



Efficient stochastic finite element methods for flow in heterogeneous porous media. Part 2: Random lognormal permeability

Luca Traverso^{1,2} | Timothy Nigel Phillips² 

¹School of Earth and Ocean Sciences,
Cardiff University, Cardiff, UK

²School of Mathematics, Cardiff
University, Cardiff, UK

Correspondence

T. N. Phillips, School of Mathematics,
Cardiff University, Cardiff CF24 4AG, UK.
Email: PhillipsTN@cf.ac.uk

Summary

Efficient and robust iterative methods are developed for solving the linear systems of equations arising from stochastic finite element methods for single phase fluid flow in porous media. Permeability is assumed to vary randomly in space according to some given correlation function. In the companion paper, herein referred to as Part 1, permeability was approximated using a truncated Karhunen-Loève expansion (KLE). The stochastic variability of permeability is modeled using lognormal random fields and the truncated KLE is projected onto a polynomial chaos basis. This results in a stochastic nonlinear problem since the random fields are represented using polynomial chaos containing terms that are generally nonlinear in the random variables. Symmetric block Gauss-Seidel used as a preconditioner for CG is shown to be efficient and robust for stochastic finite element method.

KEYWORDS

AMG, conjugate gradient, groundwater flow, MINRES, permeability, preconditioners, stochastic finite element method, stochastic mixed finite element method

1 | INTRODUCTION

This article is concerned with the numerical simulation of flow through porous media. This is an example of a diffusion problem in which the properties of the medium, such as the permeability, are not known in any precise way due to the paucity and accuracy of available measurements. Quantifying the uncertainty of the model parameters and modeling them as random variables means that the governing partial differential equation (PDE) is stochastic rather than deterministic. When the stochastic PDE (SPDE) is equipped with suitable boundary conditions, which can be defined as stochastic processes, then its solution is also a stochastic process. In this case the mean and variance provide useful statistical information about the solution.

One of the most popular approaches for solving SPDEs is the Monte Carlo Method (MCM). This approach is based on selecting an ensemble of realizations for the random input parameters from a given distribution. A deterministic problem corresponding to each realization from the ensemble is solved and statistical quantities are computed from the set of solutions. This approach is simple to implement since it can be based on the use of existing PDE solvers. However, MCM is computationally very expensive since it converges sublinearly at a rate independent of the stochastic dimension.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. *International Journal for Numerical Methods in Fluids* published by John Wiley & Sons Ltd.

Consequently, MCM requires a large number of realizations to create meaningful statistics. This limitation has motivated researchers to investigate acceleration techniques for MCM to improve its performance.

Multilevel Monte Carlo combines multigrid concepts with traditional MCM. The acceleration in convergence is guaranteed as most of the MCM simulations are carried out on coarse grids while only a very limited amount of time is spent on finer grids. When applied to the solution of PDEs^{1,2} multilevel MC methods have been shown to be incredibly efficient for problems with rough coefficients (ie, spatial random fields with large variance or small correlation lengths). These types of problems, common to radioactive waste disposal applications, require a large number of random variables (typically in excess of 100 modes in a Karhunen-Loève expansion [KLE]) in probability space to accurately represent the variability of the spatial random field.

An alternative class of methods to MCM is based on the discretization of SPDEs by the stochastic finite element method (SFEM). However, in contrast to MCM, this approach can be computationally expensive since the discrete system is significantly larger with the dimension of the problem growing factorially with the number of random variables used to describe the input spatial random field. This phenomenon is known as the curse of dimensionality.

Solution strategies for SFEM depend on the choice of basis functions for the discretization of the stochastic space. There are two popular choices. The first choice uses global tensor product polynomials.³ The advantage of this approach is that it allows the global Galerkin system to be decoupled. However, this is restricted to problems for which the permeability is approximated by normal or uniform random fields.⁴ There is no evidence that when the permeability is approximated by a lognormal random field (a very common assumption in the groundwater modeling community), the global Galerkin system can be decoupled. Furthermore, this approach has the disadvantage of the size of the stochastic space growing more rapidly than alternative methods such as polynomial chaos. Solution strategies for this choice involve iterative solvers based on Krylov subspace recycling techniques.⁴

The second choice is based on the polynomial chaos (PC) method as outlined in the original work of Ghanem and Spanos.^{5,6} In this approach, spectral representations of uncertainty in terms of multi-dimensional Hermite polynomials or polynomial chaos expansions are used to approximate both the model parameters and the solution. This enables the stochastic equations to be replaced by deterministic systems of PDEs which are then truncated and discretized. The original PC method was motivated by the Wiener chaos expansion in which Hermite polynomials are used to represent Gaussian random processes. The PC approach was later extended to generalized PC (gPC) where other sets of orthogonal polynomials are used to generate improved representations of more general random processes. This allows efficient methods to be constructed for problems for which the uncertainty in the model parameters cannot be represented using Gaussian random processes. The PC expansion is a projection of the input random variables onto the space spanned by the orthogonal polynomials. Therefore, the rate of convergence depends on the smoothness of the solution as a function of the input random variables. The optimal choice of polynomials, in the L^2 sense, is the set that is orthogonal with respect to the probability density function of the random variables that appear in the input variables. As MCM requires the solution of many PDEs over the same computational domain, PC methods generate large coupled systems of equations.

The large and highly structured linear system generated by the PC approach has to be solved using Krylov subspace iterative solvers. An efficient implementation of SFEM, which does not require the assembly of the global stiffness matrix uses a block-diagonal preconditioner (subsequently referred to as “mean-based preconditioner”) for CG based on an incomplete factorization of the mean stiffness matrix.^{7,8} Powell and Elman⁹ replaced the incomplete factorization with a black-box algebraic multigrid (AMG) solver. Ernst et al¹⁰ extended the implementation of the mean-based preconditioner to the solution of stochastic mixed finite element method (SMFEM) systems. Ullmann⁴ proposed a Kronecker product preconditioner for the stochastic linear (Gaussian/uniform random fields) and nonlinear (lognormal random field) cases. The implementation of the Kronecker preconditioner was recently extended to the SMFEM¹¹ and significantly reduced the number of CG and MINRES iterations resulting in faster convergence. However, it is more expensive to implement than mean-based preconditioners. A review of a large number of iterative solvers, including one-level iterative methods, multigrid methods, and multilevel methods (for the stochastic discretization), can be found in Rossell and Vanderwalle.¹²

In this article, the PC approach is adopted. The derivation of the global Galerkin system is described and solution strategies that take full advantage of its characteristic block structure are proposed. Numerical experiments in Part 1 showed that a symmetric block Gauss-Seidel preconditioner for CG provides a competitive alternative to traditional mean-based preconditioners. Since information associated with the off-diagonal blocks of A is incorporated into the preconditioned system, the conditioning is improved. As a result, CG requires few iterations to converge. This approach is

particularly efficient for those cases in which the off-diagonal blocks of A hold significant information on the permeability, that is, problems with large values of σ .

It is evident from the literature and from our computational analysis that mean-based preconditioners cannot be robust with respect to the permeability since they only include, in the preconditioned system, information associated with the mean value of the spatial random field. The mean information is included in the blocks of the leading diagonal of the global stochastic system, whilst variations (representing the variability of the spatial random field) about the mean are contained in the off-diagonal blocks. When the latter contributions become important the mean-based preconditioner performs poorly simply because this information is not included in the preconditioned system. For the stochastically nonlinear case this situation is exacerbated by the fact that every block of the global system has non-zero entries.

To overcome this important limitation, we propose an alternative preconditioner for SFEM in which the off-diagonal blocks of the global system are included in the preconditioned system using a block symmetric Gauss-Seidel (bSGS) algorithm. The computational analysis clearly shows that block Gauss-Seidel algorithms used either as a preconditioner for CG or as standalone solvers are more efficient than mean based preconditioners for the stochastically nonlinear case and deliver considerable CPU savings. Although for some of the test cases considered, the standalone standard Gauss-Seidel solver was the best performing solver, its performance deteriorated at a faster rate for test cases with large standard deviation than preconditioned CG. Therefore, the main finding of this work is that CG equipped with a bSGS preconditioner should be used to solve SFEM systems.

2 | THE MATHEMATICAL MODEL

The steady-state flow of water in porous media, whose material parameters are assumed to be unknown, is described by a scalar second-order stochastic partial differential equation (SPDE). The parameter that is known with the greatest degree of uncertainty is the permeability or hydraulic conductivity, C , which describes the ease with which a fluid can move through pore spaces or fractures. The permeability is modeled as a random field.

Let u denote the pressure head or mean potential and \mathbf{q} the velocity or flux. We assume that the medium occupies a bounded domain D in \mathbb{R}^2 , with Lipschitz continuous boundary Γ . Let $\Gamma = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \neq \emptyset$ and $\Gamma_D \cap \Gamma_N = \emptyset$, where Γ_D and Γ_N denote the portions of Γ where Dirichlet and Neumann boundary conditions on u are prescribed, respectively. The tensor C is a 2×2 symmetric and uniformly positive definite tensor that represents permeability. Since the coefficients in this tensor can never be completely known at every point in a heterogeneous porous medium, we assume that $C = C(\mathbf{x}, \omega)$ is a random field. Furthermore, only statistical properties of C are assumed.

Let Ω be the set of random events, \mathfrak{F} the minimal σ -algebra of the subsets of Ω , and P an appropriate probability measure, then $(\Omega, \mathfrak{F}, P)$ denotes a probability space. Then $C(\mathbf{x}, \omega) : D \times \Omega \rightarrow \mathbb{R}$ is a family of random fields indexed by \mathbf{x} such that for a fixed $\mathbf{x} \in D$, $C(\cdot, \omega)$ is a realization of the permeability and for a fixed realization $\omega \in \Omega$, $C(\mathbf{x}, \cdot)$ is a random variable.

We define the space of square integrable random variables with respect to P , $L_p^2(\Omega)$, as follows:

$$L_p^2(\Omega) = \{\xi : \Omega \rightarrow \mathbb{R}; \langle \xi^2 \rangle < \infty\},$$

where

$$\langle \xi^2 \rangle = \int_{\mathbb{R}} \xi^2(\omega) dP(\omega).$$

Let ξ be a real-valued random variable belonging to $(\Omega, \mathfrak{F}, P)$ and suppose there exists a probability density function $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that the expected value of ξ is given by

$$\langle \xi \rangle = \int_{\mathbb{R}} \xi(\omega) \rho(\omega) d\omega.$$

The mean value of the random field C at $\mathbf{x} \in D$ is $\langle C(\mathbf{x}, \cdot) \rangle$, the covariance of C at $\mathbf{x}, \mathbf{y} \in D$ is

$$\text{cov}(x, y) = \langle (C(\mathbf{x}, \omega) - \langle C(\mathbf{x}) \rangle)(C(\mathbf{y}, \omega) - \langle C(\mathbf{y}) \rangle) \rangle = \sigma^\epsilon \rho(\mathbf{x}, \mathbf{y}),$$

and the standard deviation is $\sigma(\mathbf{x}) = \sqrt{\text{var}(\mathbf{x})}$ where $\text{var}(\mathbf{x}) = \text{cov}(\mathbf{x}, \mathbf{x})$.

2.1 | Primal formulation

In this formulation, we seek a random field solution $u(\mathbf{x}, \omega) : D \times \Omega \rightarrow \mathbb{R}$ such that P -almost surely

$$\begin{aligned} -\nabla \cdot C(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega) &= f(\mathbf{x}) & \mathbf{x} \in D, \\ u(\mathbf{x}, \omega) &= g(\mathbf{x}) & \mathbf{x} \in \Gamma_D, \\ C(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega) \cdot \mathbf{n} &= 0 & \mathbf{x} \in \Gamma_N, \end{aligned} \tag{1}$$

where \mathbf{n} denotes the unit outward normal vector to Γ_N , $g(\mathbf{x})$ is the prescribed constant pressure head on Γ_D , and $f(\mathbf{x})$ is the source or sink term. Note that $f(\mathbf{x})$ and $g(\mathbf{x})$ could also be random fields but in this paper they are assumed to be deterministic functions. The solution of (1) enables the mean and standard deviation of u to be determined everywhere in D . Once u has been determined, the velocity field \mathbf{q} can be derived using Darcy's Law.

Define the solution space W and the test space W_0 to be the tensor product spaces of deterministic functions defined on D and stochastic functions defined on the probability space as follows

$$W = X \otimes L^2_P(\Omega), \quad W_0 = X_0 \otimes L^2_P(\Omega), \tag{2}$$

respectively, where

$$X = \{v \in H^1(D) : v = g \text{ on } \Gamma_D\}, \quad X_0 = \{v \in H^1(D) : v = 0 \text{ on } \Gamma_D\}.$$

Then the weak formulation of the primal problem is: find $u \in W$ such that

$$\langle a(u, w) \rangle = \langle L(w) \rangle, \quad \forall w \in W_0 \tag{3}$$

where

$$\begin{aligned} \langle a(u, w) \rangle &= \left\langle \int_D C(\mathbf{x}, \xi) \nabla u(\mathbf{x}, \xi) \cdot \nabla w(\mathbf{x}, \xi) d\mathbf{x} \right\rangle, \\ \langle L(w) \rangle &= \left\langle \int_D f(\mathbf{x}) w(\mathbf{x}, \xi) d\mathbf{x} \right\rangle. \end{aligned} \tag{4}$$

If the permeability tensor is strictly positive and bounded, that is,

$$0 < k_1 \leq C(\mathbf{x}, \omega) \leq k_2 < \infty \text{ a.e. in } D \times \Omega, \tag{5}$$

then the *Lax-Milgram* lemma can be used to establish that there exists a unique solution to the problem (3).

2.2 | Mixed formulation

An alternative formulation that allows us to derive accurate approximations for the fluxes is to reformulate the primal problem (1) as a first-order system by explicitly introducing Darcy's Law. In this formulation, the solution $u(\mathbf{x}, \omega) : D \times \Omega \rightarrow \mathbb{R}$, $\mathbf{q}(\mathbf{x}, \omega) : D \times \Omega \rightarrow \mathbb{R}^2$ is sought of the problem

$$\begin{aligned} C^{-1}(\mathbf{x}, \omega) \mathbf{q}(\mathbf{x}, \omega) + \nabla u(\mathbf{x}, \omega) &= 0 & \mathbf{x} \in D, \\ \nabla \cdot \mathbf{q}(\mathbf{x}, \omega) &= f(\mathbf{x}) & \mathbf{x} \in D, \\ u(\mathbf{x}, \omega) &= g(\mathbf{x}) & \mathbf{x} \in \Gamma_D, \\ \mathbf{q}(\mathbf{x}, \omega) \cdot \mathbf{n} &= 0 & \mathbf{x} \in \Gamma_N. \end{aligned} \tag{6}$$

The solution of (6) gives the mean potential (or pressure) and normal fluxes together with information about their mean and standard deviation everywhere in D .

The solution spaces are the tensor product spaces $Y = L^2(D) \otimes L_p^2(\Omega)$ and

$$V = \{\mathbf{v} \in H(\text{div}; D) \otimes L_p^2(\Omega) : \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma_N \times \Omega\}, \quad (7)$$

where

$$H(\text{div}; D) = \{\mathbf{v} : \mathbf{v} \in L^2(D)^2, \text{ and } \nabla \cdot \mathbf{v} \in L^2(D)\}. \quad (8)$$

The weak formulation of the mixed problem is: find $(u, \mathbf{q}) \in Y \times V$ such that

$$\begin{aligned} \langle a(\mathbf{q}, \mathbf{v}) \rangle + \langle b(\mathbf{v}, u) \rangle &= \langle (g, \mathbf{n} \cdot \mathbf{v})_{\Gamma_D} \rangle, \quad \forall \mathbf{v} \in V \\ \langle b(\mathbf{q}, w) \rangle &= -\langle (f, w) \rangle, \quad \forall w \in Y \end{aligned} \quad (9)$$

where

$$\begin{aligned} \langle a(\mathbf{q}, \mathbf{v}) \rangle &= \left\langle \int_D C^{-1}(\mathbf{x}, \xi) \mathbf{q}(\mathbf{x}, \xi) \cdot \mathbf{v}(\mathbf{x}, \xi) d\mathbf{x} \right\rangle, \\ \langle b(\mathbf{v}, w) \rangle &= \left\langle \int_D \nabla \cdot \mathbf{v}(\mathbf{x}, \xi) w(\mathbf{x}, \xi) d\mathbf{x} \right\rangle, \\ \langle (g, \mathbf{n} \cdot \mathbf{v})_{\Gamma_D} \rangle &= \left\langle \int_{\Gamma_D} g(\mathbf{x}) (\mathbf{n} \cdot \mathbf{v}(\mathbf{x}, \xi)) d\mathbf{x} \right\rangle, \\ \langle (f, w) \rangle &= \left\langle \int_D f(\mathbf{x}) w(\mathbf{x}, \xi) d\mathbf{x} \right\rangle. \end{aligned} \quad (10)$$

There exists a unique solution to this problem provided that the bilinear forms are continuous and coercive and the *inf-sup* inequality is satisfied.¹³ The permeability field is required to satisfy (5).

3 | PERMEABILITY APPROXIMATION

To convert the stochastic primal and mixed formulations into deterministic problems, we need to represent the stochastic variability of the permeability tensor $C(\mathbf{x}, \omega)$ by an appropriate set of independent random variables $\{\xi_1(\omega), \dots, \xi_d(\omega)\}$. In Part 1,¹⁴ two approaches were described to represent the stochastic variability of $C(\mathbf{x}, \omega)$. The first approach, herein referred to as *coloured noise*, assumes that the permeability varies randomly throughout D according to a given correlation function. In this case, the permeability coefficient is approximated using a KLE.^{6,9,15,16}

The permeability tensor $C(\mathbf{x}, \omega)$ possesses a proper orthogonal decomposition

$$C(\mathbf{x}, \xi(\omega)) = \mu(\mathbf{x}) + \sigma \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i \beta_i(\mathbf{x}), \quad (11)$$

where $\mu(\mathbf{x}) = \langle C(\mathbf{x}, \omega) \rangle$ and $\{\lambda_i, \beta_i(\mathbf{x})\}$ is the set of eigenvalues and eigenvectors of the covariance function $\rho(\mathbf{x}, \mathbf{y})$:

$$\int_D \rho(\mathbf{x}, \mathbf{x}') \beta_i(\mathbf{x}') d\mathbf{x}' = \lambda_i \beta_i(\mathbf{x}), \quad i = 1, \dots \quad (12)$$

Since C is nonnegative definite, the eigenvalues are real and we label them in decreasing magnitude $\lambda_1 > \lambda_2 > \dots$.

When the correlation function is of exponential type and D is a rectangular domain, there exists closed-form solutions to the eigenvalue problem (12) (see Ghanem and Spanos,⁶ for example). In this article, we make full use of the closed-form solutions and only random fields whose correlation function is of exponential or square-exponential type are considered. Examples in which the eigenvalue problem is solved numerically can be found in Lu and Zhang¹⁷ and a description of numerical algorithms is provided in Ghanem and Spanos.⁶ In such cases, the additional computational cost of solving the eigenproblem (12) needs to be considered.

In the coloured noise approach, $C(\mathbf{x}, \omega)$ is approximated by a truncated KLE, $C_d(\mathbf{x}, \xi(\omega))$, where d is the number of terms retained in the the KLE (11). The quadratic mean square convergence of $C_d(\mathbf{x}, \xi(\omega))$ to $C(\mathbf{x}, \xi(\omega))$ is guaranteed as $d \rightarrow \infty$. The choice of d usually depends on the rate of decay of the eigenvalues. For example, in applications where the eigenvalues decay slowly due to small correlation lengths, d may have to be very large.

Xiu and Karniadakis¹⁸ used uniform random variables, hence ensuring that $C_d(\mathbf{x}, \xi(\omega))$ is bounded between two positive values with probability 1. A consequence of this approach is that the random variables in (11) are not guaranteed to be independent, thus this condition needs to be assumed explicitly.¹⁹

The second approach, herein referred to as *white noise*, assumes that the permeability varies randomly and independently in D . This approach is often used to approximate parameters such as rainfall or groundwater recharge, which generally do not show strong spatial correlation. Although the permeability is spatially correlated, in practical applications, the domain D can be decomposed into subdomains corresponding to hydrogeological units on which piecewise constant hydraulic properties are assumed to hold. Different statistical parameters can be assigned to different regions of D , thus reflecting the diverse hydraulic behavior of natural deposits. A coarse subdomain decomposition of D is performed and a continuous random field C_{d_k} is associated with each subdomain $D_k, k = 1, \dots, N_D$, such that

$$C_d(\mathbf{x}, \xi(\omega)) = \bigcup_{k=1}^{N_D} C_{d_k}(\mathbf{x}, \xi(\omega)),$$

where N_D is the number of subdomains in D . Each subdomain D_k , which may be of irregular shape, is enclosed within a rectangularly shaped domain D'_k , that is, $D_k \subset D'_k$, for $k = 1, \dots, N_D$, chosen to be the smallest rectangle enclosing D_k . Although a KLE is implemented for each subdomain D_k , the eigenvalue problem (12) is solved with respect to D'_k .

When the conductivity coefficient $C(\cdot, \omega)$ is assumed to be a Gaussian process, the random variables ξ_i in (11) are normally distributed. In these circumstances, the random variables have the desirable property of being uncorrelated and independent. However, this also makes problems (1) and (6) ill-posed since the diffusion coefficient is not bounded below and above by positive constants.⁹ In fact, it is well known that the permeability is required to satisfy (5). Although Gaussian functions possess an infinite spectrum, it can be shown that well-defined discrete solutions can be obtained if a relatively small variance is used.

From a mathematical point of view, the white noise approach possesses significant advantages with respect to single domain KLE-based approaches. For example, the linear systems are tridiagonal,²⁰ which means that block diagonal preconditioners can be constructed to solve these problems efficiently.

4 | POLYNOMIAL CHAOS FOR LOGNORMAL RANDOM FIELD

Although Gaussian processes are commonly used to model uncertainty in engineering problems primarily due to their simplicity, in some cases, it is preferable to use a lognormal process, particularly when the quantity under consideration is constrained to be always positive. This is true when modeling the permeability or hydraulic conductivity of a porous medium. Lognormal random fields are very popular among physical scientists and modelers for various reasons. First, there are several studies, the data of which are summarized in References 21 and 22, that show that parameters such as hydraulic conductivity or transmissivity are often lognormally distributed. Second, although the lognormal distribution possesses an infinite upper bound, it only admits the positive part of the physical spectrum. This is obviously consistent with the physical properties of these parameters. There is substantial evidence, direct and indirect, that supports the view that the permeability is described by a lognormal distribution (Freeze²³).

The random field $C_d(\mathbf{x}, \xi(\omega))$ is a lognormal random field if the logarithm of C_d is a Gaussian random field, that is, lognormal random fields are of the form $\exp(g(\mathbf{x}, \xi))$ where g is a Gaussian random field. Lognormal random fields can be characterized by defining the mean field and covariance of the underlying Gaussian random field.

In Section 3, we have seen that a Gaussian random field can be approximated by a truncated Karhunen-Loèaussian process

$$\mathcal{L}(\mathbf{x}, \xi(\omega)) = \exp(C(\mathbf{x}, \xi(\omega))). \tag{13}$$

If we discretize $C(\mathbf{x}, \xi(\omega))$ using the truncated KLE, $C_d(\mathbf{x}, \xi(\omega))$, then

$$\mathcal{L}_d(\mathbf{x}, \xi(\omega)) = \exp(C_d(\mathbf{x}, \xi(\omega))). \tag{14}$$

There are advantages to using truncated KLE in polynomial chaos. For example, the optimal mean square convergence property of the KLE mentioned earlier means that the number of variables required for a given accuracy is minimized, hence limiting the system size.

Since $\mathcal{L}_d(\mathbf{x}, \xi(\omega))$ is a random variable, it can be projected onto a polynomial chaos basis of order p

$$\mathcal{L}_d(\mathbf{x}, \xi(\omega)) = \sum_{k=1}^P L_k(\mathbf{x}) \chi_k(\xi), \quad (15)$$

where L_k are deterministic functions derived from (14) and for which closed forms can be obtained algebraically (see References 4,6,24-26) and χ_k are multi-dimensional chaos polynomials in d random variables ξ_1, \dots, ξ_d , of degree less than or equal to p .

5 | STOCHASTIC FINITE ELEMENT APPROXIMATION

The implementation of the spectral stochastic finite element method (SSFEM) for discretizing the weak formulation of the primal problem (3) involves separate discretizations of the deterministic and stochastic spaces. According to the Galerkin method, we define the finite-dimensional subspaces $S^h \subset X_0$ and $T^h \subset L^2(\Omega)$ such that $W^h = S^h \otimes T^h \subset W = X_0 \otimes L^2(\Omega)$. The discrete variational formulation of (3) is: Find $u^h \in W^h$ such that

$$\langle a(u^h, w^h) \rangle = \langle L(w^h) \rangle, \quad \forall w^h \in W^h. \quad (16)$$

The deterministic space X_0 is discretized using the classical finite element basis functions $\phi_i(\mathbf{x})$, $i = 1, \dots, N_u$, where N_u is the number of finite element nodes. These basis functions are piecewise linear on a partition Z^h of D defined by triangular finite elements T_i , $i = 1, \dots, N_e$, such that,

$$Z^h = \bigcup_{i=1}^{N_e} T_i,$$

where N_e denotes the number of finite elements. Here h denotes the discretization parameter and describes the size of the finite elements in Z^h . Let E^h be the collection of numbered edges, e_i , $i = 1, \dots, N_{edge}$, where N_{edge} is the total number of edges in Z^h .

The stochastic space $L^2(\Omega)$ is discretized by means of polynomial chaos of order less than or equal to P in d random variables ξ_i , $i = 1, \dots, d$. The polynomial chaos basis for T^h contains multidimensional polynomials of degree less than or equal to P , $T^h = \text{span}\{\chi_1, \dots, \chi_P\}$ where

$$P = \frac{(d+p)!}{d!p!}, \quad (17)$$

and d represents the number of random variables (number of terms retained in the KLE expansion). The *polynomial chaos* basis is chosen so that the following orthogonality condition is satisfied

$$\langle \chi_i \chi_j \rangle = \langle \chi_i \rangle^2 \delta_{ij}. \quad (18)$$

In this article, the probability measure corresponds to that of a d -dimensional normal distribution. Hence, the basis for T^h consists of d -dimensional Hermite polynomials.

5.1 | Linear system

To obtain the discrete linear system associated with the weak formulation (16), the mean potential u is approximated by

$$u^h(\mathbf{x}, \xi) = \sum_{s=1}^P \sum_{r=1}^{N_u} u_{s,r} \phi_r(\mathbf{x}) \chi_s(\xi) = \sum_{s=1}^P \mathbf{u}_s(\mathbf{x}) \chi_s(\xi). \quad (19)$$

Substituting the expansion (19) for u^h in (16) and choosing test functions $w^h = \phi_k(\mathbf{x})\chi_l(\xi), k = 1, \dots, N_u, l = 1, \dots, P$, yields the linear system of equations

$$A\mathbf{u} = \mathbf{f}, \tag{20}$$

where A is a sparse matrix of size $N_u P \times N_u P$ with a block-structure

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,P} \\ A_{2,1} & A_{2,2} & \dots & A_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P,1} & A_{P,2} & \dots & A_{P,P} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_P \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_P \end{pmatrix}. \tag{21}$$

The diagonal blocks of A are defined as tensor products of the mean stiffness matrix, K_0 , and $\langle \chi_i \rangle^2$, that is,

$$A_{i,i} = \langle \chi_i \rangle^2 \otimes K_0, \quad i = 1, \dots, P, \tag{22}$$

where

$$(K_0)_{r,s} = \int_D \mu(\mathbf{x}) \nabla \phi_r(\mathbf{x}) \cdot \nabla \phi_s(\mathbf{x}) d\mathbf{x}, \tag{23}$$

and μ denotes the mean value of the conductivity field $C(\mathbf{x}, \xi)$. The off-diagonal blocks of A are tensor products of the stiffness matrices, K_l , with the coefficients of the polynomial chaos expansion, $c_{i,j,l} = \langle \xi_l \chi_i \chi_j \rangle, i, j = 1, \dots, P$, and $l = 1, \dots, d$, that is,

$$A_{i,j} = \sum_{l=1}^d [\langle \xi_l \chi_i \chi_j \rangle] \otimes K_l, \tag{24}$$

where

$$(K_l)_{r,s} = \sigma \sqrt{\lambda_l} \int_D \beta_l(\mathbf{x}) \nabla \phi_r(\mathbf{x}) \cdot \nabla \phi_s(\mathbf{x}) d\mathbf{x}. \tag{25}$$

Thus, the coefficient matrix A can be expressed in the tensor product form (see Powell⁹)

$$A = G_0 \otimes K_0 + \sum_{k=1}^d G_k \otimes K_k, \tag{26}$$

where the entries of the stochastic matrices are defined by

$$(G_0)_{r,s} = \langle \chi_r, \chi_s \rangle, \quad (G_k)_{r,s} = \langle \xi_k \chi_r \chi_s \rangle.$$

It is evident that the sparsity of the global stochastic coefficient matrix A is governed by the coefficients of the polynomial chaos expansion, while the sparsity of the blocks of A is determined by the sparsity of the deterministic finite element stiffness matrix.

The maximum degree of the polynomial chaos expansions of u and $\mathcal{L}(\mathbf{x}, \cdot)$ should be different. In fact, it can be shown that for the Galerkin matrix A to be positive definite^{4,16,27} all polynomials of degree less than or equal to $2P$ have to be included in the polynomial chaos expansion of \mathcal{L} . It is only when this condition is satisfied that a full Galerkin projection of the polynomial chaos expansion of \mathcal{L} is obtained. Following Ullmann,⁴ the number of chaos polynomials used for the representation of \mathcal{L} is

$$N = \frac{(d + 2P)!}{d!(2P)!}. \tag{27}$$

It can be demonstrated^{4,27} that the inner product $\langle \chi_k \chi_i \chi_j \rangle$ is non-zero in only finitely many cases. In fact, $\langle \chi_k \chi_i \chi_j \rangle = 0$ for all χ_k with total degree greater than $2P$. A consequence of this observation is that given a fixed number of random

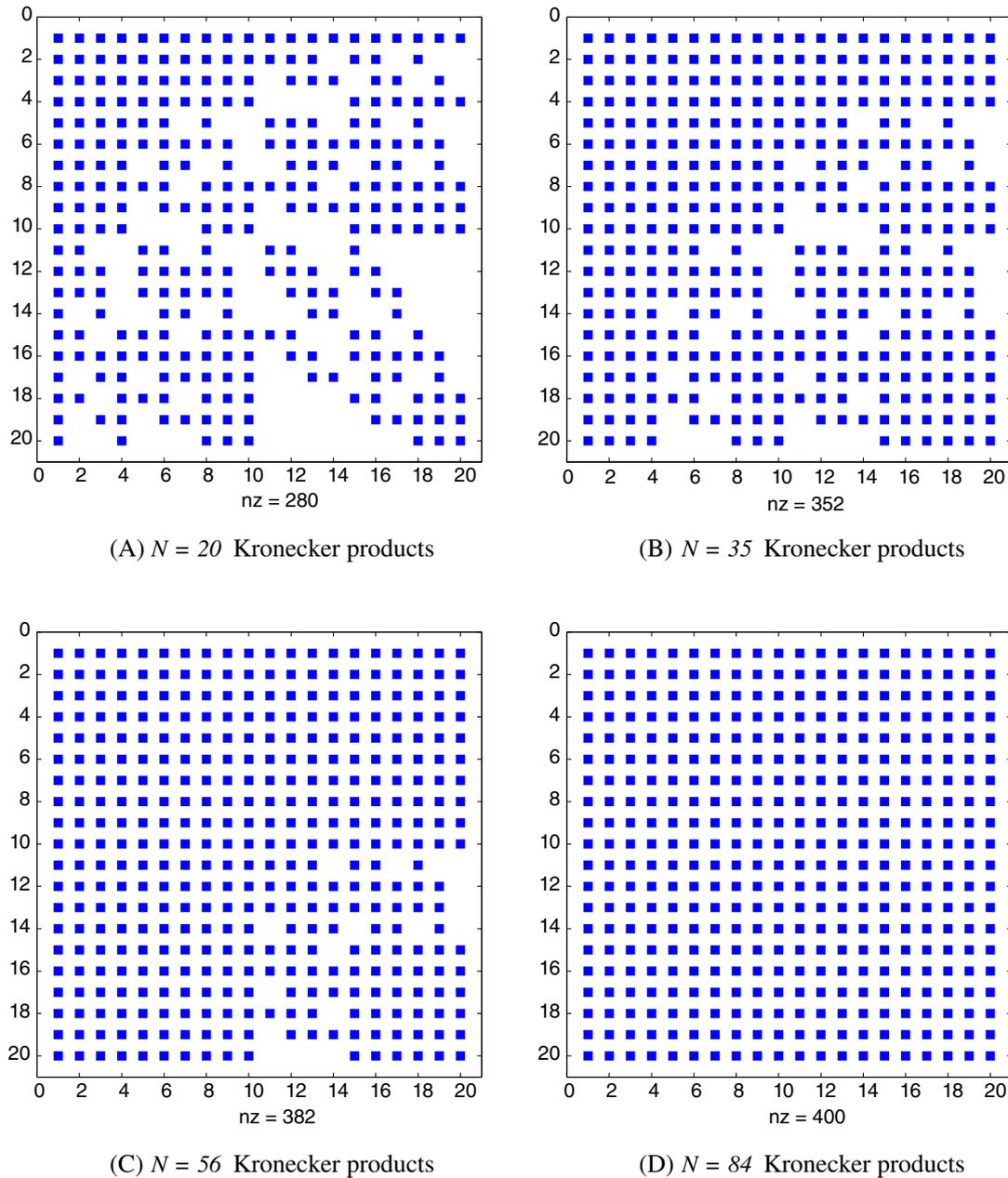


FIGURE 1 Block sparsity pattern of A for different values of N . A, $N = 20$ Kronecker products. B, $N = 35$ Kronecker products. C, $N = 56$ Kronecker products. C, $N = 84$ Kronecker products [Colour figure can be viewed at wileyonlinelibrary.com]

variables d , the infinite polynomial chaos expansion of \mathcal{L} automatically truncates itself as part of the SG method (see Figure 1D). Hence, since the expansion truncates naturally, no error is incurred in the representation of \mathcal{L} .

To make this clearer, let us consider the case in which $d = 3$ and $p_u = 3$. According to (17), the size of the stochastic space for the approximation of u is $P = 20$. Now, the number of Kronecker products N can take the value 20 if the same maximum polynomial order, $p_{\mathcal{L}}$, is used for the expansion of the lognormal conductivity coefficient. Alternatively, maximum polynomial orders of 4, 5, or 6 can be used to give the number of Kronecker products corresponding to 35, 56, or 84, respectively. Although any value of $p_{\mathcal{L}}$ can be used, it is only for $p_{\mathcal{L}} = 6$ ($p_{\mathcal{L}} = 2p_u$), which corresponds to $N = 84$, that a full Galerkin projection of the lognormal random field is obtained. Furthermore, only in this case is the global Galerkin matrix A guaranteed to be positive definite (see remark 2.3.4 of Ullmann⁴).

Figure 1 illustrates the block sparsity pattern of A for different values of N . Note that if polynomials of maximum order $p_{\mathcal{L}}$ are used for the chaos expansion of the permeability tensor, then every block of A is non-zero (see Figure 1D). If

polynomials of order higher than six are considered, the matrices G_k corresponding to orders higher than $2P$ will have only zero entries.

5.2 | Implementation and solution strategies

The dimension of the global coefficient matrix A grows rapidly with p and d , which means that it is never completely assembled due to memory constraints. As originally observed by Ghanem and Kruger,⁷ it is necessary to store $d + 1$ matrices of size $N_u \times N_u$ corresponding to each $K_k, k = 0, \dots, d$, in (26) and the non-zero entries of the stochastic matrices G_k .

The discrete linear system (20) is solved using the preconditioned conjugate gradient (PCG) method. The first preconditioners that were developed for the iterative solution of this system were based on incomplete factorizations of the diagonal blocks of A .^{7,8} We define the block-diagonal preconditioner \mathcal{P}_{bdiag} and the mean preconditioner \mathcal{P}_{mean} by

$$\mathcal{P}_{bdiag} = G_0 \otimes K_0, \quad \mathcal{P}_{mean} = I \otimes K_0, \tag{28}$$

respectively. Each PCG iteration involves the solution of a system of the form $\mathcal{P}\mathbf{z} = \mathbf{r}$, where \mathbf{r} is the residual vector. This requires the solution of P subsystems of equations each of size $N_u \times N_u$ with coefficient matrix K_0 .

Powell and Elman⁹ claimed that any efficient deterministic solver can be used for the solution of the P subsystems and proposed the use of one V-cycle of black-box AMG. The crucial advantage of using black-box AMG is that the computational cost of one V-cycle is linearly proportional to the discretization parameter h .

It is observed that, when Gaussian random variables are employed, the preconditioned system is positive definite only when the variance and the order of the polynomials are small.⁹ This is a consequence of the infinite support of the Gaussian distribution and the violation of condition (5) for the permeability tensor. Preconditioned CG breaks down when this condition is violated. Therefore, the use of Hermite polynomials is limited to problems with small variances.

For the SFEM method to be computationally efficient and competitive with respect to traditional sampling methods, the CG method needs to be equipped with robust preconditioners that are optimal with respect to h, d, p and especially C . It is well known that the performance of preconditioners deteriorates for problems in which C has a large standard deviation. This is due to the fact that the off-diagonal blocks of A , which are usually ignored when constructing these preconditioners, become increasingly significant in this case.

To overcome this deficiency, a new preconditioner that fully exploits the block structure of A is proposed. This preconditioner ensures that important information contained in the off-diagonal blocks is captured when C has a large standard deviation. This is achieved through the addition of an inner iteration to the preconditioning operation which essentially implements a full inversion of the global stiffness matrix A using a bSGS algorithm. The preconditioner, which to our knowledge has not been used in the SFEM context before, takes the form,

$$\mathcal{P}_{bSGS} = G_0 \otimes K_0 + \sum_{k=1}^d G_k \otimes K_k. \tag{29}$$

Note, however, that the preconditioner \mathcal{P}_{bSGS} is neither assembled nor inverted directly. Each inner iteration requires the solution of P subproblems of size $N_u \times N_u$ using an appropriate fast solver such as *UMFPACK* or one V-cycle of *AMG* for the deterministic problem under consideration. The symmetry of the preconditioner is guaranteed by performing two iterative sweeps (forward and backward).

Two stopping criteria are used for the proposed algorithm. The inner iteration is terminated when

$$\left\| \mathbf{z}^k - \mathbf{z}^{k-1} \right\|_{\infty} < \epsilon,$$

where $\epsilon = 10^{-8}$ unless a maximum number of iterations *maxitb* is achieved. At this stage, the current approximation for \mathbf{z} is chosen to be the preconditioned residual needed within the current CG iteration. In general, the CG algorithm should decrease the number of CG iterations and, in particular, it should improve the iteration count for those problems for which the off-diagonal blocks of A are as significant as the diagonal blocks, that is, problems in which the spatial random field has a large standard deviation.

6 | STOCHASTIC MIXED FINITE ELEMENT APPROXIMATION (SMFEM)

The approach to SMFEM for solving the mixed variational problem (6) is similar to the one presented in the previous section. However, the mixed finite element approximation requires the definition of subspaces for $H(\text{div}; D)$ in addition to $L^2(D)$. We consider the Raviart-Thomas space of lowest order RT_0 as a suitable space for the approximation of the velocity solution and $M_0(K)$ is defined to be the space of piecewise constant functions.

The stochastic space $L^2(\Omega)$ is discretized by means of polynomial chaos. The spaces for the stochastic approximation are consequently given by $V^h = Y^h \otimes T^h \subset V = H(\text{div}; D) \otimes L^2(\Omega)$ and $W^h = X^h \otimes T^h \subset W = L^2(D) \otimes L^2(\Omega)$.

The discrete variational formulation of (9) is: find $\mathbf{q}^h \in V^h$ and $u^h \in W^h$ such that

$$\begin{aligned} \langle a(\mathbf{q}^h, \mathbf{v}^h) \rangle + \langle b(\mathbf{v}^h, u^h) \rangle &= \langle (\mathbf{g}, \mathbf{n} \cdot \mathbf{v})_{\Gamma_D} \rangle, & \forall \mathbf{v}^h \in V^h \\ \langle b(\mathbf{q}^h, w^h) \rangle &= - \langle (f, w^h) \rangle. & \forall w^h \in W^h. \end{aligned} \quad (30)$$

6.1 | Linear system

The potential u^h and flux (or velocity) \mathbf{q}^h are expressed in terms of the expansions

$$u^h(\mathbf{x}, \xi) = \sum_{s=1}^P \sum_{r=1}^{N_e} u_{s,r} \phi_r(\mathbf{x}) \chi_s(\xi), \quad \mathbf{q}^h(\mathbf{x}, \xi) = \sum_{s=1}^P \sum_{r=1}^{N_{\text{edge}}} \mathbf{q}_{s,r} \psi_r(\mathbf{x}) \chi_s(\xi). \quad (31)$$

Substituting for u^h and \mathbf{q}^h using expansions (31) into (30), we obtain the discrete linear system

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{f} \end{pmatrix}, \quad (32)$$

where A is a sparse symmetric matrix of size $N_{\text{edge}}P \times N_{\text{edge}}P$ and B is a sparse block diagonal matrix of size $N_eP \times N_{\text{edge}}P$.

The structure of C where

$$C = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}, \quad (33)$$

is governed by the coefficients of the polynomial chaos expansion, while the sparsity of each of the blocks of C corresponds to the sparsity of the deterministic velocity matrix and the deterministic divergence operator.

The diagonal blocks of A are given by (22) with

$$(K_0)_{r,s} = \int_D \frac{1}{\mu} \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}, \quad (34)$$

while the off-diagonal blocks are given by (24) with

$$(K_l)_{r,s} = \sigma \sqrt{\lambda_l} \int_D \beta_l(\mathbf{x}) \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}. \quad (35)$$

The diagonal blocks of B are defined by

$$B_{i,i} = \langle \chi_i \rangle^2 \otimes B_0, \quad (36)$$

where

$$(B_0)_{r,s} = \int_D \phi_r(\mathbf{x}) \nabla \cdot \psi_s(\mathbf{x}) d\mathbf{x}. \quad (37)$$

When the permeability tensor is approximated in terms of a polynomial chaos basis of order p , A reduces to size $N_{edge}p \times N_{edge}p$ and is tridiagonal and B reduces to size $N_e p \times N_{edge}p$. The diagonal blocks of A are given by (22), while the off-diagonal blocks have the form

$$A_{i,j} = \langle k_k \chi_i^k \chi_j^k \rangle \otimes K, \tag{38}$$

where

$$K_{r,s} = \int_D \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}. \tag{39}$$

6.2 | Implementation and solution strategies

As for the SFEM method, the coefficient matrix C is never assembled. In addition to storing $d + 1$ matrices of size $N_{edge} \times N_{edge}$ and the entries of the stochastic matrices G_k , we also store the matrix B_0 of size $N_e \times N_{edge}$.

The efficient solution of the saddle-point system (32) is an active field of research.^{10,28,29} The approach adopted here follows the experience gained from solving the deterministic system. The system is solved using MINRES with a preconditioner based on the approximation of the Schur complement by sparse direct or AMG methods.

For the stochastic system, we use a preconditioner that follows from its deterministic version given by

$$\mathcal{P} = \begin{pmatrix} \tilde{N} & 0 \\ 0 & \tilde{V} \end{pmatrix}, \tag{40}$$

where

$$\tilde{N} = \text{diag}(A) = G_0 \otimes \text{diag}(K_0), \tag{41}$$

and

$$\tilde{V} = B \tilde{N}^{-1} B^T = G_0 \otimes [B_0 \text{diag}(K_0)^{-1} B_0^T]. \tag{42}$$

Following the discussion for the second-order problem in Section 5.2, the preconditioner (40) is expected to be efficient only for problems in which the spatial random field is characterized by small or moderate standard deviation. A robust preconditioner with respect to σ , for the mixed stochastic formulation remains an active area of research.

7 | COMPARISON OF SMFEM AND MCM

In this section, the SMFEM method is compared with MCM when lognormal distributions are used to describe the permeability tensor. The aim is to validate SMFEM against the established MCM approach. The simulations have all been carried out in serial using MATLAB 7.4 on a laptop PC with 4GB of RAM.

Consider test problem 2 defined in Part 1. This problem, defined in the computational domain $[0, 1]^2$, is similar to one considered by Powell and Elman.⁹ Dirichlet conditions $u(0, y) = 1$ and $u(1, y) = 0$ are imposed on the vertical boundaries and homogeneous Neumann conditions $C \nabla u \cdot \mathbf{n} = 0$ are imposed on the horizontal boundaries. Thus, the dominant flow direction is from left to right. The first order problem is solved in this section corresponding to the system of equations described in (6).

The spatial discretization uses a triangular mesh with $h = 1/64$, thus the number of unknowns given by the mixed formulation is the sum of the number of elements, $N_e = 8192$, and number of edges, $N_{edg} = 12\,416$. The stochastic space is discretized in a similar fashion to the one described for the SFEM case, that is, polynomial chaos up to order $p_u = 4$ are used for both u and \mathbf{q} .

Solution profiles for various values of p_u and N_r are shown in Figure 2. For the mean velocity (x -component) solution, the profile presented is along $y = 0.5$ and for the variance solution is along $x = 0.5$. An in-depth convergence study for a sampling point having coordinate $(0.5, 0.5)$ is reported in Table 1.

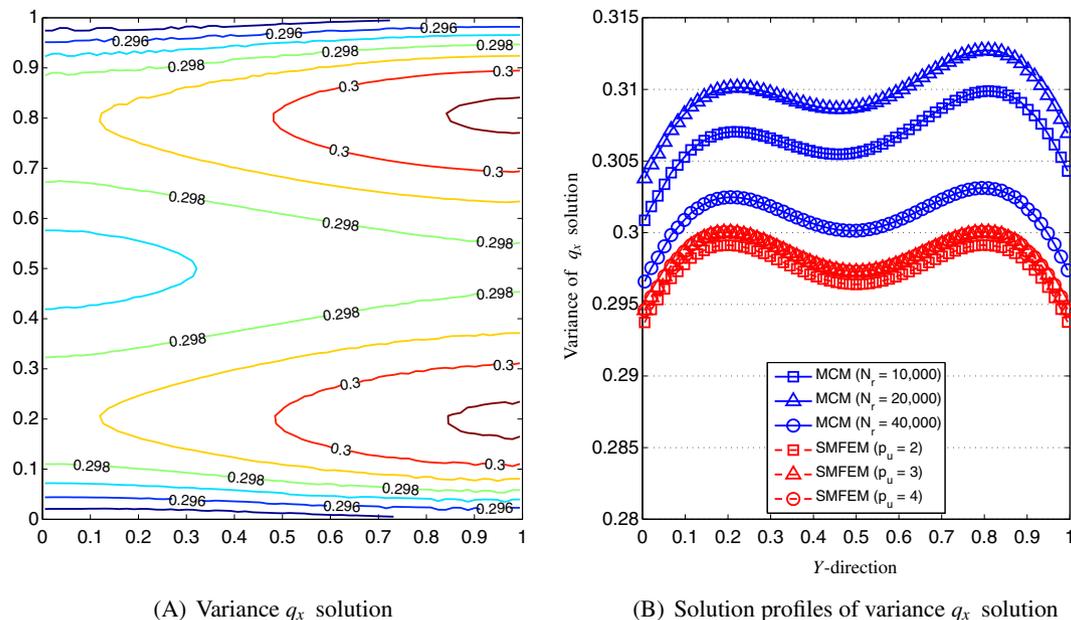


FIGURE 2 Variance solutions for test problem with lognormal distribution (SMFEM). A, Variance q_x solution. B, Solution profiles of variance q_x solution [Colour figure can be viewed at wileyonlinelibrary.com]

		$N_r = 10\,000$	$N_r = 20\,000$	$N_r = 40\,000$
\mathbf{q}_x	Sample mean	2.77182	2.77526	2.77055
	Sample variance	0.30558	0.30873	0.30014
	$t_{CPU}(sec)$	6261	12 524	25 048
		$p_u = 2$	$p_u = 3$	$p_u = 4$
\mathbf{q}_x	Mean	2.76780	2.76782	2.76782
	Variance	0.29639	0.29726	0.29728
	$t_{CPU}(sec)$	44	388	2312

TABLE 1 Convergence analysis of MCM and SMFEM for test problem with lognormal distribution

Abbreviations: MCM, Monte Carlo method; SMFEM, stochastic mixed finite element method.

Polynomials of order two are sufficient to achieve convergence to the fourth significant digit for the mean whereas polynomials of order three are required to achieve the same level of accuracy for the variance. This is in agreement with the convergence rate of the mean and variance solution for the potential recorded for the second order problem.

It is apparent from the data in Table 1 that the Monte Carlo mean for the x -component of the velocity field does not converge for the sample sizes considered. This suggests that a larger sample is required to achieve the desired level of accuracy. Equally, the variance does not converge for the maximum sample size herein considered.

Table 1 also includes solution timings for MCM and SMFEM. Although the data show that SMFEM is more efficient than MCM, this conclusion cannot be generalized. In fact, the performance of preconditioned MINRES deteriorates significantly for conductivity fields with large standard deviations when lognormal distributions are used. Evidence of this is reported and discussed in Section 9.1.

8 | ITERATIVE SOLVERS FOR SFEM

The algorithms used for the numerical experiments are structured so that the coefficient matrix is never fully assembled. Only the non-zero entries of the polynomial chaos coefficients (appropriately indexed) and N matrices (associated with the polynomial chaos discretization of the spatial random field) are stored. However, in order to guarantee the

TABLE 2 Dimensions of P , N , and total number of unknowns

	$p_u = 2$	$p_u = 3$	$p_u = 4$
$d = 4$			
P	15	35	70
N	70	210	495
$h = \frac{1}{32}$	16 335	38 115	76 230
$h = \frac{1}{64}$	63 375	147 875	295 750
$h = \frac{1}{128}$	249 615	582 435	1 164 870
$d = 6$			
P	28	84	210
N	210	924	3,003
$h = \frac{1}{32}$	30 492	91 476	228 690
$h = \frac{1}{64}$	118 300	354 900	887 250
$h = \frac{1}{128}$	465 948	1 397 844	3 494 610

well-posedness of A , N has to be very large. Hence, the memory requirements of SG for the stochastically nonlinear case are significantly larger than the corresponding linear case.

In Part 1, it was shown that the version of the mean-based preconditioners, \mathcal{P}_{bdiag} and \mathcal{P}_{mean} , based on an incomplete Choleski factorization of K_0 was significantly less efficient than the *AMG* and *UMFPACK* versions. Hence, only the latter two versions are considered. A symmetric Gauss-Seidel smoother is used for the AMG implementation. The simulations have been performed in serial within the MATLAB environment installed on the SRIF-3 Cluster machine (Merlin) at Cardiff University. Thus, the CPU timings reported in the following sections can be compared directly with those reported in Part 1.

8.1 | Block-diagonal preconditioner

8.1.1 | Test problem 1—variable σ

This test problem is described in Section 7. The conductivity coefficient \mathcal{L} is a lognormal field, the spatial variability of which is described in Section 7. The underlying Gaussian distribution has constant mean $\mu = 1$ and four different values of the standard deviation are considered. The discretization parameter is fixed at $h = 1/32$.

The size of the stochastic space and the total number of Kronecker products are as those reported in Table 2 and the total number of unknowns corresponds to those for $h = 1/32$ in Table 2.

The number of conjugate gradient iterations, preconditioned by \mathcal{P}_{bdiag} , N_{it} , and t_{CPU} are reported in Table 3. Set-up times depend only on the size of the problem (which in this case corresponds to $h = 1/32$), and therefore, they are approximately equal for all values of σ .

The results presented in Table 3 can be summarized as follows:

1. The performance of the block-diagonal preconditioner deteriorates significantly for large values of standard deviation;
2. N_{it} and t_{CPU} show exponential growth with respect to p_u , for both values of d ;
3. There is a slight deterioration in d -optimality for $p_u = 4$ for both versions of the preconditioner;

8.1.2 | Test problem 2—discontinuous-isotropic conductivity field

In this test problem, the domain $D = [0, 1]^2$ is partitioned into four equal subdomains. Dirichlet conditions are imposed on the vertical sides $\Gamma_D = \{0, 1\} \times [0, 1]$. Homogeneous Neumann boundary conditions are imposed on the horizontal boundaries. The conductivity coefficient \mathcal{L} is a spatially discontinuous lognormal random field. Four cases are presented, three of which have constant coefficient of variation δ and one with spatially variable δ . The underlying Gaussian distributions (one for each of the four sub-domains) are characterized by the following parameters:

TABLE 3 CG iterations and solution timings for \mathcal{P}_{bdiag} —Test Problem 1

	σ	$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
<i>d</i> = 4							
<i>UMFPACK</i>	0.3	11	0.7	14	4.73	17	28.66
	0.5	16	0.96	23	7.83	30	50.05
	0.7	23	1.38	34	11.52	48	80.09
	0.9	29	1.74	51	17.38	80	134.01
<i>AMG</i>	0.3	13	1.32	16	5.76	19	32.56
	0.5	18	1.23	25	8.98	32	54.67
	0.7	24	1.69	37	13.3	52	88.93
	0.9	31	2.1	54	19.33	83	142.59
<i>d</i> = 6							
<i>UMFPACK</i>	0.3	11	1.97	14	25.48	17	239.24
	0.5	17	3	24	43.65	30	419.48
	0.7	23	4.05	35	63.48	50	697.9
	0.9	30	5.27	52	94.26	83	1158.83
<i>AMG</i>	0.3	13	2.51	16	29.93	19	268.52
	0.5	18	3.5	25	46.61	33	468.54
	0.7	24	4.6	38	70.83	52	735.08
	0.9	31	5.92	55	102.78	87	1239.99

1. Case 1:

$$\mu_{D_1} = 1.0, \sigma_{D_1} = 0.5, \mu_{D_2} = 0.1, \sigma_{D_2} = 0.05,$$

$$\mu_{D_3} = 0.01, \sigma_{D_3} = 0.005, \mu_{D_4} = 0.0001, \sigma_{D_4} = 0.00005;$$

2. Case 2:

$$\mu_{D_1} = 1.0, \sigma_{D_1} = 0.7, \mu_{D_2} = 0.1, \sigma_{D_2} = 0.07,$$

$$\mu_{D_3} = 0.01, \sigma_{D_3} = 0.007, \mu_{D_4} = 0.0001, \sigma_{D_4} = 0.00007;$$

3. Case 3:

$$\mu_{D_1} = 1.0, \sigma_{D_1} = 1.0, \mu_{D_2} = 0.1, \sigma_{D_2} = 0.1,$$

$$\mu_{D_3} = 0.01, \sigma_{D_3} = 0.01, \mu_{D_4} = 0.0001, \sigma_{D_4} = 0.0001;$$

4. Case 4:

$$\mu_{D_1} = 1.0, \sigma_{D_1} = 1.0, \mu_{D_2} = 0.1, \sigma_{D_2} = 0.07,$$

$$\mu_{D_3} = 0.01, \sigma_{D_3} = 0.005, \mu_{D_4} = 0.0001, \sigma_{D_4} = 0.0001.$$

A KLE is performed for each subdomain and the number d of terms retained in the expansion is the same for each subdomain. The case in which different values of d are used in each subdomain has not been considered and is a subject for further study. The same spatial model with $l_x = l_y = 0.5$ is used for each subdomain. The discretization parameter is

TABLE 4 CG iterations and solution timings for \mathcal{P}_{bdiag} —Test Problem 2

		$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
$\delta = \frac{\sigma}{\mu}$							
<i>d = 4</i>							
<i>UMFPACK</i>	0.5	16	1.02	22	7.47	27	45.3
	0.7	22	1.33	32	10.92	43	72.27
	1.0	33	1.99	56	19.21	84	141.2
	1.0,0.7,0.5,1.0	33	1.99	55	18.73	84	140.98
<i>AMG</i>	0.5	17	1.21	23	8.28	28	48.07
	0.7	23	1.57	34	12.19	45	77.22
	1.0	34	2.32	57	20.43	86	147.55
	1.0,0.7,0.5,1.0	34	2.31	57	20.5	86	147.41
<i>d = 6</i>							
<i>UMFPACK</i>	0.5	17	3	23	41.73	28	392.59
	0.7	23	4.08	34	61.76	45	633.19
	1.0	35	6.19	58	105.52	90	1265.67
	1.0,0.7,0.5,1.0	35	6.16	58	105.88	90	1261.79
<i>AMG</i>	0.5	18	3.48	24	44.69	29	420.91
	0.7	24	4.61	35	65.76	47	670.45
	1.0	36	6.9	61	113.85	93	1313.96
	1.0,0.7,0.5,1.0	36	6.88	60	111.57	93	1320.7

fixed, $h = 1/32$, and the size of the problem is given in Table 2. Iteration count and timings for CG preconditioned with the block-diagonal of A are given in Table 4. Also, for this test problem, the set-up times are approximately equal for all values of δ .

Similar observations are drawn for this test problem as the ones highlighted in Section 8.1.1. The exponential growth of N_{it} and t_{CPU} with increasing p is also observed for this test problem. In addition, it appears that the deterioration in the performance of the preconditioner is exclusively due to the increase in the value of σ in each subdomain from case to case. In fact, in all four cases, the mean values μ_{D_1, D_2, D_3, D_4} are equal. This indicates that the preconditioned solver is robust with respect to discontinuities in the mean value of the permeability.

8.2 | Block symmetric Gauss-Seidel preconditioner

In this section, the same test problems presented in Section 8.1 are solved using a bSGS preconditioner for CG. The algorithm used in the experiments is described in Section 5.2. A fixed number of iterations, $maxitb$, is used as stopping criteria for \mathcal{P}_{bSGS} , each iteration comprising a forward and backward sweep. The experiments reported in the following sections are performed using $maxitb = 1$. The reason for this choice together with an in-depth analysis on the performance of \mathcal{P}_{bSGS} for several values of $maxitb$ is given in Section 8.2.3.

The UMFPACK implementation of the bSGS preconditioner is straightforward and it is identical to the one used in Part 1 for the linear case. In contrast, the AMG implementation requires additional pre-processing to be implemented. In fact, in contrast to the linear case, the tensor products $G_k \otimes K_k, k = 1, \dots, N$, have several contributions to the blocks of the leading diagonal of the coefficient matrix, depending on the value of d and p_u . So, for example, fixing $d = 2$ and $p_u = 3$, the contributions are shown in Table 5.

Note that the matrix G_1 is diagonal in that the product $G_1 \times K_1$ contains the mean information (this is omitted from Table 5). In the nonlinear case the AMG grids have to be computed for each diagonal block entries of the global system. Thus, the AMG pre-processing is implemented P times and the grids are stored before the iterative

(i,j)	$G_k \times K_k$
(1, 1)	
(2, 2)	$G_4(2, 2) \times K_4$
(3, 3)	$G_6 \times K_6$
(4, 4)	$(G_4 \times K_4) + (G_{11} \times K_{11})$
(5, 5)	$(G_4 \times K_4) + (G_6 \times K_6) + (G_{13} \times K_{13})$
(6, 6)	$(G_6 \times K_6) + (G_{15} \times K_{15})$
(7, 7)	$(G_4 \times K_4) + (G_{11} \times K_{11}) + (G_{22} \times K_{22})$
(8, 8)	$(G_4 \times K_4) + (G_6 \times K_6) + (G_{11} \times K_{11}) + (G_{13} \times K_{13}) + (G_{24} \times K_{24})$
(9, 9)	$(G_4 \times K_4) + (G_6 \times K_6) + (G_{13} \times K_{13}) + (G_{15} \times K_{15})$
(10, 10)	$(G_6 \times K_6) + (G_{15} \times K_{15}) + (G_{26} \times K_{26}) + (G_{28} \times K_{28})$

TABLE 5 $G_k \times K_k$ contributions to the blocks of the diagonal of A

	σ	$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
<i>d</i> = 4							
<i>UMFPACK</i>	0.3	4	0.87	4	2.44	5	14.95
	0.5	6	0.65	7	4.28	8	23.9
	0.7	8	0.86	10	6.11	12	35.8
	0.9	10	1.07	14	8.53	18	53.73
<i>AMG</i>	0.3	6	1.11	7	4.35	7	20.82
	0.5	8	0.94	9	5.6	11	32.63
	0.7	10	1.21	14	8.71	20	59.53
	0.9	14	1.64	23	14.27	46	136.93
<i>d</i> = 6							
<i>UMFPACK</i>	0.3	4	1.27	4	13.11	5	123.46
	0.5	6	1.91	7	22.9	8	198.15
	0.7	8	2.55	10	32.62	12	297.21
	0.9	11	3.51	14	45.85	19	473.32
<i>AMG</i>	0.3	6	2	7	22.94	7	171.93
	0.5	8	2.68	10	32.71	12	297.72
	0.7	11	3.66	15	49.01	22	546.12
	0.9	15	5	25	81.93	52	1279.58

TABLE 6 CG iterations and solution timings for \mathcal{P}_{bSGS} —Test Problem 1

solution process begins. Clearly, the preconditioner set-up time contributes significantly to the overall CPU cost of the solver.

8.2.1 | Test problem 1—variable σ

Table 6 reports the number of CG iterations and CPU times for test problem 2.

The results presented in Table 6 can be summarized as follows:

TABLE 7 CG iterations and solution timings for \mathcal{P}_{bSGS} —Test Problem 2

	$\delta = \frac{\sigma}{\mu}$	$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
<i>d = 4</i>							
<i>UMFPACK</i>	0.5	6	0.69	7	4.27	7	20.86
	0.7	8	0.86	9	5.54	11	33.04
	1.0	11	1.18	15	9.23	19	56.64
	1.0,0.7,0.5,1.0	11	1.19	15	9.22	19	56.73
<i>AMG</i>	0.5	7	0.91	8	4.97	10	29.7
	0.7	9	1.06	12	7.46	16	47.59
	1.0	14	1.67	23	14.43	48	143.13
	1.0,0.7,0.5,1.0	14	1.67	23	14.42	48	143.13
<i>d = 6</i>							
<i>UMFPACK</i>	0.5	6	1.91	7	22.82	8	198.34
	0.7	8	2.55	9	29.49	11	273.95
	1.0	11	3.5	15	49.18	21	520.41
	1.0,0.7,0.5,1.0	11	3.51	15	49.08	21	523.93
<i>AMG</i>	0.5	8	2.68	9	29.45	10	245.42
	0.7	10	3.38	13	42.49	18	440.4
	1.0	16	5.41	27	89.2	55	1348.74
	1.0,0.7,0.5,1.0	16	5.39	27	88.5	55	1345.02

1. The block symmetric Gauss-Seidel preconditioner displays a significant improvement in terms of number of CG iterations. This improvement becomes more evident for large values of σ ;
2. The comparison of the data with those of Table 3 (block-diagonal preconditioner) reveals that the Gauss-Seidel preconditioner is generally computationally cheaper and the improvement in performance increases with larger values of σ ;
3. The difference in performance between the exact and approximate versions of the preconditioner increases for larger σ . In fact, for $\sigma = 0.9$, the AMG solution times are approximately three times larger than for UMFPACK.

8.2.2 | Test problem 2—discontinuous-isotropic conductivity field.

Table 7 reports the number of CG iterations count and CPU times for test problem 3 using a bSGS preconditioner. As for test problem 2, a significant improvement in both N_{it} and t_{CPU} is achieved. A large saving in computational cost was recorded for higher polynomial orders and large δ . In fact, the CPU time is reduced by 60% compared with results obtained using the \mathcal{P}_{bdiag} preconditioner. Significant time reduction is equally achieved for lower polynomial orders and coefficient of variation δ . This is around 37% for $p_u = 2$ and $\delta = 0.5$, and around 53% for $p_u = 3$ and $\delta = 0.7$.

As observed previously for the preconditioner \mathcal{P}_{bdiag} , it appears that a discontinuous conductivity coefficient (jumps in the mean conductivity value at the subdomain boundaries) does not worsen the preconditioner performance. It is, in fact, the standard deviation that has a significant negative impact on the performance of both \mathcal{P}_{bdiag} and \mathcal{P}_{bSGS} . Not even using the \mathcal{P}_{bSGS} algorithm and therefore including the off-diagonal blocks of A (which retain information on the fluctuations about the mean) can optimality of N_{it} with respect to σ be achieved.

$d = 4$	$maxitb$	$\sigma = 0.3$		$\sigma = 0.5$		$\sigma = 0.7$		$\sigma = 0.9$	
		N_{it}	t_{CPU}	N_{it}	t_{CPU}	N_{it}	t_{CPU}	N_{it}	t_{CPU}
	1	5	14.95	8	23.90	12	35.79	18	53.72
	2	3	13.26	5	22.11	8	35.42	13	57.57
	3	2	11.79	4	23.50	7	41.14	10	58.79
	4	2	14.48	4	29.00	6	43.47	9	65.13
	5	2	17.39	3	26.10	5	43.45	8	69.51
	6	1	10.07	3	30.13	4	40.38	7	70.65
	8	1	12.99	2	25.87	4	51.83	6	77.51
	14	1	21.54	1	21.54	3	64.56	4	86.06
	32	1	47.26	1	47.14	1	47.17	3	141.50
	75	1	108.45	1	108.56	1	108.29	1	108.17
$d = 6$									
	1	5	123.46	8	198.15	12	297.21	19	473.32
	2	3	108.02	5	180.14	9	322.50	14	505.53
	3	2	93.71	4	188.48	7	329.93	11	518.46
	4	2	115.72	4	233.33	6	348.50	10	580.29
	5	2	138.72	3	208.24	5	347.25	8	557.42
	6	1	80.55	3	244.95	5	400.84	8	645.87
	8	1	102.93	2	205.71	4	410.97	7	719.04
	14	1	169.66	1	170.22	3	509.89	5	849.04
	34	1	393.42	1	394.49	1	392.42	3	1173.16
	80	1	899.37	1	906.18	1	905.44	1	905.81

TABLE 8 CG iterations and solution timings (sec) for \mathcal{P}_{bSGS} for various values of $maxitb$ —Test Problem 1

8.2.3 | Performance analysis

The numerical experiments presented show that CG equipped with a bSGS preconditioner is significantly more efficient than traditional mean-based preconditioners. This conclusion depends on the stopping criterion chosen for the inner preconditioned Gauss-Seidel algorithm. The results reported in the tables are obtained using a single iteration of the bSGS algorithm.

The choice of $maxitb$ can be optimized. Consider test problem 2 with fixed $p_u = 4$. Simulations are performed increasing the value of $maxitb$ until only one CG iteration is required for convergence. CG iteration count and timings for these experiments with $d = 4$ and $d = 6$ are reported in Table 8. Note that for this analysis, the UMFPACK version of the preconditioner was used.

The results reported in Table 8 suggest that the best solution times are not given by the same stopping criterion for all values of σ . In fact, it appears that for large standard deviations $\sigma = 0.7, 0.9$, the best solution timings are obtained for small $maxitb$. However, for $\sigma = 0.3, 0.5$, the best performance in terms of CPU time is obtained using moderate values of $maxitb$.

The case in which $maxitb$ is large and t_{CPU} is low corresponds to the situation in which convergence is obtained in one CG iteration. It is clear that, under these circumstances, the bulk of the computational work is performed by the preconditioner (bSGS) and very little by the main solver (CG). Given that the preconditioner should only serve as a means to improve the conditioning of the system matrix, the results showing just one CG iteration are not considered in the following analysis. On the other hand, this aspect reveals that an independent Gauss-Seidel (symmetric or not) solver could be a very efficient alternative to Krylov subspace iterative schemes. In Section 8.3, results obtained using Gauss-Seidel solvers are reported for all test problems considered in this paper.

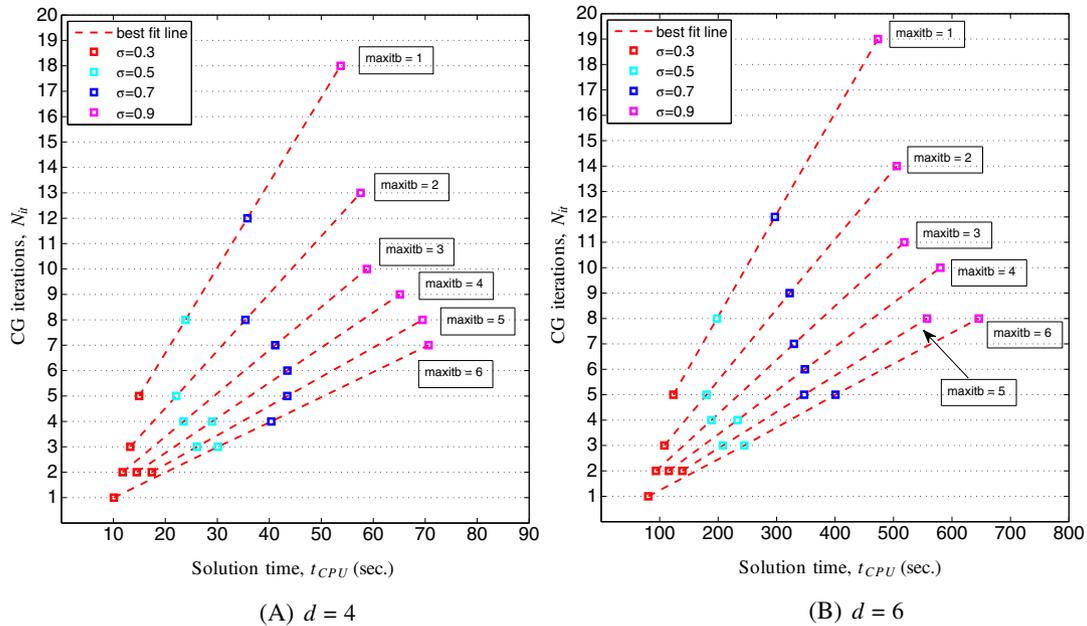


FIGURE 3 Performance analysis of CG preconditioned with \mathcal{P}_{bSGS} for Test Problem 1. A, $d = 4$. B, $d = 6$ [Colour figure can be viewed at wileyonlinelibrary.com]

Excluding the data associated with one CG iteration, Table 8 shows that, in general, a small number of internal iterations for the \mathcal{P}_{bSGS} preconditioner is sufficient to achieve the best performance for all values of standard deviation considered for this test problem. However, it is only for $\sigma = 0.9$ ($d = 4, 6$) and $\sigma = 0.7$ ($d = 6$), that the best performance is achieved using $maxitb = 1$. For $\sigma = 0.5$ ($d = 4, 6$) and $\sigma = 0.7$ ($d = 4$), the best performance is given by $maxitb = 2$, and for $\sigma = 0.3$ ($d = 4, 6$), for $maxitb = 3$.

Figure 3A and B shows the number of CG iterations versus CPU times for $maxitb = 1, 2, 3, 4, 5, 6$ for $d = 4$ and $d = 6$, respectively. The figures highlight that there is a clear linear relationship between the number of CG (preconditioned with \mathcal{P}_{bSGS}) iterations, computational time, and the standard deviation of the spatial random field for all values of $maxitb$. This figure clearly shows that the best convergence rate is given by $maxitb = 1$ and, for this reason, it was chosen as the optimal stopping criterion for the \mathcal{P}_{bSGS} preconditioner.

8.3 | Gauss-Seidel solvers

The performance analysis carried out on test problem 2 in the previous section revealed that for small standard deviation ($\sigma = 0.3$ and $\sigma = 0.5$), the Gauss-Seidel algorithm used as a stand alone solver could be a valid alternative to Krylov subspace solvers for the solution of SFEM systems with lognormal permeability. The same observation was made for the linear case in Part 1. Here results using the bSGS solver ($bSGS$) are presented. The circumstances under which Gauss-Seidel is more efficient than Krylov subspace solvers are explained.

The symmetric Gauss-Seidel solver includes a forward and a backward sweep per iteration and the algorithm is essentially the one used for \mathcal{P}_{bSGS} . The nonsymmetric case only includes a forward sweep per iteration. In both cases, the stopping criteria are determined by the error norm, satisfying a specific tolerance.

Reorderings of the block structure of A aimed at reducing the bandwidth of the coefficient matrix are irrelevant for the lognormal case given that A is block dense. In our implementation, the structure as presented in Figure 1 is retained using the summation of progressive ($i = 1, \dots, N$) Kronecker terms. This ordering is the most natural as it represents the summation of decreasing modes obtained from the polynomial chaos expansion of the permeability (see Equation (14)).

The tolerance for the GS solvers is set to 10^{-8} . In each table, we list iteration count N_{it} and solution times t_{CPU} for both $bSGS$ and bGS . Only experiments using *UMFPACK* to invert the diagonal blocks of A are reported.

	σ	$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
$d = 4$							
<i>bSGS</i>	0.3	8	0.58	9	2.91	9	12.6
	0.5	13	0.92	16	5.2	19	26.72
	0.7	21	1.47	30	9.68	41	57.33
	0.9	33	2.31	54	17.45	86	120.64
<i>bGS</i>	0.3	9	0.32	11	1.77	12	8.38
	0.5	15	0.53	20	3.23	25	17.54
	0.7	23	0.82	36	5.83	53	37.15
	0.9	35	1.25	65	10.42	110	77.24
$d = 6$							
<i>bSGS</i>	0.3	8	1.48	9	14.06	9	99.5
	0.5	13	2.4	17	26.6	20	220.36
	0.7	22	4.06	31	48.47	43	470.07
	0.9	35	6.45	59	92.13	96	1057.31
<i>bGS</i>	0.3	9	0.84	11	8.61	12	66.24
	0.5	15	1.39	20	15.66	25	138.46
	0.7	23	2.12	36	28.1	53	291.58
	0.9	35	3.23	65	50.82	111	610.59

TABLE 9 bSGS and bGS iterations and solution timings—Test Problem 2

8.3.1 | Test Problem 1—variable σ

Table 9 lists iteration counts and timings for test problem 1. The findings of this table are summarized as follows:

1. GS solvers are not optimal with respect to σ ;
2. *bSGS* is computationally more efficient than *CG* preconditioned with \mathcal{P}_{bSGS} only for small standard deviations;
3. Nonsymmetric Gauss-Seidel solver (*bGS*) is very efficient for small and moderate standard deviations. However, for large values of σ it is outperformed by *CG* preconditioned with \mathcal{P}_{bSGS} ;
4. As for the previous case, the *bGS* solver is consistently more efficient than the symmetric implementation.

8.3.2 | Test problem 2—discontinuous-isotropic conductivity field

Table 10 lists iteration count and timings for test problem 2. Similar observations to the ones highlighted for test problem 2 are found from the data presented in this table. Furthermore, the results show that a discontinuous permeability has no negative impact on the performance of Gauss-Seidel solvers. This becomes evident if we compare N_{it} for $\delta = 0.5$ for this problem with that of test problem 2 (continuous permeability) for $\sigma = 0.5$ (which corresponds to $\delta = 0.5$).

The CPU time for the block Gauss-Seidel preconditioners increases quadratically with respect to mesh size h . Therefore, on a mesh with $h = 1/64$, it is predicted that the CPU times in Table 10, which are for a mesh with $h = 1/32$, would increase by a factor of four. So for test problem 3, in which the conductivity field is discontinuous, the predicted CPU times on a mesh with $h = 1/64$ with $p_u = 4$ are 240 and 2120 seconds, respectively, for $d = 4$ and $d = 6$. If these times are compared with the CPU times for MCM on the much simpler test problem 1 given in Table 1 ($t_{CPU} = 25\,000$ for $N_r = 40\,000$), then we see that the SMFEM with block GS preconditioner is at least an order of magnitude faster in terms of CPU time.

TABLE 10 bSGS and bGS iterations and solution timings—Test Problem 2

		$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
$\delta = \frac{\sigma}{\mu}$							
<i>d = 4</i>							
<i>bSGS</i>	0.5	11	0.78	13	4.2	14	19.64
	0.7	17	1.2	23	7.41	29	40.68
	1.0	32	2.25	53	17.09	84	117.66
	1.0,0.7,0.5,1.0	32	2.24	53	17.07	84	118.49
<i>bGS</i>	0.5	12	0.43	15	2.43	17	11.97
	0.7	17	0.6	25	4.07	33	23.32
	1.0	30	1.06	53	8.59	87	61.46
	1.0,0.7,0.5,1.0	30	1.07	53	8.58	87	61.11
<i>d = 6</i>							
<i>bSGS</i>	0.5	11	2.04	13	20.29	15	178.55
	0.7	18	3.33	24	37.48	31	343.28
	1.0	35	6.47	58	90.76	95	1048.38
	1.0,0.7,0.5,1.0	35	6.45	58	90.82	95	1049.12
<i>bGS</i>	0.5	12	1.12	15	11.73	17	94.08
	0.7	17	1.58	25	19.58	34	187.85
	1.0	30	2.79	53	41.42	96	529.96
	1.0,0.7,0.5,1.0	30	2.79	53	41.48	96	533.87

8.4 | Comparison and conclusions

In this section, a large number of methods have been tested to identify the most efficient solver for the stochastic formulation of the diffusion problem. To identify the most efficient and robust methods with respect to h , σ and discontinuous μ , the data presented in the previous tables are summarized in Figures 4 and 5. Only the case $p = 4$ is considered for $d = 4, 6$. The methods included in the figures are listed below.

1. CG with \mathcal{P}_{bdiag} (AMG);
2. CG with \mathcal{P}_{bdiag} (UMFPACK);
3. CG with \mathcal{P}_{mean} (AMG);
4. CG with \mathcal{P}_{mean} (UMFPACK);
5. CG with \mathcal{P}_{bSGS} (UMFPACK);
6. *bSGS* (UMFPACK);
7. *bGS* (UMFPACK).

Note that for the AMG case, the time required to construct the grids and smoother for the approximation is added to the solution times. The UMFPACK case does not require any set-up time.

Figures 4 and 5 show that the conjugate gradient solver preconditioned with \mathcal{P}_{bSGS} is the most efficient method for problems with medium/large standard deviation and discontinuous permeability. Gauss-Seidel solvers also perform well in these circumstances and for small σ they are in fact the best-performing methods.

Mean-based preconditioners are, in general, not robust and efficient for SFEM with lognormal distributions. There is very little difference in terms of performance between the AMG and UMFPACK versions of the preconditioner.

The outcome of this analysis reveals that CG preconditioned with \mathcal{P}_{bSGS} is robust and efficient, performing well in all settings considered in this article and therefore should generally be used for the solution of SFEM with lognormal distributions. Gauss-Seidel solvers represent a viable alternative to Krylov subspace iterative methods.

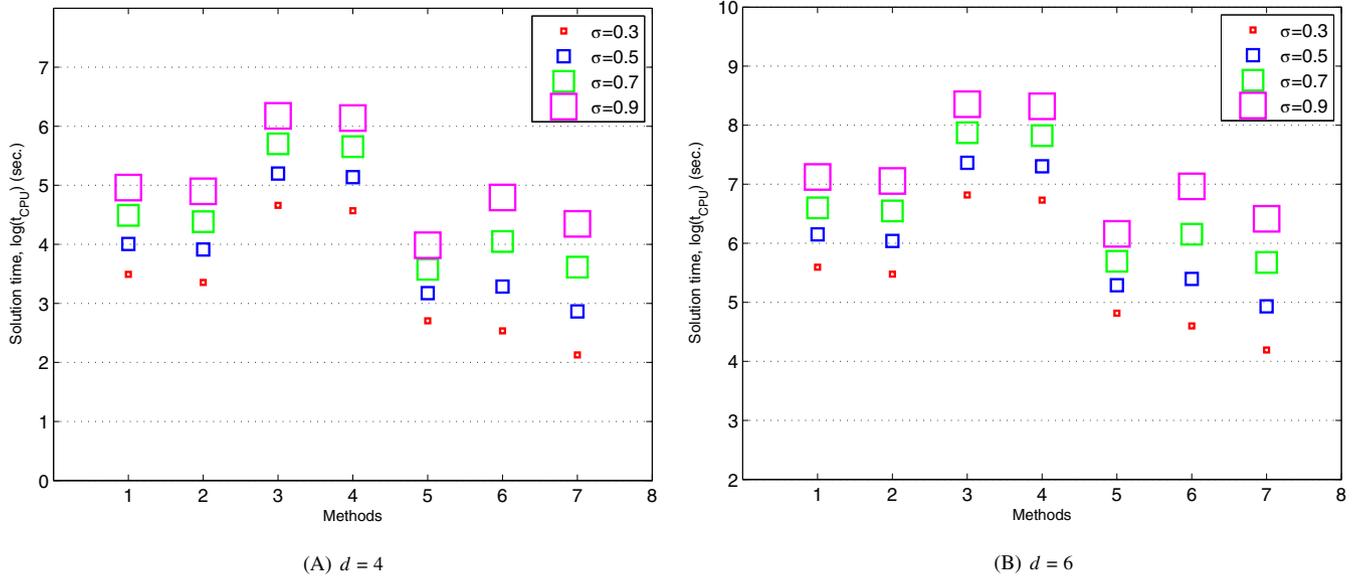


FIGURE 4 Comparison of methods for the solution of SFEM for test problem 1. A, $d = 4$. B, $d = 6$ [Colour figure can be viewed at wileyonlinelibrary.com]

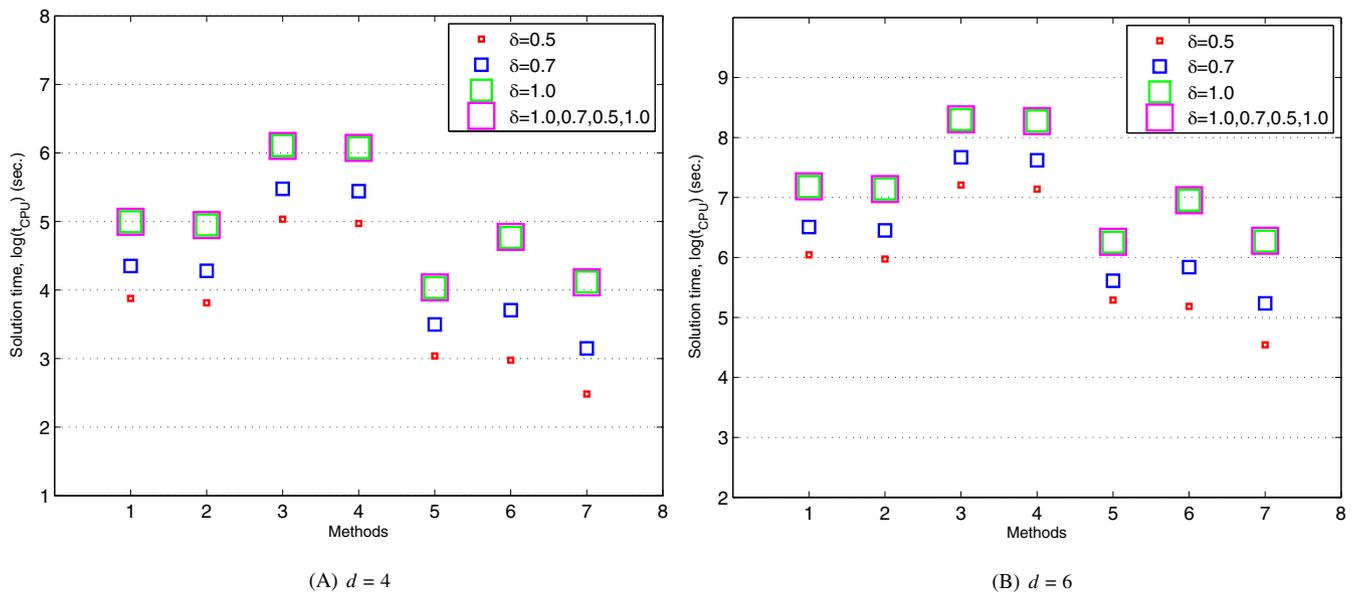


FIGURE 5 Comparison of methods for the solution of SFEM for test problem 2. A, $d = 4$. B, $d = 6$ [Colour figure can be viewed at wileyonlinelibrary.com]

9 | SMFEM SOLVERS

9.1 | Schur complement preconditioner

This section reports the performance of preconditioned MINRES for the system derived for SMFEM (cf. Section 7). The preconditioner used is the one described in Section 6.2. The Schur complement is computed exactly (using, eg, *UMFPACK*) or approximated using one V-cycle of *AMG* code.

TABLE 11 MINRES iterations and solution timings for \mathcal{P}_{comp} —Test Problem 2

	σ	$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
<i>d = 4</i>							
UMFPACK	0.3	62	7.32	75	49.05	86	330.55
	0.5	92	10.42	127	83.82	171	660.2
	0.7	136	15.71	226	149.58	346	1345.68
	0.9	206	23.47	397	262.34	698	2710.47
AMG	0.3	67	7.91	80	52.42	93	357.58
	0.5	97	11.13	136	88.92	182	700.29
	0.7	145	16.73	238	157.69	368	1422.33
	0.9	218	25.08	420	276.29	733	2846.35
<i>d = 6</i>							
UMFPACK	0.3	63	25.94	76	617.75	89	12 009.58
	0.5	95	39.68	134	1079.71	180	24 329.95
	0.7	143	59.29	237	1902.71	370	51 069.64
	0.9	216	90.11	424	3418.43	753	103 277.29
AMG	0.3	67	27.6	81	659.43	94	13 927.52
	0.5	100	41.21	141	1140.65	190	26 151.63
	0.7	150	61.97	249	2007.98	385	53 101.14
	0.9	227	94.27	440	3577.61	783	108 174.47

9.1.1 | Test problem 1—variable σ

The performance of the Schur complement preconditioner for varying σ is reported in Table 11. As for the previous case, the set-up time for the preconditioner is performed only once.

The results reported in Table 11 can be summarized as follows:

1. MINRES performance deteriorates significantly for increasing values of σ . This is in line with all methods considered in this article. However, for the nonlinear case, the use of SMFEM becomes impractical. In fact, the experiments show that for $\sigma = 0.9$, it takes more than 30 hours to solve the stochastic linear system for a very coarse discretization ($h = \frac{1}{32}$);
2. The performance of the AMG and UMFPACK versions of the Schur complement preconditioner is similar.

9.1.2 | Test problem 2—discontinuous-isotropic conductivity field

As has already been shown for other methods, the performance of the solver and preconditioners are not affected by spatial discontinuities in the permeability. In fact, the timings reported in Table 12 are comparable to those reported for the continuous test problem in Table 11.

Table 12 shows that the solver performance depends on the largest value of δ included in the domain. So, for example, for the case of variable δ (different coefficients of variation for the four sub-domains), MINRES performance is fully governed by the largest value of δ , that is, $\delta = 1$. In fact, the timings are almost equivalent to the case of constant $\delta = 1$ for all sub-domains.

		$p_u = 2$		$p_u = 3$		$p_u = 4$	
		N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)	N_{it}	t_{CPU} (sec)
$d = 4$							
<i>UMFPACK</i>	0.5	88	10.12	122	80.44	161	620.97
	0.7	130	14.88	211	139.07	316	1226.72
	1.0	235	27.15	474	314.05	864	3338.6
	1.0,0.7,0.5,1.0	235	27.03	473	314.9	864	3360.14
<i>AMG</i>	0.5	92	10.44	128	83.65	170	651.08
	0.7	136	15.55	222	145.62	333	1294.75
	1.0	247	28.29	499	329.87	898	3468.6
	1.0,0.7,0.5,1.0	247	28.39	498	327.16	898	3484.89
$d = 6$							
<i>UMFPACK</i>	0.5	91	38.03	126	1013.49	170	23 547.75
	0.7	135	56.16	223	1814.07	341	47 604.68
	1.0	250	104.13	513	4173.31	940	129 394.1
	1.0,0.7,0.5,1.0	250	104.89	513	4154.09	939	131 601.3
<i>AMG</i>	0.5	94	38.97	132	1062.56	177	24 152.55
	0.7	141	58.36	231	1870.85	355	49 131.72
	1.0	259	108.41	533	4334.59	984	136 578.2
	1.0,0.7,0.5,1.0	259	107.6	532	4287.98	984	136 140.52

TABLE 12 MINRES iterations and solution timings for \mathcal{P}_{scmp} —Test Problem 2

9.2 | Conclusions

While it was concluded that the performance of MINRES equipped with the Schur complement preconditioner described in (40) is acceptable for the solution of the stochastic mixed formulation in the linear case (see Part 1), the same cannot be concluded for the nonlinear case. The experiments reported in Tables 11 and 12 show that the CPU cost is too large (30 hours to solve test problem 2 on a coarse mesh, $h = \frac{1}{32}$) for this method to be used with lognormal random fields.

It becomes apparent that for the nonlinear case that it is crucial to include information contained in the off-diagonal blocks of the coefficient matrix into the preconditioned system. The Kronecker product preconditioner of Ullmann⁴ offers this possibility. Very recently, Powell and Ullmann¹¹ extended its implementation to the nonlinear case achieving a significant improvement in MINRES CPU cost. The authors also proposed $H(\text{div})$ preconditioning using augmenting schemes which, although being dependent on the choice of the augmentation parameter, seem to achieve very promising results.

10 | CONCLUSIONS

The performance of iterative methods for solving the linear systems of equations arising from stochastic finite element methods for single phase fluid flow in porous media is investigated. The uncertainty in the permeability is modeled using lognormal random fields and is characterized by its mean and standard deviation, which can vary spatially. In addition to being optimal with respect to the number of random variables in the KLE, the order of the polynomial chaos, and the finite element discretization parameter, the iterative solver is required to be optimal with respect to μ and σ . The mean and standard deviation may vary discontinuously throughout the domain. When the permeability possesses large standard deviation, it may fail to satisfy the condition (5). However, the condition can be satisfied by transforming the Gaussian random field into a lognormal one by projecting the KLE onto the polynomial chaos of order at most p .^{6,26} The

coefficient matrix arising from the discretization of (1) and (6) using lognormal random fields becomes block dense and ill-conditioned, which makes the linear systems very difficult and expensive to solve.

The conjugate gradient method was used to solve the linear systems of equations arising from stochastic finite element discretization of the primal formulation. Block diagonal, mean-based, and block Gauss-Seidel preconditioners were implemented to accelerate the convergence of the CG method. Two versions of the block Gauss-Seidel preconditioner were implemented—one based on a sparse direct solver and the other on AMG. For small values of σ , the preconditioner based on block Gauss-Seidel was found to be optimal with respect to h , d , and p . This preconditioner also considerably reduced the number of CG iterations in comparison to the block diagonal preconditioner. For $\sigma \geq 0.5$, this optimality is lost. However, the block Gauss-Seidel preconditioner still outperforms the block diagonal preconditioner and the improvement is greater as the mesh is refined. Furthermore, the growth in the number of CG iterations is more gradual as σ increases compared with the block diagonal preconditioner. The introduction of spatially varying random fields does not deteriorate the performance of the preconditioners. In other words, the performance is no worse than when a constant random field corresponding to the largest standard deviation used in the discontinuous case is used. The outcome of this analysis reveals that the AMG version of the block Gauss-Seidel preconditioner is very efficient and the most robust solver considered for the solution of SFEM linear systems (linear case). However, although CG with a block Gauss-Seidel preconditioner performs better than with block-diagonal and mean preconditioners, these methods are always outperformed by stand-alone Gauss-Seidel solvers when solving these systems.

For the mixed formulation, MINRES was used to solve the linear systems of equations arising from the SMFEM. A preconditioner based on the Schur complement was used. For small values of σ , optimal convergence behavior was obtained with respect to h and d . There was a weak dependence on p . However, the performance of the preconditioner deteriorated significantly for moderate to large values of σ . As for the primal formulation, the introduction of discontinuous conductivity coefficients had little impact on the performance of the preconditioner. However, SMFEM becomes more expensive than SFEM for this class of problems.

Although computationally more expensive than SFEM, the efficient solution of SMFEM is largely dependent on the preconditioner used with the chosen iterative solver. The numerical experiments showed that the Schur complement preconditioner is h -optimal not only when the complement is inverted exactly but also when it is inverted approximately using one V-cycle of AMG.

The experiments also showed that the preconditioner is not robust with respect to the permeability. The preconditioner that only uses the diagonal of the coefficient matrix A is robust when the permeability possesses small standard deviations but generally inadequate for large standard deviations. In the latter case, in fact, the off-diagonal blocks of A become significantly more important and these are not included in this preconditioner. This lack of robustness was also encountered when using the mean-based preconditioner for the solution of linear systems obtained by SFEM. The off-diagonal blocks of the coefficient matrix were successfully incorporated into a preconditioner by means of a symmetric block Gauss-Seidel algorithm. Unfortunately, due to the structure (and specifically the presence of a zero-block) of the coefficient matrix of the SMFEM system, the same approach cannot be used for the Schur complement preconditioner.

The efficient solution of discrete linear systems obtained from stochastic mixed formulations is currently a very active research area. The Kronecker product preconditioner proposed by Ullmann⁴ significantly reduces MINRES iteration counts. However, this does not always translate into improvements in CPU performance. In fact, it is shown that the Schur complement preconditioner performs better (in terms of CPU cost) than the Kronecker preconditioner for test problems in which the permeability possesses a large standard deviation.⁴ More recently, Ullmann and Powell³⁰ introduced an approximation to the Schur complement as preconditioner that is robust with respect to the discretization parameters and only slightly sensitive to the statistical properties of the permeability.

ORCID

Timothy Nigel Phillips  <https://orcid.org/0000-0001-6455-1205>

REFERENCES

1. Graham G, Kuo F, Nuyens D, Scheichl R, Sloan IH. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J Comput Phys*. 2011;230:3668-3694.
2. Cliffe KA, Giles MB, Scheichl R, Teckentrup AL. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput Visualiz Sci*. 2011;14:3-15.
3. Babuška I, Tempone R, Zouraris GE. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J Numer Anal*. 2004;42(2):800-825.

4. Ullmann E. Solution strategies for stochastic finite element discretizations (PhD Thesis). Technische Universität Bergakademie Freiberg; 2008.
5. Ghanem RG, Spanos PD. *Stochastic Finite Elements: A Spectral Approach*. New York, NY: Springer-Verlag; 1991.
6. Ghanem RG, Spanos PD. *Stochastic Finite Elements*. Mineola, NY: Dover Publications Inc; 2003.
7. Ghanem RG, Kruger RM. Numerical solution of spectral stochastic finite element systems. *Comput Methods Appl Mech Eng*. 1996;129:289-303.
8. Pellissetti MF, Ghanem RG. Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Adv Eng Softw*. 2000;313:607-616.
9. Powell CE, Elman CH. Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA J Numer Anal*. 2009;29(2):350-375.
10. Ernst OG, Powell CE, Silvester D, Ullmann E. Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data. *SIAM J Sci Comput*. 2009;31(2):1424-1447.
11. Powell C, Ullmann E. Preconditioning stochastic Galerkin saddle point systems. *SIAM J Matrix Anal A*. 2010;31:2813-2840.
12. Rossell E, Vandewalle S. Iterative solvers for the stochastic finite element method. *SIAM J Sci Comp*. 2010;32(1):372-397.
13. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. New York, NY: Springer-Verlag; 1991.
14. Traverso L, Phillips TN, Yang Y. Efficient stochastic finite element methods for flow in heterogeneous porous media. Part 1: Gaussian conductivity coefficients. *Int J Numer Meth Fluids*. 2013;74:359-385.
15. Deb MK, Babuška I, Oden JT. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Comput Methods Appl Mech Eng*. 2001;190(48):6359-6372.
16. Matthies HG, Keese A. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput Methods Appl Mech Eng*. 2005;194:1295-1331.
17. Lu Z, Zhang D. Stochastic simulations for flow in nonstationary randomly heterogeneous porous media using a KL-based moment-equation approach. *SIAM Multiscale Model Simul*. 2007;6(1):228-245.
18. Xiu D, Karniadakis GE. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Comput Methods Appl Mech Eng*. 2002;191:4927-4948.
19. Xiu D, Karniadakis GE. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput*. 2002;24(2):619-644.
20. Constantine PG. Spectral methods for parameterized matrix equations (PhD Thesis). Stanford University; 2009.
21. Gelhar LW. *Stochastic Subsurface Hydrology*. Upper Saddle River, NJ: Prentice-Hall; 1983.
22. Rubin Y. *Applied Stochastic Hydrogeology*. Oxford, UK: Oxford University Press; 2003.
23. Freeze RA. A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media. *Water Resour Res*. 1975;11(5):725-741.
24. Ghanem R. The nonlinear Gaussian spectrum of lognormal stochastic processes and variables. *ASME J Appl Mech*. 1999;66(4):964-973.
25. Ghanem R. Stochastic finite elements for heterogeneous media with multiple random non-gaussian properties. *J Appl Mech T ASME*. 1999;125(1):26-40.
26. Sudret B, Kiureghian AD. Stochastic finite element methods and reliability, a state-of-the-art-report. *Technical Report UCB/SEMM-2000/08*, University of California; 2000.
27. Keese A. Numerical solution of systems with stochastic uncertainties a general purpose framework for stochastic finite elements (PhD thesis). Technische Universität Braunschweig; 2004.
28. Furnival DG. Iterative methods for the stochastic diffusion problem (PhD Thesis). University of Maryland; 2008.
29. Elman HE, Furnival DG, Powell CE. H(div) preconditioning for a mixed finite element formulation of the diffusion problem with random data. *Math Comput*. 2010;79(270):733-760.
30. Ullmann E, Powell CE. Solving log-transformed random diffusion problems by stochastic Galerkin mixed finite element methods. *SIAM/ASA J Uncert Quantif*. 2015;3(1):509-534.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Traverso L, Phillips TN. Efficient stochastic finite element methods for flow in heterogeneous porous media. Part 2: Random lognormal permeability. *Int J Numer Meth Fluids*. 2020;92:1626–1652. <https://doi.org/10.1002/flid.4842>