbf{m}(1)}^{k_1}, \ldots, h_{\mathbf{m}(N_{\text{term}})}^{k_{N_{\text{term}}}}) L_{\mathbf{m}(1)}^{k_1}(y_1) \cdots L_{\mathbf{m}(N_{\text{term}})}^{k_{N_{\text{term}}}}(y_{N_{\text{term}}}).\end{aligned}$$

The set of abscissae for this rule is

$$\mathcal{T}(\mathbf{m}) = \bigotimes_{k=1}^{N_{\text{term}}} \mathcal{H}(\mathbf{m}(k)) = \bigotimes_{i=1}^{N_\Omega} \left(\bigotimes_{j=1}^{N_{\text{term}}(i)} \mathcal{H}(N_{\text{coll}}(i,j)) \right), \tag{12}$$

which interpolates the semidiscrete solution into the polynomial space $\mathbb{P}_{\mathbf{m}} = \prod_k \mathbb{P}_{\mathbf{m}(k)}$ in the stochastic dimensions. The tensor product weight for the point $(h_{\mathbf{m}(1)}^{k_1}, \ldots, h_{\mathbf{m}(N_{\text{term}})}^{k_{N_{\text{term}}}})$ is given by

$$w(\mathbf{k}) = \prod_{i=1}^{N_{\text{term}}} w_{\mathbf{m}(i)}^{k_i}.$$

In a fixed stochastic dimension, the one-dimensional Gauss–Hermite quadrature rules are accurate to degree $2m-1$.

Remark 3.1. By (12), the global N_{term}-dimensional tensor grid is the tensor product of N_Ω smaller tensor product grids of dimension $N_{\text{term}}(i)$. Therefore, the number of permeability realizations local to the KL region i and global to the entire domain are

$$N_{\text{real}}(i) = \prod_{j=1}^{N_{\text{term}}(i)} N_{coll}(i,j) \quad \text{and} \quad N_{\text{real}} = \prod_{i=1}^{N_\Omega} N_{\text{real}}(i), \quad \text{respectively.}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

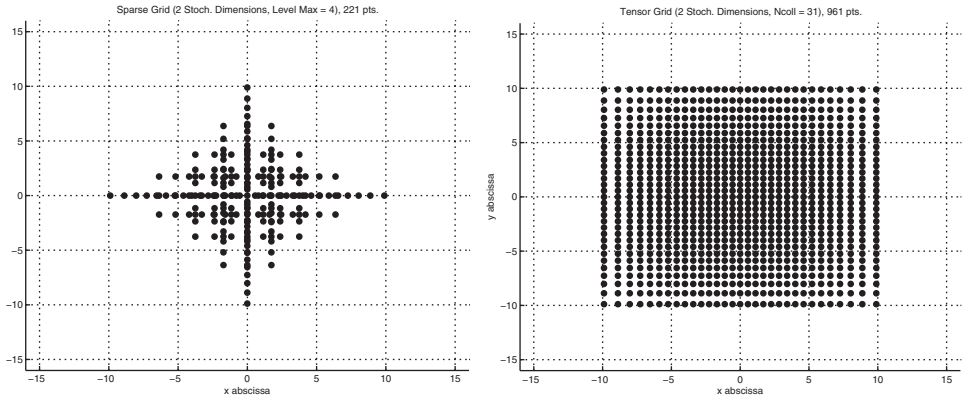

FIG. 1. *A Gauss–Hermite sparse grid (left) versus a Gauss–Hermite tensor grid (right) with a comparable number of points on each axis.*

In the case of isotropic tensor product collocation where each stochastic dimension $\mathbb{S}_j^{(i)}$ has the same polynomial accuracy $\mathbf{m} = (m, m, \ldots, m)$, the tensor grid points are $\mathcal{T}(\mathbf{m}) = \bigotimes_{k=1}^{N_{\text{term}}} \mathcal{H}(m)$ and the number of realizations reduces to $N_{\text{real}}(i) = m^{N_{\text{term}}(i)}$ and $N_{\text{real}} = m^{N_{\text{term}}}$.

Remark 3.2. We index the tensor product collocation points with a natural ordering. For the tensor grid point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$, its global collocation index $k \in \{1, \ldots, N_{\text{real}}\}$ is given by

$$k = k_1 + \sum_{i=2}^{N_{\text{term}}} k_i \prod_{j=1}^{i-1} \mathbf{m}(k). \tag{13}$$

3.4. Collocation on sparse grids. Sparse grids were first used for high dimensional quadrature by Smolyak in 1963 [33] and have been applied to stochastic collocation in such works as [39, 28]. In sparse grid collocation, the polynomial accuracy is prescribed in terms of *total* degree. Sparse grids rules are known to have the same asymptotic accuracy as tensor product rules, while requiring far fewer points as the dimension increases. This property is essential for coping with the curse of dimensionality. Therefore sparse grids are applicable for problems with higher dimensional noise, i.e., up to several hundred stochastic dimensions. A picture of comparable sparse grid and tensor grid rules is shown in Figure 1.

Sparse grid rules are linear combinations of tensor products on a family of nested one-dimensional rules. They are constructed hierarchically to have the property that the total polynomial degree is a constant independent of dimension. They are described in terms of a level $\ell_{\max}$, where the N_{term}-dimensional sparse grid quadrature rule of level $\ell_{\max}$ is accurate to degree $(2 \cdot \ell_{\max} + 1)$.

Each level between $\ell_{\max}$ and $\ell_{\min} = \max\{0, \ell_{\max} - N_{\text{term}} + 1\}$ is an integer partitioned into N_{term} non-negative parts. These partitions form multi-indices $\mathbf{p} = (p_1, \ldots, p_{N_{\text{term}}})$, $|\mathbf{p}| = \sum p_i$, denoting the levels of one-dimensional rules to use for each stochastic dimension. In our paper, the one-dimensional abscissae of level p_i are the Gauss–Hermite points $\mathcal{H}(2^{p_i+1} - 1)$. Level 0 starts with a single point, and the number of points doubles plus one on each subsequent level.

Let the multi-index $\mathbf{m} = 2^{\mathbf{p}+\mathbb{1}} - \mathbb{1}$ denote degree for each partition $\mathbf{p}$. The corresponding isotropic sparse grid Gauss–Hermite interpolant in N_{term} dimensions

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

is defined by

$$\mathcal{I}^{\mathrm{SG}}_{\ell_{\max}} f(\mathbf{y}) = \sum_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} (-1)^{\ell_{\max} - |\mathbf{p}|} \cdot \binom{N_{\mathrm{term}} - 1}{\ell_{\max} - |\mathbf{p}|} \cdot \mathcal{I}^{\mathrm{TG}}_{\mathbf{m}} f(\mathbf{y}).$$

The set of abscissae for this rule is

$$\mathcal{S}(\ell_{\min}, \ell_{\max}, N_{\mathrm{term}}) = \bigcup_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} \bigotimes_{i=1}^{N_{\mathrm{term}}} \mathcal{H}(2^{p_i+1} - 1). \tag{14}$$

Remark 3.3. The set of local permeability realizations on a sparse grid for a particular KL region satisfies the relationship

$$\mathcal{S}(\ell_{\min}, \ell_{\max}, N_{\mathrm{term}}) \subsetneq \underbrace{\mathcal{S}(0, \ell_{\max}, N_{\mathrm{term}}(1))}_{\text{Projection into } \mathbb{S}^{(1)}} \otimes \cdots \otimes \underbrace{\mathcal{S}(0, \ell_{\max}, N_{\mathrm{term}}(N_\Omega))}_{\text{Projection into } \mathbb{S}^{(N_\Omega)}}.$$

Note that $\ell_{\min} = 0$ for all the local sparse grids, unlike the global sparse grid.

The points in (14) are weakly nested because the origin is the sole value that is repeated in each one-dimensional rule. Taking tensor products of one-dimensional rules produces many repeated points that contain the origin in one or more of its components. There are both pros and cons to skipping these repeated abscissae. On the one hand, fewer function evaluations in the quadrature rule means fewer realizations to solve in (11). On the other hand, extra bookkeeping is necessary for indexing the points and calculating their collocation weights.

In Algorithm 1, we give an efficient method that provides a natural ordering for the points in a Gauss–Hermite sparse grid, which skips repeated points.

Algorithm 1. A natural ordering for sparse grid points.

input: Global Index g
$j \leftarrow 0$
for $\ell \leftarrow \ell_{\min}, \ldots, \ell_{\max}$ **do** {Loop over levels}
  for $i \leftarrow 1, \ldots, \dfrac{(\ell + N_{\mathrm{term}} - 1)!}{\ell!(N_{\mathrm{term}} - 1)!}$ **do** {Loop over partitions}
    $part \leftarrow (p_1, \ldots, p_{N_{\mathrm{term}}})$, $\quad \mathbf{m} \leftarrow 2^{part + \mathbb{1}} - \mathbb{1}$ {The ith multi-index}
    if $\ell = \ell_{\min}$ **then** add $part$ to $PartList$
    for $k = 1, \ldots, \prod_\alpha \mathbf{m}(\alpha)$ **do** {Loop over points}
      $point \leftarrow (h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\mathrm{term}}}}_{\mathbf{m}(N_{\mathrm{term}})})$ {The jth point using (13)}
      $\tilde{part} \leftarrow (\tilde{p}_1, \ldots, \tilde{p}_{N_{\mathrm{term}}})$ where $\tilde{p}_i = \begin{cases} p_i, & \text{if } h^{k_i}_{\mathbf{m}(i)} \neq 0 \\ 0, & \text{if } h^{k_i}_{\mathbf{m}(i)} = 0 \end{cases}$
      if ($\ell = \ell_{\min}$ and $\tilde{part} \in PartList$) **then** {Repeated point; skip it}
      else if ($\ell > \ell_{\min}$ and $\tilde{part} \neq part$) **then** {Repeated point; skip it}
      else $j \leftarrow j + 1$ {Unique point; count it}
      if ($j = g$) **then return** $point, part$
    end for
  end for
end for

Suppose that a sparse grid point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\mathrm{term}}}}_{\mathbf{m}(N_{\mathrm{term}})})$ occurs in a set of partitions $\mathcal{P}$. If it is used in a single function evaluation with subsequent occurrences skipped

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

by Algorithm 1, then its quadrature weight must be calculated by the formula

$$\sum_{\mathbf{p}\in\mathcal{P}}(-1)^{\ell_{\max}-|\mathbf{p}|}\cdot\binom{N_{\text{term}}-1}{\ell_{\max}-|\mathbf{p}|}\prod_{i=1}^{N_{\text{term}}}w_{\mathbf{m}(i)}^{k_i}.$$

Note that in a sparse grid rule, quadrature weights may become negative.

4. Error analysis. In this section we present a priori error estimates for the solution to the stochastic MMMFEM. As in previous stochastic collocation papers (see e.g., [8, 39, 28, 19]), the error is decomposed into deterministic and stochastic errors (see Theorem 4.4). Furthermore, we employ a duality argument to show superconvergence for the pressure (see Theorem 4.5).

Note that throughout this entire section, we tacitly assume that Assumption 3.1 holds. To avoid technical details for the approximation of the Neumann boundary condition, we further assume that $g_N \in \mathbf{V}_h(D)\cdot\mathbf{n}$.

We start with some definitions. We define the space of weakly continuous velocities by

$$\mathbf{V}_{h,0}(D)=\left\{\mathbf{v}\in\mathbf{V}_h(D)\mid\sum_{i=1}^{N_D}\langle\mathbf{v}|_{D_i}\cdot\mathbf{n}_i,\mu\rangle_{\Gamma_i}=0\quad\forall\mu\in M_H(\Gamma)\right\}.$$

Recall that for any of the standard mixed spaces, $\nabla\cdot\mathbf{V}_{h,i}(D)=W_{h,i}(D)$. Let, for $\varepsilon>0$, $\Pi_i:(H^\varepsilon(D_i))^d\cap\mathbf{V}_i(D)\to\mathbf{V}_{h,i}(D)$, $\Pi\mathbf{q}|_{D_i}=\Pi_i\mathbf{q}$, be the standard MFE projection operators. A projection operator $\Pi_0:(H^{1/2+\varepsilon}(D))^d\cap\mathbf{V}(D)\to\mathbf{V}_{h,0}(D)$ is defined in [5, 6], satisfying

$$(\nabla\cdot(\Pi_0\mathbf{q}-\mathbf{q}),w)_{D_i}=0\qquad\forall\, w\in W_{h,i}(D_i), \tag{15a}$$

$$\|\Pi_0\mathbf{q}-\Pi\mathbf{q}\|\le C\sum_{i=1}^{N_D}\|\mathbf{q}\|_{r+1/2,D_i}h^rH^{1/2}\qquad 0\le r\le k+1, \tag{15b}$$

$$\|\Pi_0\mathbf{q}-\mathbf{q}\|\le C\sum_{i=1}^{N_D}\|\mathbf{q}\|_{r,D_i}h^{r-1/2}H^{1/2}\qquad 1\le r\le k+1\ , \tag{15c}$$

$$\left(\sum_{i=1}^{N_D}\|\Pi_0\mathbf{q}\|^2_{H(div,D_i)}\right)^{1/2}\le C\sum_{i=1}^{N_D}\|\mathbf{q}\|_{1,D_i}. \tag{15d}$$

Note that (15d) is not explicitly stated in [5, 6], but follows easily from the results there.

For any $\varphi\in L^2(D)$, define its $L^2(D)$ projection $\hat{\varphi}$ onto $W_h(D)$ by

$$(\varphi-\hat{\varphi},w)=0\qquad\forall w\in W_h(D).$$

Similarly, let $\mathcal{P}_H$ denote the $L^2(\Gamma)$ projection onto $M_H(\Gamma)$. Let $\mathcal{I}_H^c$ be the nodal interpolant operator into the space $M_H^c(\Gamma)$, which is the subset of continuous functions in $M_H(\Gamma)$, where we use the Scott–Zhang operator [32] to define the nodal values of

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

ψ if it does not have pointwise values. We will make use of the following inequalities:

$$\|\psi - \mathcal{I}_H^c \psi\|_{t,\Gamma_i} \le C\|\psi\|_{s,\Gamma_i} H^{s-t} \qquad 0 \le s \le r+1, \quad 0 \le t \le 1, \tag{16a}$$

$$\|\psi - \mathcal{P}_H \psi\|_{-t,\Gamma_i} \le C\|\psi\|_{s,\Gamma_i} H^{s+t} \qquad 0 \le s \le r+1, \quad 0 \le t \le 1, \tag{16b}$$

$$\|\varphi - \hat{\varphi}\| \le C\|\varphi\|_t h^t \qquad 0 \le t \le l+1, \tag{16c}$$

$$\|q\|_{t,\Gamma_i} \le C\|q\|_{t+1/2,D_i} \qquad 0 < t, \tag{16d}$$

$$\|\mathbf{v}\cdot\mathbf{n}\|_{\Gamma_i} \le Ch^{-1/2}\|\mathbf{v}\|_{D_i} \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}(D_i), \tag{16e}$$

$$\langle \psi, \mathbf{q}\cdot\mathbf{n}\rangle_{\Gamma_i} \le C\|\psi\|_{1/2,\Gamma_i}\|\mathbf{q}\|_{H(div;D_i)} \tag{16f}$$

$$\|(\mathbf{q} - \Pi_i \mathbf{q})\cdot\mathbf{n}_i\|_{-t,\Gamma_i} \le C\|\mathbf{q}\|_{s,\Gamma_i} h^{s+t} \qquad 0 \le s \le k+1, \quad 0 \le t \le k+1, \tag{16g}$$

where $\|\cdot\|_{-t}$ is the norm of H^{-t}, the dual of H^t (not H_0^t). Bound (16a) is found in [32], the L^2-projection approximations (16b), (16c), and (16g) are found in [12], the nonstandard trace theorem (16d) is found in [23], the trace inequality (16e) is found in [5], and the bound (16f) follows from the normal trace inequality for $H(\mathrm{div}; D_i)$ functions.

It is easy to see that (8) is equivalent to finding $\mathbf{u}_h : \mathbb{S} \to \mathbf{V}_{h,0}^{g_N}(D)$ and $p_h : \mathbb{S} \to W_h(D)$ such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\sum_{i=1}^{N_D} (K^{-1}\mathbf{u}_h, \mathbf{v})_{D_i} = \sum_{i=1}^{N_D} (p_h, \nabla\cdot\mathbf{v})_{D_i} - \langle \mathbf{v}\cdot\mathbf{n}_i, g_D\rangle_{\Gamma_D} \quad \forall \mathbf{v} \in \mathbf{V}_{h,0}^0(D), \tag{17a}$$

$$\sum_{i=1}^{N_D} (\nabla\cdot\mathbf{u}_h, w)_{D_i} = \sum_{i=1}^{N_D} (f, w)_{D_i} \qquad \forall w \in W_h(D). \tag{17b}$$

We form error equations by integrating system (17) in $\mathbb{S}$ against the PDF and subtracting it from system (7):

$$\int_{\mathbb{S}} \sum_{i=1}^{N_D} (K^{-1}(\mathbf{u} - \mathbf{u}_h), \mathbf{v})_{D_i} \rho(\mathbf{y}) d\mathbf{y} = \int_{\mathbb{S}} \Big[\sum_{i=1}^{N_D} (\hat{p} - p_h, \nabla\cdot\mathbf{v})_{D_i} - \langle p, \mathbf{v}\cdot\mathbf{n}_i\rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \quad \forall \mathbf{v} \in \mathbf{V}_{h,0}^0(D), \tag{18a}$$

$$\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot(\mathbf{u} - \mathbf{u}_h), w)_{D_i} \rho(\mathbf{y}) d\mathbf{y} = 0 \qquad \forall w \in W_h(D). \tag{18b}$$

Recall that k, l, r, and m denote the polynomial degrees of approximation for the velocity space, pressure space, mortar space, and collocation interpolant, respectively. In all mixed methods we consider, $l = k$ or $l = k-1$. The next result follows easily from the deterministic multiscale bound on the velocity, which is originally proved in Theorem 4.1 in [6] and improved in [35]. Sufficient conditions on the data for the required smoothness of the solution can be found in [23].

LEMMA 4.1. *There exists a positive constant C independent of h and H such that for $0 \le q \le l+1$, $1 \le t \le k+1$, $1/2 \le \tilde{t} \le k+1$, and $1/2 \le s \le r+1$,*

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(D)\otimes L^2(\mathbb{S})} \le C(\|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2} + \|\mathbf{u}\|_{H^t(D)\otimes L^2(\mathbb{S})} h^t + \|\mathbf{u}\|_{H^{\tilde{t}+1/2}(D)\otimes L^2(\mathbb{S})} h^{\tilde{t}} H^{1/2}),$$

$$\|\nabla\cdot(\mathbf{u} - \mathbf{u}_h)\|_{L^2(D_i)\otimes L^2(\mathbb{S})} \le C\|\nabla\cdot\mathbf{u}\|_{H^q(D_i)\otimes L^2(\mathbb{S})} h^q \qquad 1 \le i \le N_D.$$

For the pressure bound we need the following inf-sup condition.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

LEMMA 4.2. *There exists a positive constant γ independent of h and H such that for all $w : \mathbb{S} \to W_h(D)$,*

$$\sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \geq \gamma \|w\|_W.$$

Proof. Let $w : \mathbb{S} \to W_h(D)$. Consider the auxiliary problem, for ρ-almost every $\mathbf{y} \in \mathbb{S}$

$$\nabla\cdot\boldsymbol{\psi}(\cdot,\mathbf{y}) = w(\cdot,\mathbf{y}) \ \text{ in } D, \qquad \boldsymbol{\psi}(\cdot,\mathbf{y}) = \mathbf{g}(\cdot,\mathbf{y}) \ \text{ on } \partial D,$$

where $\mathbf{g} \in (H^{1/2}(\partial D))^d$ is constructed to satisfy $\int_{\partial D} \mathbf{g}\cdot\mathbf{n} = \int_D w$ and $\mathbf{g}\cdot\mathbf{n} = 0$ on Γ_N. More precisely, we take $\mathbf{g} = (\int_D w)\varphi\mathbf{n}$, where $\varphi \in C^0(\partial D)$ is such that $\int_{\partial D} \varphi = 1$ and $\varphi = 0$ on Γ_N. Clearly $\|\mathbf{g}\|_{1/2,\partial D} \leq C\|w\|$. It is known [17] that the above problem has a solution satisfying

$$\|\boldsymbol{\psi}\|_1 \leq C(\|w\| + \|\mathbf{g}\|_{1/2,\partial D}) \leq C\|w\|. \tag{19}$$

Then

$$\sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \geq \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\Pi_0\boldsymbol{\psi}, w)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\Pi_0\boldsymbol{\psi}\|_V} \geq \gamma\|w\|_W,$$

where we have used (15a), (15d), and (19) for the last inequality. ∎

From Lemma 4.2, we can derive a multiscale bound on the semidiscrete pressure.

LEMMA 4.3. *There exists a positive constant C independent of h and H such that for $1 \leq t \leq k+1$, $1/2 \leq \tilde{t} \leq k+1$, and $1/2 \leq s \leq r+1$,*

$$\begin{aligned}&\|p - p_h\|_W \\ &\leq C(\|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2} + \|\mathbf{u}\|_{H^t(D)\otimes L^2(\mathbb{S})} h^t + \|\mathbf{u}\|_{H^{\tilde{t}+1/2}(D)\otimes L^2(\mathbb{S})} h^{\tilde{t}} H^{1/2}).\end{aligned}$$

Proof. Taking $w = \hat{p} - p_h$ in Lemma 4.2 and using (18a) gives

$$\begin{aligned}\|\hat{p} - p_h\|_W &\leq \frac{1}{\gamma} \sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla\cdot\mathbf{v}, \hat{p} - p_h)_{D_i} \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \\ &= \frac{1}{\gamma} \sup_{\mathbf{v}:\mathbb{S}\to\mathbf{V}^0_{h,0}(D)} \frac{\int_{\mathbb{S}} \sum_{i=1}^{N_D} \left[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \mathbf{v})_{D_i} + \langle p - \mathcal{I}_H^c p, \mathbf{v}\cdot\mathbf{n}_i\rangle_{\Gamma_i}\right] \rho(\mathbf{y}) d\mathbf{y}}{\|\mathbf{v}\|_V} \\ &\leq C\left(\|\mathbf{u} - \mathbf{u}_h\|_{L^2(D)\otimes L^2(\mathbb{S})} + \|p\|_{H^{s+1/2}(D)\otimes L^2(\mathbb{S})} H^{s-1/2}\right),\end{aligned}$$

where we have used (16a) and (16d) in the last inequality. The proof is completed using Lemma 4.1, the triangle inequality, and (16c). ∎

THEOREM 4.4. *Assume that the solution $(\mathbf{u}, p)$ to (2) is sufficiently smooth, so that the norms that appear in Lemma 4.1 are well defined. Then there exists a positive constant C independent of h and H such that for $0 \leq q \leq l+1$, $1 \leq t \leq k+1$, $1/2 \leq \tilde{t} \leq k+1$, and $1/2 \leq s \leq r+1$,*

$$\|\mathbf{u} - \mathbf{u}_{h,m}\|_V + \|p - p_{h,m}\|_W \leq C(H^{s-1/2} + h^q + h^t + h^{\tilde{t}} H^{1/2}) + \eta. \tag{20a}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

For tensor product grid collocation,

$$\eta \le C \sum_{i=1}^{N_\Omega} \exp\left(-c_i \sqrt{m_i}\right), \tag{20b}$$

For sparse grid collocation,

$$\eta \le C(\sigma) \begin{cases} \exp(-\sigma N_{\text{term}} 2^{\ell_{\max}/N_{\text{term}}}), & \text{for large } \ell_{\max}, \\ \exp(-\sigma e \log_2(\ell_{\max})), & \text{for large } N_{\text{term}}. \end{cases} \tag{20c}$$

In the above c_i and σ are positive constants that depend on the smoothness of $\mathbf{u}$ and p in $\mathbb{S}$.[3]

Proof. The left-hand side of (20a) can be decomposed into deterministic and stochastic errors:

$$\begin{aligned} &\|\mathbf{u} - \mathbf{u}_{h,m}\|_V + \|p - p_{h,m}\|_W \\ &\qquad \le (\|\mathbf{u} - \mathbf{u}_h\|_V + \|p - p_h\|_W) + (\|\mathbf{u}_h - \mathcal{I}_m \mathbf{u}_h\|_V + \|p_h - \mathcal{I}_m p_h\|_W). \end{aligned}$$

The deterministic error, represented by the first two terms, is bounded in Lemmas 4.1 and 4.3. Assuming K is smooth enough in $\mathbb{S}$, which is true for the KL expansion (5), the estimate of the stochastic error in the case of tensor product grid collocation (20b) can be found in [8], and in the case of sparse grid collocation (20c) can be found in [28]. ∎

In the next theorem we establish superconvergence for the pressure.

THEOREM 4.5. *Assume that the problem (2) is H^2-elliptic regular. Under the assumptions of Theorem 4.4, there exists a positive constant C independent of h and H such that for $0 \le q \le l+1$, $1 \le t \le k+1$, $1/2 \le \tilde{t} \le k+1$, and $1/2 \le s \le r+1$,*

$$\|\hat{p} - p_{h,m}\|_W \le C(H^{s+1/2} + h^q H + h^t H + h^{\tilde{t}} H^{3/2}) + \eta, \tag{21}$$

where η is defined in Theorem 4.4.

Proof. Consider the following auxiliary problem in mixed form. For ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\boldsymbol{\psi}(\cdot, \mathbf{y}) = -K(\cdot, \mathbf{y}) \nabla \varphi(\cdot, \mathbf{y}) \quad \text{in } D, \tag{22a}$$

$$\nabla \cdot \boldsymbol{\psi}(\cdot, \mathbf{y}) = \hat{p} - p_{h,m} \quad \text{in } D, \tag{22b}$$

$$\varphi(\cdot, \mathbf{y}) = 0 \quad \text{on } \Gamma_D \tag{22c}$$

$$\boldsymbol{\psi}(\cdot, \mathbf{y}) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_N\ . \tag{22d}$$

The H^2-elliptic regularity implies

$$\|\varphi(\cdot, \mathbf{y})\|_2 \le C \|\hat{p} - p_{h,m}\|. \tag{22e}$$

We have

$$\begin{aligned} \|\hat{p} - p_{h,m}\|_W^2 &= \int_{\mathbb{S}} (\hat{p} - p_{h,m}, \hat{p} - p_{h,m}) \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \left[(\nabla \cdot \boldsymbol{\psi}, \hat{p} - p_h) + (\hat{p} - p_{h,m}, p_h - \mathcal{I}_m p_h) \right] \rho(\mathbf{y}) d\mathbf{y} \\ &= I + II. \end{aligned}$$

[3] The precise statement on the smoothness requirement is given in [8] for tensor grid and [28] for sparse grid.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Applying the Cauchy–Schwarz inequality,

$$|II| \le \|\hat{p} - p_{h,m}\|_W \|p_h - \mathcal{I}_m p_h\|_W = \|\hat{p} - p_{h,m}\|_W \eta.$$

Taking $\mathbf{v} = \Pi_0 \boldsymbol{\psi} \in \mathbf{V}_{h,0}^0(D)$ in (18a) and using (15a), we have

$$\begin{aligned} I &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} (\nabla \cdot \Pi_0 \boldsymbol{\psi}, \hat{p} - p_h)_{D_i} \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \Pi_0 \boldsymbol{\psi})_{D_i} + \langle p - \mathcal{P}_H p, \Pi_0 \boldsymbol{\psi} \cdot \mathbf{n}_i \rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \text{by (18b)} \\ &= I_1 + I_2. \end{aligned}$$

We can break up I_1 into three terms by

$$\begin{aligned} I_1 &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \Pi_0 \boldsymbol{\psi} - \boldsymbol{\psi})_{D_i} - (\mathbf{u} - \mathbf{u}_h, \nabla \varphi)_{D_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\ &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[(K^{-1}(\mathbf{u} - \mathbf{u}_h), \Pi_0 \boldsymbol{\psi} - \boldsymbol{\psi})_{D_i} + (\nabla \cdot (\mathbf{u} - \mathbf{u}_h), \varphi - \hat{\varphi})_{D_i} \\ &\qquad - \langle (\mathbf{u} - \mathbf{u}_h) \cdot \mathbf{n}_i, \varphi - \mathcal{I}_H^c \varphi \rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\ &= I_{11} + I_{12} - I_{13}. \end{aligned}$$

Upper bounds for I_{11}, I_{12}, and I_{13} can be obtained using the Cauchy–Schwarz inequality.

$$\begin{aligned} I_{11} &\le C \left(\int_{\mathbb{S}} \|\mathbf{u} - \mathbf{u}_h\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|\Pi_0 \boldsymbol{\psi} - \boldsymbol{\psi}\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le C \sqrt{hH}\ \|\mathbf{u} - \mathbf{u}_h\|_V \left(\int_{\mathbb{S}} \|\boldsymbol{\psi}\|_1^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \qquad \text{by (15c)} \\ &\le C \sqrt{hH}\ \|\mathbf{u} - \mathbf{u}_h\|_V\ \|\hat{p} - p_{h,m}\|_W \qquad \text{by (22e), (22a).} \end{aligned}$$

$$\begin{aligned} I_{12} &\le C \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\nabla \cdot (\mathbf{u} - \mathbf{u}_h)\|_{D_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \left(\int_{\mathbb{S}} \|\varphi - \hat{\varphi}\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\le Ch \|\mathbf{u} - \mathbf{u}_h\|_V \|\hat{p} - p_{h,m}\|_W \qquad \text{by (16c), (22e).} \end{aligned}$$

$$\begin{aligned} I_{13} &\le C \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\mathbf{u} - \mathbf{u}_h\|_{H(div, D_i)}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \\ &\qquad \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|\varphi - \mathcal{I}_H^c \varphi\|_{1/2, \Gamma_i}^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \qquad \text{by (16f)} \\ &\le CH \|\mathbf{u} - \mathbf{u}_h\|_V \|\hat{p} - p_{h,m}\|_W \qquad \text{by (16a), (16d), (22e).} \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We bound I_2 as follows:

$$
\begin{aligned}
I_2 &= \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\langle p - \mathcal{P}_H p, (\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}) \cdot \mathbf{n}_i + (\Pi_i \boldsymbol{\psi} - \boldsymbol{\psi}) \cdot \mathbf{n}_i + \boldsymbol{\psi} \cdot \mathbf{n}_i \rangle_{\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \\
&\le \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p - \mathcal{P}_H p\|_{\Gamma_i} \Big(\|(\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}) \cdot \mathbf{n}_i\|_{\Gamma_i} + \|(\Pi_i \boldsymbol{\psi} - \boldsymbol{\psi}) \cdot \mathbf{n}_i\|_{\Gamma_i} \Big) \\
&\qquad + \|p - \mathcal{P}_H p\|_{-1/2,\Gamma_i} \|\boldsymbol{\psi} \cdot \mathbf{n}_i\|_{1/2,\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \text{by (16f)} \\
&\le C \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p\|_{s,\Gamma_i} H^s \Big(\|\Pi_0 \boldsymbol{\psi} - \Pi_i \boldsymbol{\psi}\|_{D_i} h^{-1/2} + \|\boldsymbol{\psi}\|_{1/2,\Gamma_i} h^{1/2} \Big) \\
&\qquad + \|p\|_{s,\Gamma_i} H^{s+1/2} \|\boldsymbol{\psi}\|_{1/2,\Gamma_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \text{by (16b), (16e), (16g)} \\
&\le C \int_{\mathbb{S}} \sum_{i=1}^{N_D} \Big[\|p\|_{s+1/2,D_i} H^s \Big(\|\boldsymbol{\psi}\|_{1,D_i} H^{1/2} + \|\boldsymbol{\psi}\|_{1,D_i} h^{1/2} \Big) \\
&\qquad + \|p\|_{s+1/2,D_i} H^{s+1/2} \|\boldsymbol{\psi}\|_{1,D_i} \Big] \rho(\mathbf{y}) d\mathbf{y} \qquad \text{by (16d), (15b)} \\
&\le C H^{s+1/2} \left(\int_{\mathbb{S}} \sum_{i=1}^{N_D} \|p\|^2_{s+1/2,D_i} \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} \|\hat{p} - p_{h,m}\|_W .
\end{aligned}
$$

The proof is completed by combining the above inequalities and using Theorem 4.4. ∎

5. Collocation-MMMFEM algorithms. To form the fully discrete stochastic solution to (11), each realization requires solving a deterministic problem using the MMMFEM. For these we employ a parallel substructuring domain decomposition algorithm [22, 5, 6] that reduces the global problem to a coarse scale interface problem for a mortar pressure. In this section we present three algorithms based on combining stochastic collocation with different implementations of the solution of the interface problem.

We begin by describing the reduction to an interface problem. We decompose the solutions into two parts

$$\mathbf{u}_h^{\{k\}} = \mathbf{u}_h^{*,\{k\}}(\lambda_H) + \overline{\mathbf{u}}_h^{\{k\}} \quad \text{and} \quad p_h^{\{k\}} = p_h^{*,\{k\}}(\lambda_H) + \overline{p}_h^{\{k\}}.$$

For each realization $k = 1, \ldots, N_{\text{real}}$, the pair $(\mathbf{u}_h^{*,\{k\}}, p_h^{*,\{k\}}) \in \mathbf{V}_h^0(D) \times W_h(D)$ solves subdomain problems with zero source and outside boundary conditions, and has $\lambda_H^{\{k\}}$ as a Dirichlet boundary condition along Γ; i.e., for $i = 1, \ldots, N_D$,

$$((K^{\{k\}})^{-1} \mathbf{u}_h^{*,\{k\}}, \mathbf{v})_{D_i} - (p_h^{*,\{k\}}, \nabla \cdot \mathbf{v})_{D_i} = -\langle \mathbf{v} \cdot \mathbf{n}_i, \lambda_H^{\{k\}} \rangle_{\Gamma_i} \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0(D), \tag{23a}$$

$$(\nabla \cdot \mathbf{u}_h^{*,\{k\}}, w)_{D_i} = 0 \quad \forall w \in W_{h,i}(D), \tag{23b}$$

and the pair $(\overline{\mathbf{u}}_h^{\{k\}}, \overline{p}_h^{\{k\}}) \in \mathbf{V}_h^{g_N}(D) \times W_h(D)$ solves subdomain problems with source f, boundary conditions g_D and g_N on ∂D, and zero Dirichlet boundary conditions along Γ; i.e., for $i = 1, \ldots, N_D$,

$$((K^{\{k\}})^{-1} \overline{\mathbf{u}}_h^{\{k\}}, \mathbf{v})_{D_i} - (\overline{p}_h^{\{k\}}, \nabla \cdot \mathbf{v})_{D_i} = -\langle \mathbf{v} \cdot \mathbf{n}_i, g_D \rangle_{\partial D_i \cap \Gamma_D} \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}^0(D), \tag{24a}$$

$$(\nabla \cdot \overline{\mathbf{u}}_h^{\{k\}}, w)_{D_i} = (f, w)_{D_i} \quad \forall w \in W_{h,i}(D). \tag{24b}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Notice that systems (23a)–(23b) and (24a)–(24b) are completely decoupled from each other across all subdomains. The former requires a mortar function $\lambda_H^{\{k\}}$ while the latter does not. Their sum equals (11a)–(11b), so what remains is to enforce equation (11c), which couples the subdomains together. This leads to the variational interface problem: find $\lambda_H^{\{k\}} \in M_H(\Gamma)$ such that for all $k = 1, \ldots, N_{\text{real}}$,

$$b_H^{\{k\}}(\lambda_H^{\{k\}}, \mu) = g_H^{\{k\}}(\mu) \qquad \forall \mu \in M_H(\Gamma), \tag{25}$$

where the bilinear forms $b_H^{\{k\}} : L^2(\Gamma) \times L^2(\Gamma) \to \mathbb{R}$, $b_{H,i}^{\{k\}} : L^2(\Gamma_i) \times L^2(\Gamma_i) \to \mathbb{R}$, and linear functional $g_H^{\{k\}} : L^2(\Gamma) \to \mathbb{R}$ are defined by

$$b_H^{\{k\}}(\lambda, \mu) = \sum_{i=1}^{N_D} b_{H,i}^{\{k\}}(\lambda, \mu), \; b_{H,i}^{\{k\}}(\lambda, \mu) = \langle -\mathbf{u}_{h,i}^{*,\{k\}}(\lambda) \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i},$$
$$g_H^{\{k\}}(\mu) = \sum_{i=1}^{N_D} \langle \overline{\mathbf{u}}_{h,i}^{\{k\}} \cdot \mathbf{n}_i, \mu \rangle_{\Gamma_i}.$$

Note that $b_H^{\{k\}}$ measures the jump in flux across subdomain boundaries and requires interprocess communication, while $b_{H,i}^{\{k\}}$ measures the flux on a single subdomain.

It is shown in [5] that if Assumption 3.1 holds and $\Gamma_D \neq \emptyset$, then $b_H^{\{k\}}$ is symmetric and positive definite on $M_H(\Gamma)$. Therefore we use the conjugate gradient (CG) algorithm to solve the interface problem (25). It is convenient to rewrite (25) using an operator notation: find $\lambda_H^{\{k\}} \in M_H(\Gamma)$ such that for $k = 1, \ldots, N_{\text{real}}$,

$$B_H^{\{k\}} \lambda_H^{\{k\}} = g_H^{\{k\}}, \tag{26}$$

where the linear operators $B_H^{\{k\}} : M_H(\Gamma) \to M_H(\Gamma)$, $B_{H,i}^{\{k\}} : M_{H,i}(\Gamma_i) \to M_{H,i}(\Gamma_i)$, and the vector $g_H^{\{k\}} \in M_H(\Gamma)$ are defined by

$$B_H^{\{k\}} \lambda = \sum_{i=1}^{N_D} B_{H,i}^{\{k\}} \lambda, \qquad \langle B_{H,i}^{\{k\}} \lambda, \mu \rangle_{\Gamma_i} = b_{H,i}^{\{k\}}(\lambda, \mu) \qquad \forall \mu \in M_{H,i},$$
$$\langle g_H^{\{k\}}, \mu \rangle_\Gamma = g_H^{\{k\}}(\mu) \qquad \forall \mu \in M_H.$$

The operator $B_H^{\{k\}}$ is known as the Steklov–Poincaré operator [30].

The dominant cost for solving the interface problem (26) in the MMMFEM is the solution of Dirichlet-to-Neumann subdomain problems (23) on each CG iteration.

The three collocation-MMMFEM algorithms are presented below. We measure their computational cost in terms of the number of subdomain solves.

5.1. Collocation with traditional MMMFEM. The first method we consider is based on the traditional implementation of the MMMFEM. It requires one subdomain solve (23) per interface CG iteration. We call this Method S1 and present it in Algorithm 2.

In step 3 of Algorithm 2, the interface iteration can be summarized as follows. The vector $g_H^{\{k\}}$ is formed by solving system (24). Starting from an initial guess, we

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 2 (Method S1). Collocation without a Multiscale Flux Basis.

1: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
2: Generate permeability realization $K^{\{k\}}$ corresponding to global index k
3: Solve interface problem (26) using the Traditional Implementation of the MMMFEM
4: Multiply solution by collocation weight and sum to statistical moments
5: **end for**

iterate until convergence on the value of $\lambda_H^{\{k\}}$ using the conjugate gradient algorithm. On each CG iteration, the action of the operator $B_H^{\{k\}}$ is performed in four steps:

(a) Project mortar data onto subdomain boundaries: $\gamma_i = \mathcal{Q}_{h,i}(\lambda_{H,i}^{\{k\}})$,
(b) Solve the subdomain problem (23) with Dirichlet boundary data γ_i,
(c) Project the resulting fluxes onto mortar space: $\xi_i = -\mathcal{Q}_{h,i}^T(\mathbf{u}_h^{*,\{k\}}(\gamma_i) \cdot \mathbf{n}_i)$,
(d) Compute flux jumps across subdomain interfaces: $B_H^{\{k\}} \lambda_H^{\{k\}} = \sum_{i=1}^{N_D} \xi_i$.

Steps (a)–(c) evaluate the action of the flux operator $B_{H,i}^{\{k\}}$ and are done by every subdomain in parallel. Step (d) evaluates the action of the jump operator $B_H^{\{k\}}$ and requires interprocess communication across every interface.

Note that the number of CG iterations for solving (26) grows with the condition number of the problem. In the traditional implementation of the MMMFEM, so does the number of subdomain solves. When this method is coupled with the stochastic collocation method, this cost is multiplied by the number of realizations. The computational cost for each subdomain is given by

$$\begin{pmatrix} \text{Number of solves for} \\ \text{Method S1} \end{pmatrix} = \sum_{k=1}^{N_{\text{real}}} (N_{\text{iter}}(k) + 3).$$

The three additional solves at each realization come from solving (24) to form the right-hand side in (26) and recovering the solution in the interior after the convergence of the CG iteration.

5.2. Collocation with a deterministic multiscale flux basis. An alternative implementation of the MMMFEM was recently presented [20] that forms a multiscale flux basis. Each subdomain solves a subdomain problem for each one of its mortar degrees of freedom before the interface iteration begins. The solutions to these problems form a basis of coarse scale flux responses containing all the necessary information to solve a deterministic problem. No interprocess communication is required in the formation of the basis. Linear combinations of the basis are used to evaluate the flux operators during the interface iteration so that no additional subdomain solves are necessary, except one or more additional solves at the conclusion of the iteration to recover the fine scale solution. The computational cost is a fixed and controllable quantity, and therefore does not worsen with the condition number of the problem. Indeed, it was shown to be more efficient than the traditional implementation in most cases for deterministic problems. This gain in computational efficiency increases with the number of subdomains, and also in cases where a basis can be computed once and then reused many times.

This approach can be coupled to stochastic collocation method in a straightforward way by forming a new deterministic multiscale basis for each realization. We call this Method S2 and present it in Algorithm 3.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 3 (Method S2). Collocation with a Deterministic Multiscale Flux Basis.

1: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
2: Generate permeability realization $K^{\{k\}}$ corresponding to global index k
3: Compute multiscale flux basis for global index k
4: Solve interface problem (26) with MMMFEM using the basis from Step 3
5: Multiply solution by collocation weight and sum to statistical moments
6: **end for**

In step 3 of Algorithm 3, the formation of the multiscale flux basis can be summarized as follows. For each subdomain D_i, $i = 1, \ldots, N_D$, let $\{\phi_{H,i}^{(j)}\}_{j=1}^{N_{\text{dof}}(i)}$ denote a mortar basis for $M_{H,i}(\Gamma_i)$. Their individual flux responses for realization k are computed by evaluating the action of the operator $B_{H,i}^{\{k\}}$ on these functions. This is done via steps (a)–(c) from the interface iteration, i.e., for $j = 1, \ldots, N_{\text{dof}}(i)$,

(a') Project a mortar basis function onto subdomain boundary: $\gamma_i^{(j)} = \mathcal{Q}_{h,i}(\phi_{H,i}^{(j)})$,
(b') Solve the subdomain problem (23) with Dirichlet boundary data $\gamma_i^{(j)}$,
(c') Project the resulting flux onto mortar space: $\psi_{H,i}^{(j),\{k\}} = -\mathcal{Q}_{h,i}^T(\mathbf{u}_h^{*,\{k\}}(\gamma_i^{(j)}) \cdot \mathbf{n}_i)$.

The functions $\{\psi_{H,i}^{(j),\{k\}}\}_{j=1}^{N_{\text{dof}}(i)}$ are saved to form the multiscale flux basis for subdomain D_i on global realization k. They are discarded and recalculated for realization $k+1$.

In step 4 of Algorithm 3, the multiscale flux basis is used in the interface iteration as follows. Suppose

$$\lambda_{H,i}^{\{k\}} = \sum_{j=1}^{N_{\text{dof}}(i)} \lambda_{H,i}^{(j),\{k\}} \phi_{H,i}^{(j)}$$

is the current mortar data on a CG iteration. Since the flux operator $B_{H,i}^{\{k\}}$ is linear, steps (a)–(c) in the traditional implementation are replaced with the linear combination

$$\xi_i = B_{H,i}^{\{k\}}\left(\lambda_{H,i}^{\{k\}}\right) = \sum_{j=1}^{N_{\text{dof}}(i)} \lambda_{H,i}^{(j),\{k\}} \psi_{H,i}^{(j),\{k\}}.$$

Then step (d) proceeds as usual.

The computational cost for each subdomain D_i is given by

$$\begin{pmatrix} \text{Number of solves for} \\ \text{Method S2} \end{pmatrix} = (N_{\text{dof}}(i) + 2) * (N_{\text{real}}).$$

Note that each subdomain performs a different number of solves because each one may have a different number of mortar degrees of freedom.

5.3. Collocation with a stochastic multiscale flux basis. The main idea behind the multiscale flux basis implementation is to form a basis containing all the necessary information to solve the interface problem by solving as few linear systems as possible. In Method S2, if each realization saves just a few solves, then when performing several thousand realizations the overall savings will be great compared to Method S1.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

The dominant cost in Method S2 is computing the multiscale flux basis for each global realization. In the setting of nonstationary random porous media with localized KL regions throughout the domain, one can get even greater computational savings with the formation of a stochastic multiscale flux basis. Recalling Remarks 3.1 and 3.3, both tensor product and sparse grids have a repeated local structure in the KL regions. A stochastic multiscale flux basis can be formed by looping over all local realizations of a subdomain's KL region in a precomputation loop before the stochastic collocation begins. We call this Method S3 and present it in Algorithm 4.

Algorithm 4 (Method S3). Collocation with a Stochastic Multiscale Flux Basis.

1: **for** $k' = 1, \ldots, N_{\text{real}}(j)$ **do** {Precomputation Loop}
2: Generate permeability realization corresponding to local index k'
3: Compute and store multiscale flux basis for local index k'
4: **end for**
5: **for** $k = 1, \ldots, N_{\text{real}}$ **do** {Collocation Loop}
6: Generate permeability realization corresponding to global index k
7: Convert global index k to local index k' {Using Algorithm 5 or 6}
8: Solve interface problem (26) with MMMFEM using the basis with local index k' from Precomputation Loop
9: Multiply solution by collocation weight and sum to statistical moments
10: **end for**

While the memory required for Method S3 is more than for Method S2, it is still not significantly large due to the following three reasons. First, the multiscale basis functions are stored only on the boundaries of the subdomains; second, each processor stores only the basis for its subdomain; and third, each processor stores only the bases for the stochastic realizations for its KL region.

The computational cost for a subdomain D_i that belongs to a KL region $D_{KL}^{(j)}$ is given by

$$\begin{pmatrix} \text{Number of solves for} \\ \text{Method S3} \end{pmatrix} = (N_{\text{dof}}(i) * N_{\text{real}}(j)) + (2 * N_{\text{real}}).$$

Each subdomain performs a different number of solves because each one may have a different number of mortar degrees of freedom and may belong to different KL regions with different numbers of local realizations. Note that the dominant cost is in the first term and it is proportional to the number of local realizations $N_{\text{real}}(j)$, while the dominant cost in Method S2 is proportional to the number of global realizations N_{real}.

In step 7 of Algorithm 4, the global to local collocation index conversion is the key step in being able to perform Method S3. Any algorithm developed for this purpose would depend on the ordering of the points. For a tensor product grid, recall from (13) that we chose to follow the natural ordering by one-dimensional (1-D) point first, local dimension next, and KL region last. In Algorithm 5, we give a global to local index conversion algorithm for this ordering. It is very similar to the algorithm one uses to convert an integer from one base into another, with the modification that each digit has a different base.

For a sparse grid, the indexing of the points is far more complicated than a tensor product grid due to its hierarchical construction and skipping of repeated points. Nevertheless, it is still possible to formulate a global to local index conversion that is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

more efficient than a brute force approach. Recall from Algorithm 1 that we chose to follow the natural ordering by level first, followed by a partition of that level into N_Ω parts, followed by 1-D point, and global dimension last. In Algorithm 6, we give the global to local index conversion algorithm for this ordering. It is a modification of Algorithm 5, where the global point and partition are truncated, and the indexing scheme is applied to the local dimensions.

Algorithm 5. Global to Local Index Conversion for a Tensor Product Grid.

input: Global Index g, KL region r
$remainder \leftarrow g$
for $i = 1, \ldots, N_\Omega$ **do**
  $modulus \leftarrow 1$
  for $j = 1, \ldots, N_\Omega - i - 1$ **do**
    $modulus \leftarrow modulus * N_{\text{real}}(j)$
  end for
  if $(N_\Omega - i + 1 = r)$ **then return** $remainder/modulus$ {Return Local Index}
  $remainder \leftarrow \mathbf{mod}(remainder, modulus)$
end for

Algorithm 6. Global to Local Index Conversion for a Sparse Grid.

input: Global Index g, KL region r
$point \leftarrow (h_1, \ldots, h_{N_{\text{term}}})$
$part \leftarrow (p_1, \ldots, p_{N_{\text{term}}})$ {Using Algorithm 1 with index g}
$subpoint \leftarrow (h_{\kappa(r,1)}, \ldots, h_{\kappa(r,N_{\text{term}}(r))})$
$subpart \leftarrow (p_{\kappa(r,1)}, \ldots, p_{\kappa(r,N_{\text{term}}(r))})$ {Truncate to local dimensions}
$l \leftarrow 1$
if $(subpoint = 0)$ **then return** l {Special case for 0 partition}
for $\ell = 1, \ldots, \ell_{\max}$ **do** {Loop over sublevels}
  for $i = 1, \ldots, \dfrac{(\ell + N_{\text{term}}(r) - 1)!}{\ell!(N_{\text{term}}(r) - 1)!}$ **do** {Loop over subpartitions}
    $newpart \leftarrow (q_1, \ldots, q_{N_{\text{term}}(r)})$ {The ith multi-index}
    $\mathbf{m} = 2^{newpart + \mathbb{1}} - \mathbb{1}$
    for $j = 1, \ldots, \prod_\alpha \mathbf{m}(\alpha)$ **do** {Loop over subpoints}
      $newpoint \leftarrow (k^{j_1}_{\mathbf{m}(1)}, \ldots, k^{j_{N_{\text{term}}(r)}}_{\mathbf{m}(N_{\text{term}})})$ {The jth point using (13)}
      $\tilde{newpart} \leftarrow (\tilde{p}_1, \ldots, \tilde{p}_{N_{\text{term}}(r)})$ where $\tilde{p}_i = \begin{cases} p_{\kappa(r,i)}, & \text{if } k^{j_i}_{\mathbf{m}(i)} \neq 0 \\ 0, & \text{if } k^{j_i}_{\mathbf{m}(i)} = 0 \end{cases}$
      if $(\tilde{newpart} = newpart)$ **then**
        $l \leftarrow l + 1$
        if $(newpoint = subpoint)$ **then return** l {Return Local Index}
      else
        {Repeated point; skip it}
      end if
    end for
  end for
end for

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

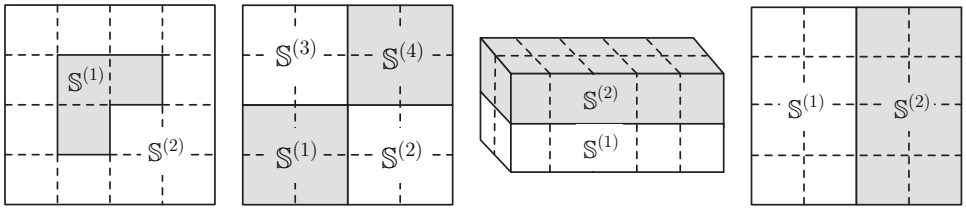

FIG. 2. *Subdomain and KL region layouts for Examples* 1–4. *Dashed lines represent subdomain boundaries and shading distinguishes between KL regions.*

6. Numerical examples. In this section we present four computational examples that illustrate the behavior of the stochastic collocation Methods S1, S2, and S3 for various nonstationary porous media; see Figure 2. In each case we test both tensor product and sparse grid collocations. Example 1 is in 2-D with two KL regions and a highly heterogeneous L-shape inclusion. Example 2 is a 2-D checkerboard with four KL regions and demonstrates a procedure for adaptive mesh refinement in the spatial grid. Example 3 is a 3-D benchmark test with either two or twenty KL regions, and is a much more computationally intensive problem to solve than Examples 1 and 2. In these three examples we compare the relative computational efficiency of Methods S1, S2, and S3 in terms of the maximum number of subdomain linear systems and the maximum total runtime per processor. Finally, Example 4 is a 2-D physical and stochastic space convergence test with two equally sized KL regions.

The numerical experiments use the covariance function listed in [42]. In three dimensions it is given by

$$C_Y^{(i)}(\mathbf{x},\bar{\mathbf{x}}) = \left(\sigma_Y^{(i)}\right)^2 \exp\left[-\frac{|x_1-\bar{x}_1|}{\eta_1^{(i)}} - \frac{|x_2-\bar{x}_2|}{\eta_2^{(i)}} - \frac{|x_3-\bar{x}_3|}{\eta_3^{(i)}}\right].$$

Here $\sigma_Y^{(i)}$ and $\eta_j^{(i)}$ denote the variance and the correlation length in the jth spatial dimension, respectively, for KL region $D_{KL}^{(i)}$. Since it is separable, (3) can be solved in each KL region semianalytically. Details can be found in [42, Appendix A].[4]

The numerical experiments were programmed using a parallel FORTRAN flow simulator named PARCEL [13]. The spatial discretization uses the lowest order Raviart–Thomas elements on rectangles or bricks. Physical discretizations were chosen to satisfy Assumption 3.1. The runtimes were recorded by compiling the code without optimization using Intel's ifort compiler and MKL library, and run with MVAPICH2 on a parallel cluster of Xeon E5430 2.66GHz processors.

6.1. Example 1: L shape.

Description. This example has $N_\Omega = 2$ KL regions in the domain $(0,1)^2$. We test a low number of KL terms with an isotropic tensor product grid and a large number of KL terms with a level $\ell_{\max} = 1$ sparse grid. KL region $\mathbb{S}_1$ is an L-shaped inclusion with a mean value of $E[Y^{(1)}] = 3.0$, $N_{\text{term}}(1) = 3 \times 3 = 9$ (with tensor grid) and $14 \times 14 = 196$ terms (with sparse grid), correlation lengths $\eta_j^{(1)} = 0.01$, and variance $(\sigma^{(1)})^2 = 1.0$. KL region $\mathbb{S}_2$ is the remainder of the domain with a mean value of $E[Y^{(2)}] = 0.0$, $N_{\text{term}}(2) = 2 \times 1 = 2$ (with tensor grid) and $N_{\text{term}}(2) = 2 \times 2 = 4$

[4]These eigenvalue or eigenfunction computations are performed in each 1-D spatial dimension separately, and then multiplied together in a nondecreasing series. For this reason, in the numerical results we report the number of terms in each spatial dimension separately.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 2

Runtime and linear systems with the three collocation algorithms for Example 1. *Values in parenthesis denote the cost of the precomputation loop.*

$N_{\text{term}} = 11$, *Tensor product collocation,* $N_{\text{coll}} = 2$*: degree* $= 3$ *(*2048 *realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	542,498	208,896	55,296	(51,200)
Runtime in seconds	301.8	202.5	166.6	(11.5)

$N_{\text{term}} = 200$, *sparse grid collocation,* $\ell_{\max} = 1$*: degree* $= 3$ *(*401 *realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	35,082	26,466	25,954	(25,152)
Runtime in seconds	34.7	33.4	32.5	(5.5)

(with sparse grid) terms, correlation lengths $\eta_j^{(2)} = 0.1$, and variance $(\sigma^{(2)})^2 = 1.0$. Flow is induced from left to right with Dirichlet boundary conditions $g_D = 1$ on face $\{x = 0\}$ and $g_D = 0$ on face $\{x = 1\}$, and no-flow homogeneous Neumann boundary conditions on the other two edges.

The domain for Example 1 is divided into $N_D = 4 \times 4 = 16$ subdomains. Tensor product collocation uses a uniform spatial grid, with all subdomains containing 25×25 elements, and continuous linear mortars with 10 elements on all interfaces. Sparse grid collocation uses a nonuniform spatial grid such that subdomains in KL region $\mathbb{S}_1$ have 20×20 elements, and subdomains in KL region $\mathbb{S}_2$ have 4×4 elements. The interfaces are discretized with continuous linear mortars, with the number of elements on $\mathbb{S}_1 - \mathbb{S}_1$, $\mathbb{S}_1 - \mathbb{S}_2$, and $\mathbb{S}_2 - \mathbb{S}_2$ interfaces being 10, 4, and 2 elements, respectively.

Discussion. First we test isotropic tensor product collocation with a low number of terms. Using $N_{\text{coll}} = 2$ collocation points in $N_{\text{term}} = 9 + 2 = 11$ stochastic dimensions requires a total of $N_{\text{real}} = 2^{11} = 2048$ global realizations and a maximum number of $N_{\text{real}}(i) = 2^9 = 512$ local realizations, giving a global to local ratio of 4.0. Table 2 shows that the number of linear systems is reduced by 61% with a deterministic multiscale basis and by 90% with a stochastic multiscale basis. However, the runtime is reduced only by 33% and 45%, respectively. This is because the use of a multiscale basis in Methods S2 and S3 does nothing to reduce the interprocess communication during the CG iterations at each realization of the stochastic collocation loop. Notice that Method S3 took only 11.5 seconds to compute the stochastic multiscale basis, because the local systems are relatively small and easy to solve. Plots of the calculated statistics are shown in Figure 3.

Next we test sparse grid collocation with a large number of terms. Using a level $\ell_{\max} = 1$ sparse grid in $N_{\text{term}} = 196 + 4 = 200$ stochastic dimensions requires a total of $N_{\text{real}} = 401$ global realizations and a maximum number of $N_{\text{real}}(i) = 393$ local realizations, giving the much smaller global to local ratio of 1.02. In this case the number of linear systems is reduced by 25% with a deterministic multiscale basis and by 26% with a stochastic multiscale basis. The runtimes, however, remain nearly constant, since the interprocessor communication dominated the cost for solving the small subdoman problems. Plots of the calculated statistics are shown in Figure 4.

This example shows that for both tensor product and sparse grid collocation, the number of subdomain solves is reduced significantly by the deterministic multiscale basis in Method S2 and even further by the stochastic multiscale flux basis in Method S3. The gain from Method S3 is smaller for sparse grid collocation, due to smaller global to local ratio. In both cases the runtime is not reduced as much, since the communication cost is significant relative to the cost of solving subdomain

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

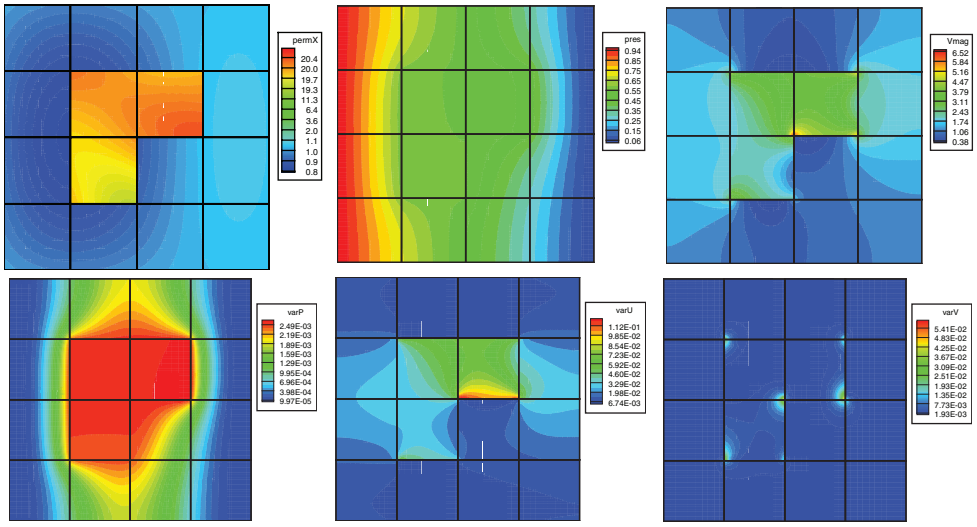

FIG. 3. *Realization of permeability (top left), mean pressure (top middle), mean velocity magnitude (top right), variance of pressure (bottom left), variance of horizontal velocity (bottom middle), and variance of vertical velocity (bottom right) for tensor product collocation in Example* 1.

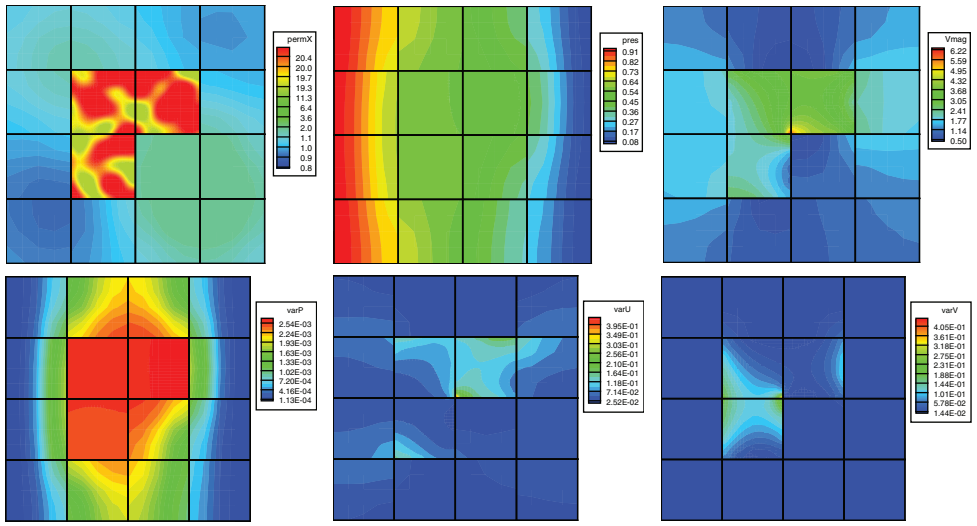

FIG. 4. *Realization of permeability (top left), mean pressure (top middle), mean velocity magnitude (top right), variance of pressure (bottom left), variance of horizontal velocity (bottom middle), and variance of vertical velocity (bottom right) for level* $\ell_{\max} = 1$ *sparse grid collocation in Example* 1.

problems, which are rather small in this example. One way to reduce the time spent on communication would be to use a preconditioner for the interface problem, which could be done in conjunction with the multiscale basis implementation.

Another observation is that third order accuracy with a tensor product grid on 10 stochastic dimensions requires 2048 realizations, while third order accuracy with a sparse grid on 200 stochastic dimensions requires only 401 realizations. It would not

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 3

Runtime and linear systems across refinement levels 1–4 *with the three collocation algorithms for Example* 2. *Values in parentheses denote the cost of the precomputation loop.*

$N_{\text{term}} = 12$, *Tensor product collocation,* $N_{\text{coll}} = 2$*: degree* $= 3$ *(*4096 *realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	3,722,250	655,360	35,200	(1,152)
Runtime in seconds	5,353	5,409	5,280	(0.2)

$N_{\text{term}} = 12$, *sparse grid collocation,* $\ell_{\max} = 2$*: degree* $= 5$ *(*361 *realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	341,836	57,760	11,552	(4,104)
Runtime in seconds	493.7	493.4	487.0	(0.5)

be possible to perform tensor product collocation in 200 dimensions because it would require over $1.6E60$ realizations.

6.2. Example 2: Checkerboard.

Description. This example demonstrates an adaptive procedure used to refine the spatial grid. There are $N_\Omega = 4$ KL regions on the domain $(0,1)^2$. The bottom-left and upper-right KL regions $\mathbb{S}^{(i)}, i = 1,4$ each have a mean value of $E[Y^{(i)}] = 4.6$, $N_{\text{term}}(i) = 2 \times 1 = 2$ terms, correlation lengths $\eta_j^{(i)} = 0.1$, and variance $(\sigma^{(i)})^2 = 1.0$. The top-left and bottom-right KL regions $\mathbb{S}^{(i)}, i = 2,3$ each have a mean value of $E[Y^{(i)}] = 0.0$, $N_{\text{term}}(i) = 2 \times 2 = 4$ terms, correlation lengths $\eta_j^{(i)} = 0.01$, and variance $(\sigma^{(i)})^2 = 100.0$. Tensor product collocation with $N_{\text{coll}} = 2$ and sparse grid collocation with level $\ell_{\max} = 2$ are performed in $N_{\text{term}} = 2+4+4+2 = 12$ stochastic dimensions, requiring 4096 and 361 global realizations, and 16 and 57 maximum local realizations, respectively, giving global to local ratios of 256.0 and 6.33, per mesh adaptation. Flow is induced from left to right with the same boundary conditions as in Example 1.

The adaptive procedure is as follows. The domain is divided into $N_D = 8 \times 8 = 64$ subdomains, and all interfaces are discretized with continuous linear mortars. On the coarsest level each subdomain has a 2×2 local grid, and each mortar has a single element. The stochastic collocation method is performed with Methods S1, S2, or S3 using this spatial grid. Upon completion of the collocation, a residual-based a posteriori error indicator developed in [36, 6] is computed using the expectation of the pressure together with the mean permeability. The spatial grids of subdomains that contain errors beyond a given tolerance are refined, as well as the mortars that touch those refined subdomains. At this point, the entire collocation is performed again using the new spatial grid. The procedure stops when no subdomain needs refinement.

Discussion. The low number of random dimensions allowed running both tensor product and sparse grid collocation on the same test. Table 3 shows their computational cost. Method S2 results in an 82%–83% decrease and Method S3 leads to a 93%–99% decrease in the number of linear systems required to solve the collocation on refinement levels 1–4 when compared to Method S1. Once again, the runtimes remain almost constant, because the small size of the linear systems keeps the problems communication bound.

Figure 5 shows the first four levels of spatial grid refinement in the adaptive procedure. The grids are similar to what one expects for a deterministic problem with the given mean permeability. Figures 6 and 7 demonstrate how both expectation and variance of pressure and velocity magnitude are improved on progressively finer spatial grids in the case of sparse grid collocation.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

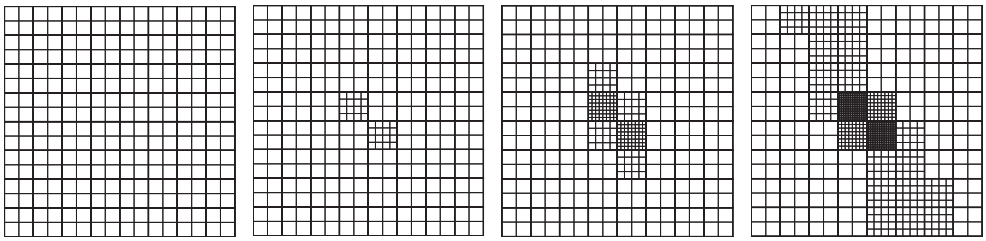

FIG. 5. *Spatial grids for refinement levels* 1–4 *with* $\ell_{\max} = 2$ *sparse grid in Example* 2.

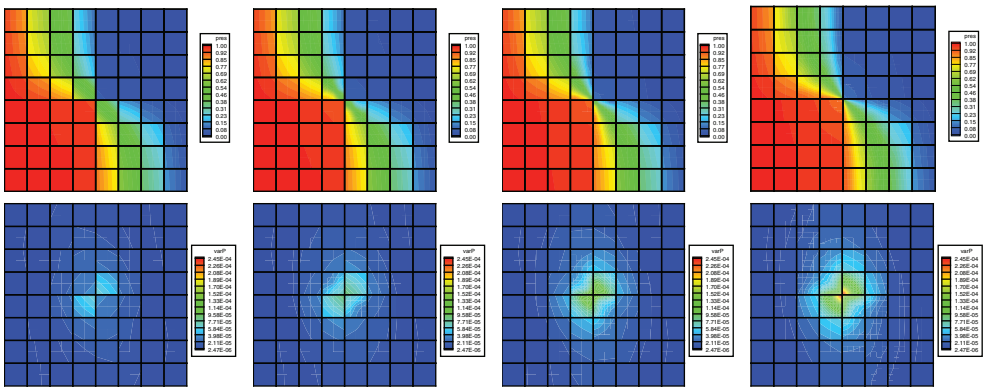

FIG. 6. *Mean pressure (top) and pressure variance (bottom) for refinement levels* 1–4 *with* $\ell_{\max} = 2$ *sparse grid in Example* 2.

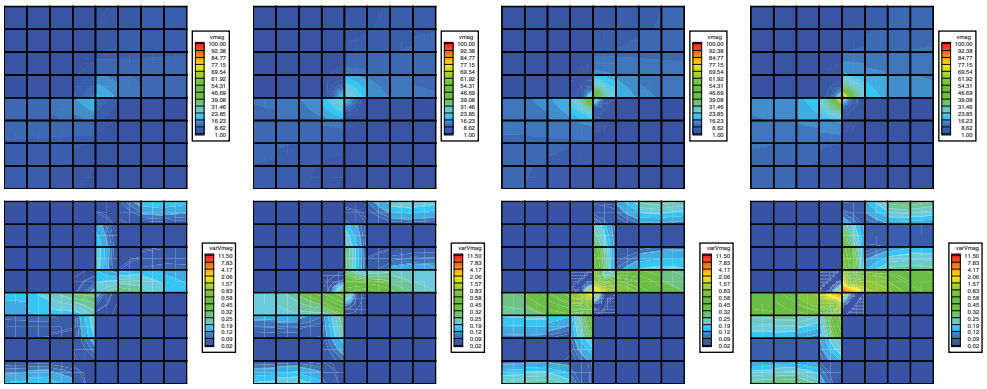

FIG. 7. *Mean velocity magnitude (top) and velocity magnitude variance (bottom) for refinement levels* 1–4 *with* $\ell_{\max} = 2$ *sparse grid in Example* 2.

6.3. Example 3: SPE10 benchmark.

Description. The mean permeability in the third example is a three-dimensional scalar field of actual geological measurements, obtained from the x component of the Society of Petroleum Engineers' (SPE) Comparative Solution Project.[5] It is a challenging benchmark problem with a Cartesian grid of $60 \times 220 \times 85$, giving a total of 1,122,000 finite elements; see Figure 8. This data set is part of a Brent sequence,

[5]For more information, see http://www.spe.org/csp.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

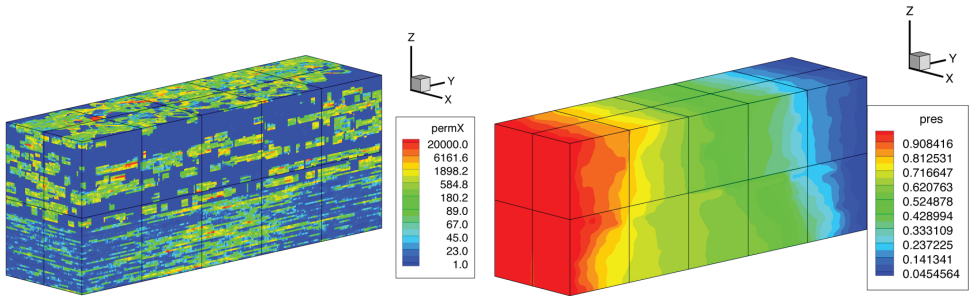

FIG. 8. *Permeability realization (left) and its corresponding solution (right) for Example* 3a.

with the lower 35 layers representing a prograding Tarbert formation, and the top 50 layers representing a fluvial Upper Ness formation. A flow is induced from front to back with Dirichlet boundary conditions $g_D = 1$ on face $\{y = 0\}$, $g_D = 0$ on face $\{y = 220\}$, and no-flow homogeneous Neumann boundary conditions on the other four faces.

The fine scale grid is broken up into $N_D = 2 \times 5 \times 2 = 20$ subdomains of nearly equal size. On the interfaces, the mortar space is composed of faces with linear mortars with a single 1×1 element.

In Example 3a, we perform tensor product collocation with $N_\Omega = 2$ statistically independent KL regions, roughly coinciding with the two geologic formations of the deterministic data. The first region includes the lower 10 subdomains and is described by the parameters: $N_{\text{term}}(1) = 1 \times 4 = 4$ terms, correlation lengths $\eta_j^{(1)} = 6.0$, and variance $(\sigma^{(1)})^2 = 1.7$. The second region includes the upper 10 subdomains and is described by the parameters: $N_{\text{term}}(2) = 1 \times 6 = 6$ terms, correlation lengths $\eta_j^{(2)} = 10.0$, and variance $(\sigma^{(2)})^2 = 1.2$. In Example 3b we switch to sparse grid collocation and increase the number of terms in the bottom and top KL regions to $N_{\text{term}}(1) = 4 \times 4 \times 4 = 64$ and $N_{\text{term}}(2) = 5 \times 5 \times 5 = 125$, respectively. In Example 3c we increase the number of KL regions to $N_\Omega = 20$, one in each subdomain, each with $N_{\text{term}}(i) = 2 \times 3 \times 2 = 12$ terms.

Discussion. In this 3-D benchmark problem, the size of the subdomain problems is sufficiently large so that the time spent solving a typical linear system dominates the time needed to perform interprocessor communication. Tensor product collocation with $N_{\text{coll}} = 2$ in $N_{\text{term}} = 4 + 6 = 10$ stochastic dimensions requires a total of $N_{\text{real}} = 2^{10} = 1024$ global realizations, and $N_{\text{real}}(i) = 64$ maximum local realizations, giving a global to local ratio of 16.0. Table 4 shows that the number of linear systems was reduced by 92% with a deterministic multiscale basis and 99% with a stochastic multiscale basis. Because of the sheer size of the subdomain problems, the runtime was also dramatically reduced by 85% and 89%, respectively. Figures 8–12 show the results of the computations.

In Example 3b, sparse grid collocation with $\ell_{\max} = 1$ in $N_{\text{term}} = 64 + 125 = 189$ stochastic dimensions requires a total of $N_{\text{real}} = 379$ global realizations, and $N_{\text{real}}(i) = 251$ maximum local realizations, giving a global to local ratio of 1.51. The number of linear systems is reduced by 91% and 94%. The runtime is reduced by 83% with deterministic multiscale basis, but is slightly worse with a stochastic multiscale basis with a reduction of 80%. Method S3 is faster than Method S2 in the tensor grid case but not in the sparse grid case due to different global to local ratios. When this ratio

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 4

Runtime and linear systems with the three collocation algorithms for Examples 3a *and* 3b *with* $N_\Omega = 2$ *KL regions. Values in parentheses denote the cost of the precomputation loop.*

$N_{\text{term}} = 10$, *tensor product collocation,* $N_{\text{coll}} = 2$ *(1024 realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	236,964	18,432	3,072	(1,024)
Runtime in hours	110.34	16.95	11.72	(0.82)

$N_{\text{term}} = 189$, *sparse grid collocation,* $\ell_{\max} = 1$ *(379 realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	79,047	6,822	4,774	(4,016)
Runtime in hours	37.16	6.27	7.33	(3.32)

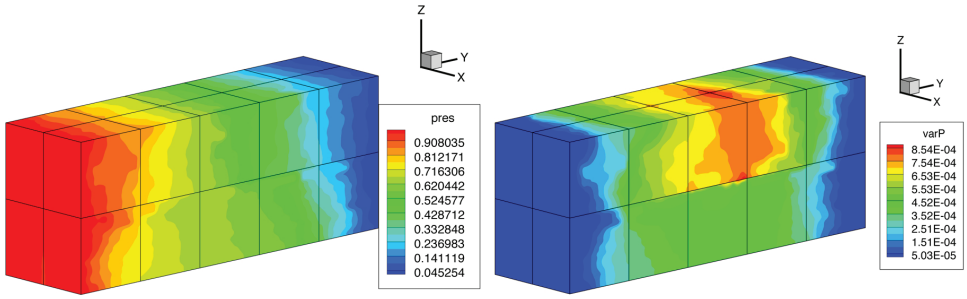

FIG. 9. *Mean solution (left) and pressure variance (right) for Example* 3a.

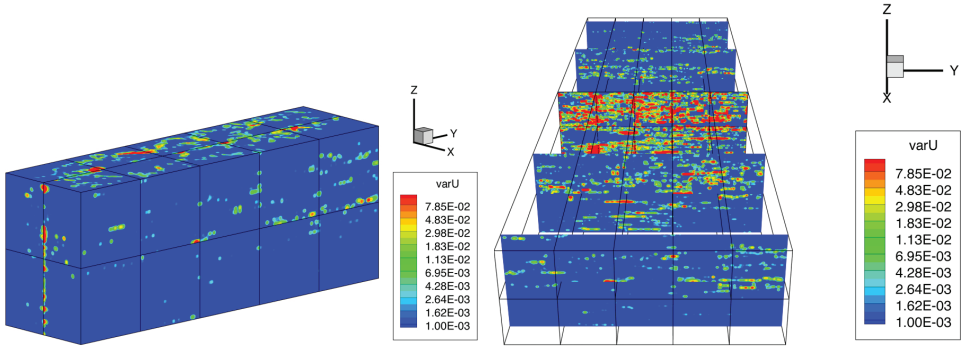

FIG. 10. *Velocity variance in x direction (left) with several cross sections (right) for Example* 3a.

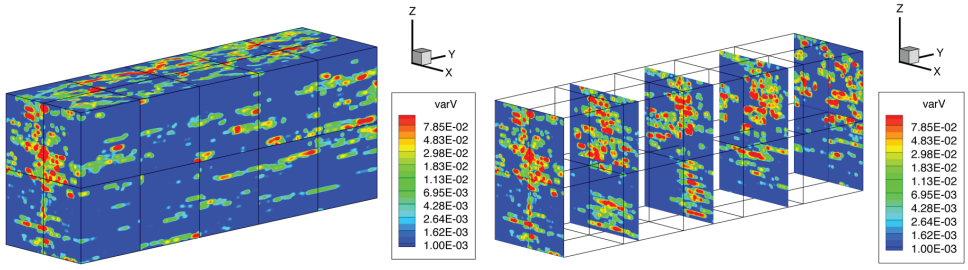

FIG. 11. *Velocity variance in y direction (left) with several cross sections (right) for Example* 3a.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

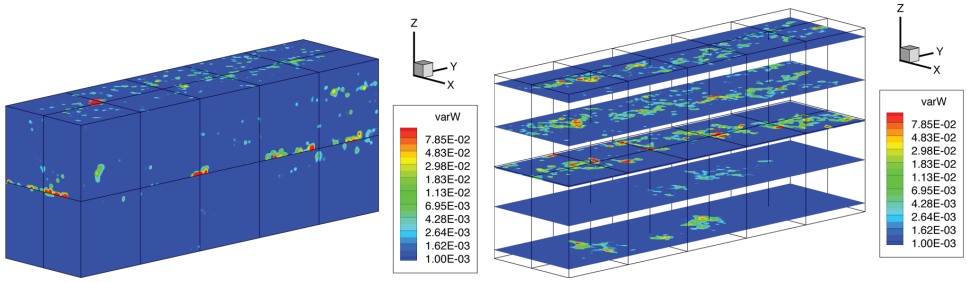

FIG. 12. *Velocity variance in z direction (left) with several cross sections (right) for Example* 3a.

TABLE 5

Runtime linear systems with the three collocation algorithms for Example 3c *with* $N_\Omega = 20$ *KL regions. Values in parentheses denote the cost of the precomputation loop.*

$N_{\text{term}} = 240$, *sparse grid collocation,* $\ell_{\max} = 1$ *(481 realizations)*

	Method S1	**Method S2**	**Method S3**	
Max. linear systems	101,826	8,658	1,362	(400)
Runtime in hours	47.8	7.97	5.50	(0.38)

is smaller, the stochastic multiscale basis is reused fewer times and the runtime for forming it becomes a factor. For instance, in Example 3b the precomputation loop is over 45% of the total runtime, while in Example 3a it is only 7%. The structure of a tensor product grid causes this ratio to remain very large, even when the difference between global and local dimensions is small.

Recall that the main benefit to using a sparse grid is that the number of points grows more modestly than a tensor grid as dimension increases. Unfortunately this means the global to local ratio is smaller, so Method S3 is faster than Method S2 only when the difference between global and local dimensions is large. Indeed, in Example 3c we show this effect, using $N_\Omega = 20$ KL regions each having $N_{\text{term}}(i) = 12$ dimensions. Sparse grid collocation with $\ell_{\max} = 1$ in $N_{\text{term}} = 12 * 20 = 240$ stochastic dimensions requires a total of $N_{\text{real}} = 481$ global realizations, and $N_{\text{real}}(i) = 41$ maximum local realizations, giving a global to local ratio of 19.0. The results are given in Table 5, and in this case Method S3 shows an improvement in runtime over Method S2. Both multiscale basis methods are still far superior to the traditional implementation.

6.4. Example 4: Convergence test.

Description. This example tests convergence rates in both stochastic and physical space. The goal is to confirm numerically the theoretical convergence rates established in section 4. There are $N_\Omega = 2$ KL regions on the domain $(0,1)^2$ with $N_D = 4 \times 4 = 16$ subdomains. A mean value of $E[Y] = 5000(1 - \sin(20x)\sin(20y))$ is used throughout the domain. KL region $\mathbb{S}^{(1)}$ is the left half of the domain with $N_{\text{term}}(1) = 2 \times 1 = 2$ terms, correlation length $\eta^{(1)} = 0.13$, and variance $(\sigma^{(1)})^2 = 1.0$. KL region $\mathbb{S}^{(2)}$ is the right half of the domain with $N_{\text{term}}(2) = 2 \times 2 = 4$ terms, correlation length $\eta^{(2)} = 0.09$, and variance $(\sigma^{(2)})^2 = 1.1$. The norms reported are of the form $\|E[u]\|_{L^2(D)}$, which is controlled by the norms bounded in the theory, since

$$\|E[u]\|_{L^2(D)} \leq \left(E\left[\|u\|^2_{L^2(D)}\right]\right)^{1/2}.$$

Discussion. Figure 13 shows convergence rates in stochastic space for four different sampling methods, wherein all tests have a fixed spatial grid. The subdomains

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

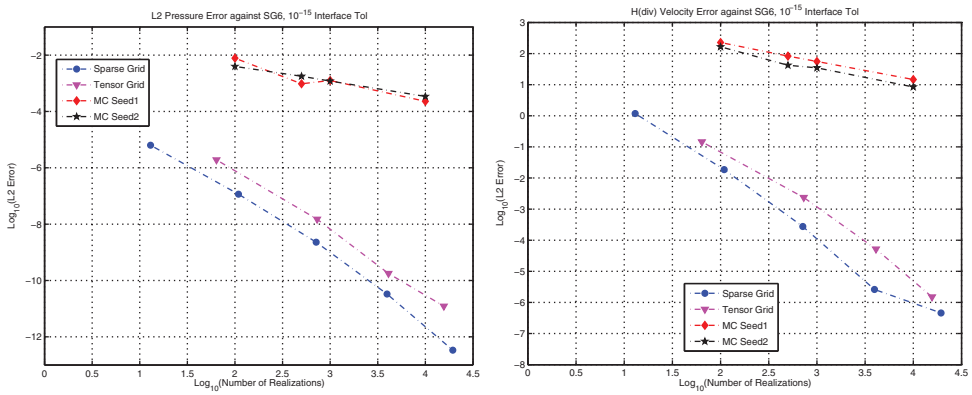

FIG. 13. *Log-log plot of convergence in stochastic space for Example* 4. *Different types of sampling methods are shown in absolute L^2-error for pressure (left) and H(div)-error for velocity (right).*

TABLE 6

Convergence in physical space for Example 4. *Relative errors reported against finest grid level; convergence rates given in parentheses.*

$\mathbb{S}^{(1)}$ Grid	Mort.	$\mathbb{S}^{(2)}$ Grid	$\frac{\|\|\|E[p-p_{true}]\|\|\|}{\|\|\|E[p_{true}]\|\|\|}$	$\frac{\|\|\|E[\mathbf{u}-\mathbf{u}_{true}]\|\|\|}{\|\|\|E[\mathbf{u}_{true}]\|\|\|}$	$\frac{\|\|\|E[\nabla\cdot(\mathbf{u}-\mathbf{u}_{true})]\|\|\|}{\|\|\|E[\nabla\cdot\mathbf{u}_{true}]\|\|\|}$
4×5	2	3×7	1.04E-02	1.76E-01	2.56E-01
8×10	4	6×14	3.93E-03 (1.40)	6.35E-02 (1.47)	1.02E-01 (1.33)
16×20	8	12×28	1.39E-03 (1.50)	1.71E-02 (1.89)	3.30E-02 (1.63)
32×40	16	24×56	3.74E-04 (1.90)	3.80E-03 (2.17)	1.14E-02 (1.53)
64×80	32	48×112	–	–	–

have 20×20 grids in the first region and 17×15 grids in the second region. Continuous linear mortars with 10 elements are used on all interfaces. The numerical solution with level 6 sparse grid is used as a "true" stochastic solution in the computation of the errors. The figures show log-log plots of absolute errors versus number of stochastic realizations. We compare sparse grid and tensor grid stochastic collocation to two Monte Carlo simulations with different random seeds. As expected, both collocation methods converge significantly faster than the Monte Carlo simulations, with the sparse grid being more accurate than the tensor product grid with the same number of realizations. Furthermore the error graphs for the two stochastic collocation methods confirm that the errors exhibit exponential convergence in stochastic space, as predicted by Theorem 4.4. The slight tapering off in the slope of the velocity error with sparse grid collocation is due to the effect of spatial discretization error.

Table 6 shows convergence rates in physical space. The L^2 errors are approximated by the midpoint rule; i.e., the element errors are evaluated at the cell centers. This approximation is denoted by $|||\cdot|||$. A level $\ell_{\max}=3$ sparse grid rule is used in stochastic dimensions, but we note that these results are within round-off from an isotropic tensor grid rule with $N_{\text{coll}}=3$. The first three columns show the refinements of spatial grids in each KL region and the mortar grid. Recall that the spatial discretization uses the lowest order Raviart–Thomas elements and linear mortars; therefore in Theorem 4.4, $q=t=\tilde{t}=1$ and $s=2$. Furthermore, $H\sim 2h$, so Theorem 4.4 predicts $O(h)$ convergence for the pressure and the velocity, and Theorem 4.5 predicts $O(h^2)$ convergence for the pressure at the cell centers. We do observe that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the convergence for the pressure approaches second order. We also observe $O(h^2)$ convergence for the velocity and $O(h^{3/2})$ convergence for its divergence. While for the sake of space we did not present a superconvergence estimate for the velocity, the results are consistent with Theorem 4.2 in [6] for the deterministic multiscale mortar mixed method.

7. Conclusions. Three methods are presented to quantify uncertainty for flow in nonstationary porous media that couple stochastic collocation with a mortar mixed finite element discretization. These methods are nonintrusive, requiring the solution of deterministic problems at specified collocation points, and are more efficient than Monte Carlo simulations. Method S1 uses the traditional implementation of the MMMFEM on each realization, Method S2 uses a deterministic multiscale flux basis on each realization, and Method S3 forms a stochastic multiscale flux basis across local realizations. A tensor product grid is suitable to handle relatively few random dimensions, while a sparse grid is necessary to handle a larger number of random dimensions.

We are able to draw three conclusions from the numerical examples. First, the computational workload in terms of the maximum number of linear systems solved by every subdomain is reduced by several orders of magnitude via the use of a deterministic multiscale basis, and it is further reduced via the use of a stochastic multiscale basis. Second, these savings do not always reduce runtimes because these techniques do not reduce the amount of interprocess communication. In order to see an improvement in runtime for the multiscale basis methods, the linear systems associated with the subdomain problems must be large enough to dominate the overhead in runtime associated with the interprocess communication during the interface iteration. Unpreconditioned multiscale basis techniques change a processor-laden simulation to a communication-laden simulation. A preconditioner could be used to reduce the amount of communication as well as the number of linear systems. Third, the ratio of global realizations to local realizations influences whether Method S3 is faster than Method S2. The smaller this ratio is, the larger is the relative cost of the precomputation loop. A tensor product grid inherently has a large ratio, while a sparse grid has a large ratio only when the difference between global and local dimensions is large.

Acknowledgments. The authors would like to thank the Institute for Computational Engineering and Sciences at The University of Texas at Austin for the use of their computing resources.

REFERENCES

[1] J.E. Aarnes and Y. Efendiev, *Mixed multiscale finite element methods for stochastic porous media flows*, SIAM J. Sci. Comput., 30 (2008), pp. 2319–2339.

[2] J.E. Aarnes, S. Krogstad, and K.-A. Lie, *A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids*, Multiscale Model. Simul., 5 (2006), pp. 337–363.

[3] M. Abramowitz and I.A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed., Dover, New York, 1972.

[4] T. Arbogast, *Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems*, SIAM J. Numer. Anal., 42 (2004), pp. 576–598.

[5] T. Arbogast, L.C. Cowsar, M.F. Wheeler, and I. Yotov, *Mixed finite element methods on nonmatching multiblock grids*, SIAM J. Numer. Anal., 37 (2000), pp. 1295–1315.

[6] T. Arbogast, G. Pencheva, M.F. Wheeler, and I. Yotov, *A multiscale mortar mixed finite element method*, Multiscale Model. Simul., 6 (2007), pp. 319–346.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[7] B.V. Asokan and N. Zabaras, *A stochastic variational multiscale method for diffusion in heterogeneous random media*, J. Comput. Phys., 218 (2006), pp. 654–676.
[8] 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 (2008), pp. 1005–1034.
[9] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
[10] H. Chang and D. Zhang, *A comparative study of stochastic collocation methods for flow in spatially correlated random fields*, Commun. Comput. Phys., 6 (2009), pp. 509–535.
[11] Z. Chen and T.Y. Hou, *A mixed multiscale finite element method for elliptic problems with oscillating coefficients*, Math. Comp., 72 (2003), pp. 541–576.
[12] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
[13] L.C. Cowsar, C. Woodward, and I. Yotov, *PARCEL v*1.04 *user guide*, Technical report 96-28, TICAM, University of Texas at Austin, 1996.
[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 (2001), pp. 6359–6372.
[15] P. Dostert, Y. Efendiev, and T.Y. Hou, *Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty quantification*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 3445–3455.
[16] G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer, Berlin, 1996.
[17] G.P. Galdi, *An Introduction to the Mathematical Theory of the Navier-Stokes Equations. Vol.* I. *Linearized Steady Problems*, Springer-Verlag, New York, 1994.
[18] B. Ganapathysubramanian and N. Zabaras, *Modelling diffusion in random heterogeneous media: Data-driven models, stochastic collocation and the variational multi-scale method*, J. Comput. Phys., 226 (2007), pp. 326–353.
[19] B. Ganis, H. Klie, M.F. Wheeler, T. Wildey, I. Yotov, and D. Zhang, *Stochastic collocation and mixed finite elements for flow in porous media*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 3547–3559.
[20] B. Ganis and I. Yotov, *Implementation of a mortar mixed finite element method using a multiscale flux basis*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 3989–3998.
[21] R.G. Ghanem and P.D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.
[22] R. Glowinski and M.F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, in Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, 1988.
[23] P. Grisvard, *Elliptic problems in nonsmooth domains*, Monogr. Stud. Math. 24, Pitman, Boston, 1985.
[24] T.Y. Hou and X.H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comput. Phys., 134 (1997), pp. 169–189.
[25] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.B. Quincy, *The variational multiscale method—a paradigm for computational mechanics*, Comput. Methods Appl. Mech. Engrg., 166 (1998), pp. 3–24.
[26] C. Jin, X.-C. Cai, and C. Li, *Parallel domain decomposition methods for stochastic elliptic equations*, SIAM J. Sci. Comput., 29 (2008), pp. 2096–2114.
[27] Z. Lu and D. Zhang, *Stochastic simulations for flow in nonstationary randomly heterogeneous porous media using a KL-based moment-equation approach*, Multiscale Model. Simul., 6 (2007), pp. 228–245.
[28] 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 (2008), pp. 2309–2345.
[29] G. Pencheva and I. Yotov, *Balancing domain decomposition for mortar mixed finite element methods on non-matching grids*, Numer. Linear Algebra Appl., 10 (2003), pp. 159–180.
[30] A.M. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, New York, 2008.
[31] R.A. Raviart and J.M. Thomas, *A mixed finite element method for* 2*nd order elliptic problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Math. 606, Springer-Verlag, New York, 1977, pp. 292–315.
[32] L.R. Scott and S. Zhang, *Finite element interpolation of nonsmooth functions satisfying boundary conditions*, Math. Comp., 54 (1990), pp. 483–493.
[33] S.A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, in Dokl. Akad. Nauk, 4 (1963), p. 12.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[34] G. Stefanou, *The stochastic finite element method: Past, present and future*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 1031–1051.
[35] M.F. Wheeler, G. Xue, and I. Yotov, *A multiscale mortar multipoint flux mixed finite element method*, Technical report 10-33, ICES Report, The University of Texas at Austin, 2010.
[36] M.F. Wheeler and I. Yotov, *A posteriori error estimates for the mortar mixed finite element method*, SIAM J. Numer. Anal., 43 (2005), pp. 1021–1042.
[37] M. F. Wheeler, T. Wildey, and I. Yotov, *A multiscale preconditioner for stochastic mortar mixed finite elements*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 1251–1262.
[38] C. L. Winter and D. M. Tartakovsky, *Groundwater flow in heterogeneous composite aquifers*, Water Resour. Res., 38 (2002).
[39] D. Xiu and J.S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[40] D. Xiu and G.E. Karniadakis, *Modeling uncertainty in steady state diffusion problem via generalized polynomial chaos*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4927–4948.
[41] D. Zhang, *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*, Academic Press, San Diego, CA, 2002.
[42] D. Zhang and Z. Lu, *An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions*, J. Comput. Phys., 194 (2004), pp. 773–794.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.