Stochastic Preconditioning of Domain Decomposition Methods for Elliptic Equations with Random Coefficients

Joao Felício Dos Reis, Olivier P Le Maître, Pietro M Congedo, Paul Mycek

Stochastic Preconditioning of Domain Decomposition Methods for Elliptic Equations with Random Coefficients

João F. Reis[a,*], Olivier P. Le Maître[b], Pietro M. Congedo[a], Paul Mycek[c]

[a] *Inria, CMAP, CNRS, École Polytechnique, IPP, Palaiseau, France*
[b] *CMAP, CNRS, Inria, École Polytechnique, IPP, Palaiseau, France*
[c] *Cerfacs, Toulouse, France*

Abstract

This paper aims at developing an efficient preconditioned iterative domain decomposition (DD) method for the sampling of linear stochastic elliptic equations. To this end, we consider a non-overlapping DD method resulting in a Symmetric Positive Definite (SPD) Schur system for almost every sampled problem. To accelerate the iterative solution of the Schur system, we propose a new stochastic preconditioning strategy that produces a preconditioner adapted to each sampled problem and converges toward the ideal preconditioner (*i.e.*, the Schur operator itself) when the numerical parameters increase. The construction of the stochastic preconditioner is trivially parallel and takes place in an off-line stage, while the evaluation of the sample's preconditioner during the sampling stage has a low and fixed cost. One key feature of the proposed construction is a factorized form combined with Polynomial Chaos expansions of local operators. The factorized form guaranties the SPD character of the sampled preconditioners while the local character of the PC expansions ensures a low computational complexity. The stochastic preconditioner is tested on a model problem in 2 space dimensions. In these tests, the preconditioner is very robust and significantly more efficient than the deterministic median-based preconditioner, requiring, on average, up to 7 times fewer iterations to converge. Complexity analysis suggests the scalability of the preconditioner with the number of subdomains.

Keywords: Stochastic Preconditioner, Sampling Method, Domain Decomposition, Parallel Computation, Preconditioned Conjugate Gradient Method

1. Introduction

Efficient solution methods for stochastic partial differential equations (SPEs) are critical due to the spread of computational and simulation approaches in sciences and engineering, which calls for the characterization of the model's uncertainty and variability in operating conditions. In this context, the availability of robust solvers designed to tackle the specific task of uncertainty quantification, probabilistic inference, and sampling schemes, constitutes a crucial aspect of extending and promoting the use of advanced practices of uncertainty analysis and management. The present work focuses on a particular type of SPDEs: the elliptic equations with stochastic coefficients. This choice is motivated by the omnipresence of elliptic equations in many scientific domains (elasticity, porous media flows, electromagnetics, steady diffusion problems, ...), which make the development of an elliptic equation solver applicable to many application fields.

The stochastic elliptic equation has been used in multiple works and serves as a benchmark problem for testing and comparing solution methods for UQ problems. Two classes of methods exist for the resolution of SPDEs: the simulation methods and the functional representation methods. Simulation methods rely on

*Corresponding author
Email addresses: joao.reis@inria.fr (João F. Reis), olivier.le-maitre@polytechnique.edu (Olivier P. Le Maître), pietro.congedo@inria.fr (Pietro M. Congedo), mycek@cerfacs.fr (Paul Mycek)

samples (or realizations) of the model's solution, corresponding to particular values of the coefficient selected randomly or deterministically, to estimate statistics of quantities of interest [1, 2]. Therefore, simulation methods associate deterministic solvers with sampling and statistical estimation procedures. The weakness of simulation methods is generally the low convergence rate of statistical estimators. Consequently, most of the efforts to improve simulation methods have concerned this aspect (let us mention, for instance, the multilevel MC method [3] to improve convergence rates) while the deterministic solver is not concerned with the computational optimization and taken "as is." In the second class of methods, the functional approximation, one approximates the functional dependencies of the quantity of interest (or directly the model solution) on the stochastic coefficients. These methods include the extensively studied spectral methods [4, 5] which have been applied to numerous linear and non-linear PDEs with random coefficients [6, 7, 8, 9, 10, 11, 12, 13]. An issue of the spectral method is the need to introduce a discretization of the random coefficient using a finite set of random variables. Problems with complex uncertainty sources require many random variables for their parametrization, resulting in a high-dimensional functional approximation problem. To temper the curse of dimensionality in this situation, it has been proposed to exploit structures in the dependences by deriving low-rank representations in suitable tensor formats (for elliptic problems see [14, 15, 16, 17, 18, 19, 20, 21]). Some construction methods for functional approximations, often termed non-intrusive methods, rely on samples (observations) of the model solution (*e.g.*, regression methods [22] and spectral projection methods [23, 24, 25, 26]). Similar to the simulation methods, the literature on non-intrusive methods quite overlooks the role of the deterministic solver in the construction cost, to focus on the minimization of the number of solves to get the approximation. It appears that intrusive (Galerkin) strategies are gathering the essentials of the work on solvers (see, *e.g.* [27, 28, 8] and references below for domain decomposition methods).

The present work is not restricted to a particular UQ method but aims at reducing the computational cost related to the generation of the samples in a generic sampling-based approach (which could be a Monte Carlo or non-intrusive method). We target stochastic elliptic problems with complex stochastic coefficient fields requiring a high-dimensional parametrization, making straightforward spectral methods prohibitively costly (for domain decomposition methods in the context of Galerkin methods, we refer to [29, 30, 31, 32]), and for which more advanced functional representations would demand large sample sets for their construction.

For the acceleration of the sample computation, we build on the previous works on domain decomposition (DD) methods for stochastic elliptic problems published in [33, 34]. Precisely, we consider linear problems leading, after spatial discretization, to a symmetric positive definite (SPD) system with size not amenable to direct solution methods and requiring iterative strategies [35]. The spatial discretization is a standard finite-element (FE) method, but the approach proposed in the paper can be extended to other discretization procedures amenable to a non-overlapping domain decomposition method. A non-overlapping partition of the domain is then introduced to results in a set of local (small size) FE problems related by their boundary conditions. The FE problem can be condensed to form a Schur complement problem for the subdomains' boundary values [36, 37, 38]. The Schur problem's size is much smaller than the original problem and can be solved iteratively, without having to form the Schur system explicitly. However, in most situations, the preconditioning of the iterative method is necessary to obtain high computational performances. For the preconditioning, one can use a different preconditioner for each sample, providing that the determination and set-up times of the preconditioner are not too significant. In practice, the latter condition prevents the on-line construction of highly efficient preconditioners and favors moderately effective ones requiring less analysis of the system to solve. Alternatively, one can use for all samples the same high-quality preconditioner, factorizing its determination and set-up cost over multiple samples. A classical strategy [29], in the context of the sampled stochastic system, consists of selecting the preconditioner of a particular deterministic system (often the mean or median of the stochastic system) to precondition all samples. However, a unique deterministic preconditioner is not adequate when the stochastic system has high variability, motivating the use of a stochastic (sample dependent) preconditioner constructed off-line and with low on-line evaluation costs.

In [34], the authors proposed constructing, in an off-line stage, a spectral approximation of the stochastic Schur problem. The stochastic approximation consists of a summation over the subdomains' contribution that enables the use of low-dimensional local parametrizations of the stochastic coefficient and local Polyno-

mial Chaos (PC) expansions. The use of local parametrizations to reduce the stochastic dimension follows ideas similar to the works in [39, 40, 41, 42]. The numerical results of [34] proved the convergence of the approximation to the exact stochastic Schur problem when the stochastic discretization parameters (PC order and the number of local random variables) increase. Subsequently, in the on-line stage, the approximation of the Schur problem is sampled and solved to generate samples of the subdomains' boundary values; the corresponding global solutions are retrieved solving local problems only. Although the approach of [34] presents the clear advantage of bypassing all local solves of the iterative resolution of the Schur problem, it yields a solution that does not exactly satisfy the original FE problem, because the boundary values solve an approximate Schur problem. The central idea of the present work is then to exploit the approximated Schur problem to precondition the iterative solution of the sampled Schur problem. In this context, our contribution constitutes an alternative to classical preconditioning techniques for sampled stochastic problems.

The organization of the paper is as follows. Section 2 briefly introduces the stochastic elliptic equation, the sampling approach, and the deterministic DD method and Schur system. Section 3 concerns the preconditioning of the sampled Schur complement system; we start with the median-based preconditioner before introducing the stochastic Schur system in Sections 3.1 and 3.2. Section 3.3 introduces the PC expansions of the local influence operators that form the approximation of the stochastic Schur operator, while Section 3.4 discusses alternative factorizations of the influence operators aiming at ensuring an almost surely SPD approximation of the Schur operator. Finally, Section 3.5 summarizes the proposed approach and provides sketches of implementation in the form of algorithms. In Section 4, we present extensive numerical tests to assess the performance of the proposed approach. The limitation of the median-based preconditioner are first illustrated (Section 4.1), before contrasting the performances of the proposed preconditioners (Sections 4.2-4.3). The impact of the numerical parameters of the preconditioner is carefully analyzed in Section 4.4; a brief complexity analysis follows in Section 4.5, with a discussion synthesizing the findings in Section 4.6. We close the paper with some final remarks and prospective research directions in Section 5.

2. Sampling method for Stochastic Elliptic Equations

We are interested in computing statistics from some functional of the solution of a stochastic elliptic equation. This section provides some mathematical background and notations, before introducing the stochastic elliptic equation, the generic sampling method and finally a brief overview of DD and the Schur complement method.

2.1. Deterministic and Stochastic spaces

Let $x = (x_1, \dots, x_n) \in \Omega \subseteq \mathbb{R}^n$ be the n-dimensional spatial domain with boundary $\partial\Omega$. Consider the space of square-integrable functions $f : x \in \Omega \mapsto f(x) \in \mathbb{R}$, denoted by $L^2(\Omega)$. The space $L^2(\Omega)$ is a Hilbert space when equipped with the inner product $\langle \cdot, \cdot \rangle_\Omega$ and associated norm $\| \cdot \|_\Omega$ defined as follows

$$\forall f, g \in L^2(\Omega), \quad \langle f, g \rangle_\Omega \doteq \int_\Omega f(x) g(x)\, dx, \quad \|f\|_\Omega = \langle f, f \rangle_\Omega^{1/2} < +\infty. \tag{1}$$

The subspace of the space of square-integrable functions with square-integrable spatial derivatives, denoted by $H^1(\Omega) \subset L^2(\Omega)$, is defined as

$$H^1(\Omega) \doteq \left\{ f(x) \in L^2(\Omega) : \partial_{x_i} f(x) \in L^2(\Omega),\ i = 1, \dots, n \right\}. \tag{2}$$

Let $\mathcal{P} \doteq (\Theta, \Sigma_\Theta, \mu_\Theta)$ denote a probability space, Θ a set of random events, Σ_Θ a sigma-algebra associated with Θ and μ_Θ probability measure. The space of second-order random variables $\boldsymbol{u} : \theta \in \Theta \mapsto \boldsymbol{u}(\theta) \in \mathbb{R}$, such that $\mathbb{E}\left[\boldsymbol{u}^2\right] < \infty$ is denoted by $L^2(\Theta)$. The expectation operator $\mathbb{E}[\cdot]$ is defined, for any random variable $\boldsymbol{u}$, as

$$\mathbb{E}[\boldsymbol{u}] \doteq \int_\Theta \boldsymbol{u}(\theta)\, d\mu_\Theta(\theta). \tag{3}$$

The space $L^2(\Theta)$ is again a Hilbert space when equipped with the inner product

$$\forall \boldsymbol{u}, \boldsymbol{v} \in L^2(\Theta), \quad \langle \boldsymbol{u}, \boldsymbol{v} \rangle_\Theta \doteq \mathbb{E}[\boldsymbol{u}\boldsymbol{v}], \tag{4}$$

and associated norm $\|\boldsymbol{u}\|_\Theta \doteq \langle \boldsymbol{u}, \boldsymbol{u} \rangle_\Theta^{1/2}$.

We define the space of second-order stochastic processes $\boldsymbol{u} : (x, \theta) \in \Omega \times \Theta \mapsto \boldsymbol{u}(x, \theta) \in \mathbb{R}$, and denote it by $L^2(\Omega, \Theta)$. This Hilbert space is equipped with the inner product

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\Omega \times \Theta} \doteq \mathbb{E}[\langle \boldsymbol{u}(x, \theta), \boldsymbol{v}(x, \theta) \rangle_\Omega]. \tag{5}$$

2.2. Stochastic Elliptic Equation

The stochastic elliptic equation we are interested in has the form

$$\begin{aligned} &\nabla \cdot [\boldsymbol{\kappa}(x, \theta) \nabla \boldsymbol{u}(x, \theta)] = -f(x) \quad x \in \Omega,\, \theta \in \Theta \\ &\boldsymbol{u}(x, \theta) = 0, \quad x \in \partial\Omega,\, \theta \in \Theta, \end{aligned} \tag{6}$$

where $f(x)$ is a deterministic source, $u_{\partial\Omega}$ the deterministic Dirichlet boundary datum, and $\boldsymbol{\kappa}$ is the stochastic coefficient field of the equation. The equalities in the equations of (6) stand in the $\mathcal{P}$-almost surely sense and for almost every x. The developments below readily extend to the case of deterministic or stochastic inhomogeneous Dirichlet boundary conditions by writing the sought solution as $\boldsymbol{u}(x, \theta) = \boldsymbol{u}_0(x, \theta) + \boldsymbol{u}_{BC}(x, \theta)$, where $\boldsymbol{u}_{BC}(x, \theta)$ is given and satisfies the boundary conditions, while $\boldsymbol{u}_0(x, \theta)$ solves (6) with the modified right-hand-side $-f(x) - \nabla \cdot [\boldsymbol{\kappa}(x, \theta) \nabla \boldsymbol{u}_{BC}(x, \theta)]$.

Problem (6) is well posed and $\boldsymbol{u}(x, \theta) \in L^2(\Omega, \Theta)$ with $\boldsymbol{u}(x, \cdot) \in H^1(\Omega)$ a.s. provided that the coefficient $\boldsymbol{\kappa}$ satisfies some mild conditions [43]. In this work, we restrict ourselves to the case of $\boldsymbol{\kappa}$ being a stationary log-normal stochastic process, whose log is a centered Gaussian process $\boldsymbol{G}$ with covariance function C:

$$\boldsymbol{G}(x, \theta) \doteq \log \boldsymbol{\kappa}(x, \theta) \sim \mathcal{N}(0, C). \tag{7}$$

Without loss of generality, we take for $C : (x, x') \in \Omega \times \Omega \mapsto \mathbb{R}$ as

$$C(x, x') \doteq \sigma^2 \exp\left(-\frac{\|x - x'\|_\Omega^\gamma}{\gamma {\ell_c}^\gamma} \right), \tag{8}$$

with variance $\sigma^2 \in \mathbb{R}_+$, correlation length $\ell_c \in \mathbb{R}_+$ and regularity parameter $\gamma \in [1, 2]$.

2.3. Sampling Method

We now briefly outline the sampling approach, in the context of the Monte Carlo method to compute statistics from the solution of equation (6). Let $z(\boldsymbol{u})$ be a real-valued functional of the solution; for instance, we may consider $z(\boldsymbol{u}) = \boldsymbol{u}(y, \theta)$, for some $y \in \Omega$, or

$$z(\boldsymbol{u}) = \int_{\Omega' \subset \Omega} \boldsymbol{u}(y, \theta) dy. \tag{9}$$

We are interested in approximating the $\mathbb{E}[z(\boldsymbol{u})]$ using a sampling (Monte Carlo) method. This amounts to estimate $\mathbb{E}[z(\boldsymbol{u})]$ as

$$\mathbb{E}[z(\boldsymbol{u})] \approx \frac{1}{M} \sum_{m=1}^{M} z\left(u^{(m)} \right), \tag{10}$$

where $u^{(m)} \in H^1(\Omega)$ is a random sample of the solution $\boldsymbol{u}(x, \theta)$ of (6) for the sampled coefficient value $\kappa^{(m)}$:

$$\begin{aligned} &\nabla \cdot \left(\kappa^{(m)} \nabla u^{(m)} \right) = -f, \quad x \in \Omega, \\ &u^{(m)} = 0, \quad x \in \partial\Omega. \end{aligned} \tag{11}$$

The random estimate in (10) is unbiased, provided that the $\kappa^{(m)}$ are drawn randomly, and has an error whose variance is $\mathbb{V}[z(\boldsymbol{u})]/M$. Then, a large set of solution samples must be computed to ensure that the sampling error $\mathcal{O}(M^{1/2})$ is small enough, and thus to have an accurate approximation of $\mathbb{E}[z(\boldsymbol{u})]$. The size of the sample set entails a significant computational effort.

2.4. Domain Decomposition and the Schur Complement System

Let us now introduce a divide to parallelize strategy to compute the solution of each problem (11). We start by describing the decomposition of the domain Ω, which will be the foundation of the method proposed in this work. For simplicity of notation, in the rest of this section we drop the sample index (m) in the definition of problem (11).

Consider a partition of Ω into D subdomains $\Omega^{(d)}$, each with boundary $\partial\Omega^{(d)}$, where $\overline{\cup_{d=1}^{D}\Omega^{(d)}} = \Omega$. The subdomains can overlap, if $\Omega^{(d)} \cap \Omega^{(d')} \neq \emptyset$, or be non-overlapping, when $\Omega^{(d)} \cap \Omega^{(d')} = \emptyset$ for all pairs of distinct subdomains. In this work we restrict ourselves to the case of non-overlapping partitions. We denote by $\Gamma^{(d)}$ the part of boundary $\partial\Omega^{(d)}$ that does not include $\partial\Omega$, and the union of all such boundaries of all subdomains will be called the internal boundaries of Ω and denoted by $\Gamma \doteq \cup_{d=1}^{D}\Gamma^{(d)}$. The resolution of problem (11) can be reduced to determining u_Γ such that the solutions w_d of the local problems,

$$\begin{aligned} &\nabla \cdot [\kappa \nabla w_d] = -f \quad x \in \Omega^{(d)}, \\ &w_d = u_\Gamma, \quad x \in \Gamma^{(d)}, \\ &w_d = 0, \quad x \in \overline{\Omega^{(d)}} \cap \partial\Omega \end{aligned} \tag{12}$$

satisfy some compatibility conditions at the internal boundaries.

Now we introduce the particular domain decomposition (DD) method used in this work. We start be introducing the discrete version of the deterministic problem (11). Let $\mathcal{T}$ be a triangulation of Ω and denote by $\mathcal{N}$ the set of nodes in $\mathcal{T}$ that belong to $\Omega \setminus \partial\Omega$. The cardinality of $\mathcal{N}$ is Nod. We denote by $\{\Phi_l\}_{l=1}^{\text{Nod}}$ the finite element basis and approximate the solution u as

$$H^1(\Omega) \ni u(x) \approx \sum_{l=1}^{\text{Nod}} \Phi_l(x) u_l, \tag{13}$$

where u_l is the nodal value. Let $\mathcal{N}_\Gamma^{(d)}$ be the set nodes on $\partial\Omega^{(d)} \backslash \partial\Omega$, with cardinality $\mathrm{N}_\Gamma^{(d)}$. Define the set of all *internal* boundary nodes as $\mathcal{N}_\Gamma = \bigcup_{d=1}^{D} \mathcal{N}_\Gamma^{(d)}$, with cardinality N_Γ. Let $\mathcal{N}_{\text{in}}$ denote the set of interior nodes that belong to the $\mathcal{N} \setminus \mathcal{N}_\Gamma$. Proceeding with a FE discretization and Galerkin approach [34], and dropping again the sample index, problem (11) can be recast in the finite dimensional system

$$[\mathrm{A}]\mathrm{u} = \mathrm{b}, \tag{14}$$

where the solution is defined as vector of nodal values $\mathrm{u} = (u_1 \cdots u_{\text{Nod}})^\top$. The FE matrix $[\mathrm{A}] \in \mathbb{R}^{\text{Nod}\times\text{Nod}}$ is assumed to be symmetric positive definite (SPD) with entries

$$[\mathrm{A}]_{l,l'} = \int_\Omega \kappa(x) \nabla\Phi_l(x) \cdot \nabla\Phi_{l'}(x) dx, \tag{15}$$

while the right-hand side is

$$\mathrm{b}_l = \int_\Omega f(x)\Phi_l(x) dx. \tag{16}$$

System (14) can be reorganized in the form

$$\begin{bmatrix} [\mathrm{A}_{\Gamma,\Gamma}] & [\mathrm{A}_{\Gamma,\text{in}}] \\ [\mathrm{A}_{\text{in},\Gamma}] & [\mathrm{A}_{\text{in},\text{in}}] \end{bmatrix} \begin{bmatrix} \mathrm{u}_\Gamma \\ \mathrm{u}_{\text{in}} \end{bmatrix} = \begin{bmatrix} \mathrm{b}_\Gamma \\ \mathrm{b}_{\text{in}} \end{bmatrix}. \tag{17}$$

The Schur complement of the discrete DD problem is given by the matrix $[\mathrm{S}] \in \mathbb{R}^{\mathrm{N}_\Gamma \times \mathrm{N}_\Gamma}$ defined as

$$[\mathrm{S}] \doteq [\mathrm{A}_{\Gamma,\Gamma}] - [\mathrm{A}_{\Gamma,\text{in}}][\mathrm{A}_{\text{in},\text{in}}]^{-1}[\mathrm{A}_{\text{in},\Gamma}]. \tag{18}$$

This gives the Schur system

$$[\mathrm{S}]\mathrm{u}_\Gamma = \mathrm{b}_\mathrm{S}, \quad \mathrm{b}_\mathrm{S} \doteq \mathrm{b}_\Gamma - [\mathrm{A}_{\Gamma,\text{in}}][\mathrm{A}_{\text{in},\text{in}}]^{-1}\mathrm{b}_{\text{in}}. \tag{19}$$

Classically, system (19) is solved by a matrix-free iterative method, since applying [S] to a given iterate u_Γ amounts to solving local problems (expressed by the matrix operator $[\mathrm{A}_{\mathrm{in,in}}]^{-1}$). The matrix [S] is SPD such that classical Conjugate Gradient (CG) methods can be applied. In practice, the conditioning of [S] degrades as N_Γ increases and preconditioners are necessary to ensure a convergence of the iterates to u_Γ in a decent number of iterations. Several approaches have been proposed to precondition system (19) [36, 37, 38]. In the following, we introduce different kinds of preconditioning strategies that are suited for solving the deterministic problems (11) for a large number of samples.

3. Stochastic Preconditioners for the Schur Complement Systems

In the previous section, we introduced the sampling approach to estimate statistics from some functional of the solution of equation (6). In the DD approach, each solution sample is computed by solving the Schur complement associated with the sample's deterministic problem in (11). Therefore, we have to solve many Schur systems (19) corresponding to different samples of $\boldsymbol{\kappa}$. In practice, solving for each realization of $\boldsymbol{\kappa}$ the Schur system (19) by a direct approach is costly as it demands to solve many local problems to assemble [S]. Instead, it is usually more effective to solve (19) iteratively without assembling [S]. In that case, it is crucial to use an effective Preconditioned CG (PCG) method to achieve converged statistics in acceptable computational times. The preconditioner should ensure a sufficient convergence rate for all sampled problems, while its set-up time per sample should be minimal.

A minimal set-up time is achieved when the same deterministic preconditioner is used for all samples. Obviously such deterministic preconditioner must be carefully selected to provide suitable convergence for all samples. In Section 3.1, we introduce the preconditioner corresponding to the Schur system for the median value of the field. Alternatives to fixed deterministic preconditioners are stochastic preconditioners that adapt to each sample. Sample-dependent preconditioners can demand a significant extra computational effort to be set-up. However, they are potentially more effective than deterministic ones. If the gain in convergence rate overcomes the additional computational burden, sample-dependent strategies are preferable. Of course, one could consider to rely on a existing deterministic preconditioning method and set-up from scratch the preconditioner adapted to each sample of κ. Along this idea, it is necessary to balance the average set-up cost with the average iteration cost when selecting and setting a particular preconditioning approach. In this paper, we propose a novel sample-dependent preconditioner based on a surrogate of $[\mathbf{S}](\theta)$. The surrogate construction relies on PC expansions of local operators, which can be carried out in parallel in a preprocessing stage. In doing so, most of the set-up cost of our preconditioner is pushed off-line, such that the construction time is factorized over the subsequent computation of arbitrary many samples. The remaining in-line set-up cost for each sample, on the contrary, is low and weakly dependent on the stochastic discretization parameters that control the performance of the preconditioner. This latter characteristic is to be contrasted with alternative approaches where improvements of the preconditioner usually come with more expensive set-up procedures for each sample.

3.1. Deterministic preconditioner

One strategy consists of constructing a single deterministic preconditioner to be used for all samples. This approach is attractive because the construction time of the preconditioner and possibly its decomposition is factorized over a large number of samples. Denoting by $[\mathbf{S}](\theta)$ the stochastic Schur operator derived below, and $\overline{[\mathrm{S}]}$ the deterministic preconditioner, we want to ensure that $[\mathbf{S}](\theta)^{-1}\overline{[\mathrm{S}]}$ is close to the identity for almost all events θ. One straightforward choice is to define $\overline{[\mathrm{S}]}$ as the average of $[\mathbf{S}](\theta)$, but the construction would demand the evaluation of several samples of $[\mathbf{S}](\theta)$. Instead, we can proceed through the direct deterministic construction of $\overline{[\mathrm{S}]}$ using the average or median value of the stochastic coefficient field. In the following, we shall consider the deterministic preconditioner $\overline{[\mathrm{S}]}$ constructed on the median $\bar{\kappa}$ of $\boldsymbol{\kappa}$, therefore the notation $\overline{[\mathrm{S}]} = [\mathrm{S}](\bar{\kappa})$. Specifically, $\overline{[\mathrm{S}]}$ corresponds to the reduction of the matrix $\overline{[\mathrm{A}]}$ in (18), with entries

$$\overline{[\mathrm{A}]}_{l,l'} \doteq \int_\Omega \bar{\kappa}(x)\nabla\Phi_l(x)\cdot\nabla\Phi_{l'}(x)dx \quad \forall\, l,l' \in \mathcal{N}. \tag{20}$$

We remark that $\overline{[\mathrm{S}]}$ is SPD, being based on a particular realization of the stochastic elliptic problem. We call the CG method preconditioned by $\overline{[\mathrm{S}]}$ the Median Preconditioned CG (MPCG) method.

As will be evidenced in Section 4, deterministic preconditioners based on a statistic of the coefficient field can be ineffective because they tend to neglect spatial variability and heterogeneities in the realizations of $\boldsymbol{\kappa}$. In particular, for the stationary fields considered in this work, any statistic of $\boldsymbol{\kappa}$ is spatially constant, while realizations can exhibit large deviations in Ω when the variance is significant and the field not too correlated.

3.2. Stochastic Schur Complement System

The stochastic Schur complement system is the stochastic counterpart of system (19). The solution of problem (6) is again approximated in a FE space V^h using *random nodal values* $\boldsymbol{u}_l(\theta)$:

$$\boldsymbol{u}(x,\theta) \approx \sum_{l=1}^{\mathrm{Nod}} \Phi_l(x)\boldsymbol{u}_l(\theta) \in V^h \times L^2(\Theta) \subset H^1(\Omega) \times L^2(\Theta). \tag{21}$$

Similarly to the previous section, we define the stochastic Schur complement system for the random values at the internal boundary nodes as

$$[\mathbf{S}](\theta)\mathbf{u}_\Gamma(\theta) = \mathbf{b}_\mathrm{S}(\theta), \tag{22}$$

where the stochastic Schur complement matrix $[\mathbf{S}]$ is derived from the stochastic FE matrix with entries

$$[\mathbf{A}]_{l,l'}(\theta) = \int_\Omega \boldsymbol{\kappa}(x,\theta)\nabla\Phi_l(x)\cdot\nabla\Phi_{l'}(x)dx \quad \forall\, l,l' \in \mathcal{N}. \tag{23}$$

Following [34], the Schur complement system can be written as the sum of local contributions from individual subdomains:

$$[\mathbf{S}](\theta) = \sum_{d=1}^{D} [\mathrm{R}^{(d)}][\mathbf{S}]^{(d)}(\theta)[\mathrm{R}^{(d)}]^\top, \tag{24}$$

where $[\mathbf{S}]^{(d)} \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma^{(d)}}$ denotes the stochastic *influence matrix* of subdomain $\Omega^{(d)}$, and $[\mathrm{R}^{(d)}] \in \mathbb{R}^{\mathrm{N}_\Gamma \times \mathrm{N}_\Gamma^{(d)}}$ is the so-called restriction operator, which is a deterministic matrix that maps local boundary nodes of $\Omega^{(d)}$ to global internal boundary nodes. The influence matrix $[\mathbf{S}]^{(d)}$ is the boundary-to-boundary operator of the local stochastic problem, see [34] for more details. The interest in the representation of the stochastic Schur matrix in (24) stems from the fact that the influence matrices $[\mathbf{S}]^{(d)}$ depend on the stochastic field $\boldsymbol{\kappa}$ over their respective subdomains $\Omega^{(d)}$, only. This property is heavily exploited in the following to construct Polynomial Chaos (PC) surrogates of the local influence matrix.

3.3. PC Expansion of Local Operators

Let us denote by $\boldsymbol{\kappa}^{(d)}(x,\theta)$ the restriction of $\boldsymbol{\kappa}$ to $x \in \Omega^{(d)}$. Since $[\mathbf{S}]^{(d)}(\theta)$ is a function of $\boldsymbol{\kappa}^{(d)}$, we start by approximating $\boldsymbol{\kappa}^{(d)}$ using a finite dimensional parametrization. A natural approach is to rely on the local truncated KL expansion of the Gaussian process $\boldsymbol{G}^{(d)} = \log \boldsymbol{\kappa}^{(d)}$ over $\Omega^{(d)}$:

$$\boldsymbol{G}^{(d)}(x,\theta) \approx \widehat{\boldsymbol{G}}^{(d)}(x,\theta) \doteq \sum_{i=1}^{N_{\mathrm{KL}}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta), \quad x \in \Omega^{(d)}, \tag{25}$$

where $(\lambda_i^{(d)}, \hat{\phi}_i^{(d)}(x))$ are eigenpairs of the covariance function of $\boldsymbol{G}^{(d)}$, see Appendix A for more details. We recall that $\boldsymbol{G}$ being Gaussian, the random vector $\boldsymbol{\xi}^{(d)} \doteq \left(\boldsymbol{\xi}_1^{(d)}, \ldots, \boldsymbol{\xi}_{N_{\mathrm{KL}}^{(d)}}^{(d)}\right)$ has i.i.d. components, $\boldsymbol{\xi}_i^{(d)} \sim N(0,1)$. Further, we introduce the local approximation of $\boldsymbol{\kappa}$ as

$$\boldsymbol{\kappa}^{(d)}(x,\theta) \approx \hat{\boldsymbol{\kappa}}^{(d)}(x,\theta) \doteq \exp\left[\sum_{i=1}^{N_{\mathrm{KL}}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta)\right], \tag{26}$$

and we denote by $[\widehat{\mathbf{S}}]^{(d)}(\theta)$ the stochastic influence matrix of the subdomain based on $\hat{\boldsymbol{\kappa}}^{(d)}$. Clearly, the KL truncation to the $N_{\mathrm{KL}}^{(d)}$ dominant modes of the coefficient will affect the error in the approximation of the influence matrix: the larger $N_{\mathrm{KL}}^{(d)}$, the closer $[\mathbf{S}]^{(d)}$ and $[\widehat{\mathbf{S}}]^{(d)}$. In fact, as illustrated in Section 4, $N_{\mathrm{KL}}^{(d)}$ controls the trade-off between the effectiveness of the stochastic preconditioner and the complexity of its construction through the approximation of the influence matrices. For instance, using $N_{\mathrm{KL}}^{(d)} = 0$ for all subdomains results in the deterministic preconditioner $\overline{[\mathrm{S}]}$.

For $N_{\mathrm{KL}}^{(d)} \geq 1$, we now have to approximate the dependencies of $[\widehat{\mathbf{S}}]^{(d)}$ on the vector of $N_{\mathrm{KL}}^{(d)}$ independent standard Gaussian random variables $\boldsymbol{\xi}_i^{(d)}$, with joint probability density function $p_{\boldsymbol{\xi}^{(d)}}$. For simplicity, we drop the subdomain index (d) temporarily. We introduce the weighted Hilbert space $L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{\mathrm{KL}}})$ defined by

$$\boldsymbol{f} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{\mathrm{KL}}}) \iff \int_{\mathbb{R}^{N_{\mathrm{KL}}}} |\boldsymbol{f}(\mathrm{y})|^2 \, p_{\boldsymbol{\xi}}(\mathrm{y}) \, d\mathrm{y} < \infty,$$

for any function $\boldsymbol{f} : \mathbb{R}^{N_{\mathrm{KL}}} \to \mathbb{R}$, equipped with the (weighted) inner product and associated norm

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\boldsymbol{\xi}} = \int_{\mathbb{R}^{N_{\mathrm{KL}}}} \boldsymbol{f}(\mathrm{y}) \boldsymbol{g}(\mathrm{y}) p_{\boldsymbol{\xi}}(\mathrm{y}) \, d\mathrm{y}, \quad \|\boldsymbol{f}\|_{\boldsymbol{\xi}} = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\boldsymbol{\xi}}^{1/2}.$$

We remark that for any $\boldsymbol{f}, \boldsymbol{g} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{\mathrm{KL}}})$, $\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\boldsymbol{\xi}} = \langle \boldsymbol{f}(\boldsymbol{\xi}), \boldsymbol{g}(\boldsymbol{\xi}) \rangle_{\Theta}$, so that $\boldsymbol{f} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{\mathrm{KL}}}) \iff \boldsymbol{f}(\boldsymbol{\xi}) \in L^2(\Theta)$. Following [44, 45], we introduce the Polynomial Chaos basis of $L^2_{\boldsymbol{\xi}}(N_{\mathrm{KL}})$. Since the random variables $\boldsymbol{\xi}_i$ are independent and follow the standard Gaussian distribution, the PC basis consists in the infinite set of orthonormal Hermite polynomials $\boldsymbol{\Psi}_{\alpha}^{N_{\mathrm{KL}}}(\boldsymbol{\xi})$. The multi-variate Hermite polynomials are defined as products of univariate orthonormal Hermite polynomials, through

$$\boldsymbol{\Psi}_{\alpha}^{N_{\mathrm{KL}}}(\boldsymbol{\xi}) = \prod_{j=1}^{N_{\mathrm{KL}}} \boldsymbol{\varphi}_{\alpha_j}(\boldsymbol{\xi}_j), \tag{27}$$

where $\alpha = (\alpha_1 \cdots \alpha_{N_{\mathrm{KL}}}) \in \mathbb{N}^{N_{\mathrm{KL}}}$ is a multi-index and $\boldsymbol{\varphi}_j$ is the univariate Hermite polynomial of degree $\alpha_j \in \mathbb{N}$. Any function $\boldsymbol{f} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{\mathrm{KL}}})$ has a PC expansion [44, 45] of the form

$$\boldsymbol{f}(\boldsymbol{\xi}) = \sum_{\alpha \in \mathbb{N}^{N_{\mathrm{KL}}}} f_{\alpha} \boldsymbol{\Psi}_{\alpha}^{N_{\mathrm{KL}}}(\boldsymbol{\xi}). \tag{28}$$

In practice the PC expansion (28) must be finite, and a truncation of the series (28) is needed. The truncation is usually performed by prescribing a polynomial degree $p \geq 0$ to define a finite set of multi-indices $\mathcal{B}$ in the summation. In this work, we shall consider the following truncations strategies

- partial-degree:
$$\mathcal{B} \doteq \left\{ \alpha \in \mathbb{N}^{N_{\mathrm{KL}}} : \max_{1 \leq j \leq N_{\mathrm{KL}}} \{\alpha_j\} = \|\alpha\|_{\infty} \leq p \right\};$$
- total-degree:
$$\mathcal{B} \doteq \left\{ \alpha \in \mathbb{N}^{N_{\mathrm{KL}}} : \sum_{j=1}^{N_{\mathrm{KL}}} \alpha_j = \|\alpha\|_{\ell_1} \leq p \right\};$$
- hyperbolic-cross:
$$\mathcal{B} \doteq \left\{ \alpha : \prod_{j=1}^{N_{\mathrm{KL}}} (\alpha_j + 1) \leq p + 1 \right\}.$$

Then, the truncated PC expansion of $\boldsymbol{f}$,

$$\boldsymbol{f}(\boldsymbol{\xi}) \approx \sum_{\alpha \in \mathcal{B}} f_\alpha \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}}(\boldsymbol{\xi}),$$

has a finite number of terms $J = |\mathcal{B}|$. For a fixed p, the hyperbolic-cross truncation gives the smallest PC basis, while the partial-degree truncation gives the largest one with $J = (p+1)^{N_{\mathrm{KL}}}$.

Several approaches are possible to estimate the PC coefficients f_α. A stochastic Galerkin method was employed to compute the PC coefficients of the stochastic influence matrices in [34]. In the present work, we rely on a more versatile Non-Intrusive (NI) approach, which uses a quadrature method to determine the PC coefficients. We motivate this choice by the subsequent developments of Section 3.4, which are readily amenable to generic NI approaches. Thanks to the orthonormality of the PC basis, $\langle \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}}, \boldsymbol{\Psi}_\beta^{N_{\mathrm{KL}}} \rangle_{\boldsymbol{\xi}} = \delta_{\alpha,\beta}$, the coefficient f_α is given by

$$f_\alpha = \langle \boldsymbol{f}, \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}} \rangle_{\boldsymbol{\xi}} = \int_{\mathbb{R}^{N_{\mathrm{KL}}}} \boldsymbol{f}(\mathrm{y}) \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}}(\mathrm{y}) p_{\boldsymbol{\xi}}(\mathrm{y}) d\mathrm{y}. \tag{29}$$

In our NI implementation, the integral in (29) is simply discretized by means of a N_{KL}-dimensional quadrature formula,

$$f_\alpha \approx \sum_{q=1}^{N_{\mathrm{Q}}} \boldsymbol{f}(\mathrm{y}_q) \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}}(\mathrm{y}_q) w_q, \tag{30}$$

where $\mathrm{y}_q \in \mathbb{R}^{N_{\mathrm{KL}}}$ and $w_q \in \mathbb{R}$ are the quadrature nodes and weights of the formula. Without loss of generality, we employed tensored Gauss quadrature formulas of sufficiently high degree to ensure the discrete orthonormality of the basis polynomials:

$$\sum_{q=1}^{N_{\mathrm{Q}}} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}}(\mathrm{y}_q) \boldsymbol{\Psi}_\beta^{N_{\mathrm{KL}}}(\mathrm{y}_q) w_q = \delta_{\alpha,\beta}, \quad \forall \alpha, \beta \in \mathcal{B}.$$

This characteristic guaranties an estimation of the PC coefficients free of internal aliasing. Also, the complexity of the NI projection directly relates to the number of quadrature nodes N_{Q} which increases exponentially fast with both the number of local random variables N_{KL} and the maximum polynomial degree p.

Returning to the approximation of $\left[\widehat{\mathbf{S}}\right]^{(d)}$, and reintroducing the subdomain index, we assume that all its entries are in $L^2(N_{\mathrm{KL}}^{(d)})$; it comes

$$\left[\widetilde{\mathbf{S}}\right]^{(d)}(\theta) = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{S}]_\alpha^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}} \left(\boldsymbol{\xi}^{(d)}(\theta) \right), \tag{31}$$

with

$$[\mathrm{S}]_\alpha^{(d)} = \sum_{q=1}^{N_{\mathrm{Q}}^{(d)}} \left[\widehat{\mathrm{S}}\right]_q^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}}(\mathrm{y}_q^{(d)}) w_q^{(d)}, \tag{32}$$

and where $\left[\widehat{\mathrm{S}}\right]_q^{(d)}$ is the realization of the influence matrix for the realization of $\hat{\boldsymbol{\kappa}}^{(d)}$ corresponding to $\boldsymbol{\xi}^{(d)} = \mathrm{y}_q$ in (26). In the following, we restrict ourselves to a uniform PC order p for all subdomains, while the number of local random variables $N_{\mathrm{KL}}^{(d)}$ will be fixed or adapted for each subdomain depending on the numerical experiments. Finally, the formal expression of the stochastic preconditioner is

$$\left[\widetilde{\mathbf{S}}\right](\theta) \doteq \sum_{d=1}^{D} [\mathrm{R}]^{(d)} \left[\widetilde{\mathbf{S}}\right]^{(d)}(\theta) \left([\mathrm{R}]^{(d)} \right)^\top. \tag{33}$$

For each realization $\kappa^{(m)}(x)$, the corresponding preconditioner is obtained using (33), where the constitutive influence matrices $\left[\widetilde{\mathbf{S}}\right]^{(d)}(\theta)$ are evaluated using (31). The random variables $\boldsymbol{\xi}^{(d)}(\theta^{(m)})$ are computed

by projecting $\log \kappa^{(m)}$ on the local KL modes (see Appendix A). Denoting by $\phi_i^{(d)}$ the extension of $\hat{\phi}_i^{(d)}$ to Ω with compact support in $\overline{\Omega^{(d)}}$, and observing that the $\phi_i^{(d)}$ form an orthonormal system, it comes

$$\boldsymbol{\xi}_i^{(d)}(\theta^{(m)}) = \frac{1}{\sqrt{\lambda_i^{(d)}}} \int_\Omega \log\left(\kappa^{(m)}(x)\right) \phi_i^{(d)}(x)\, dx, \quad \forall i = 1, \dots, N_{\mathrm{KL}}^{(d)} \text{ and } d = 1, \dots, D. \tag{34}$$

In practice, the integrals in (34) are numerically approximated using the quadrature rule employed to discretize the (local) KL eigenvalue problem (A.3). In this paper, we use element-wise constant quadrature to estimate (A.3).

In the rest of the paper, we call the preconditioner defined by (33) and (31) the Direct PC (DPC) preconditioner, and the PCG method using this preconditioner the Direct Preconditioned CG (DPCG) method.

As illustrated later, one issue of the DPC preconditioner is that it is not guaranteed to be SPD for all samples of $\boldsymbol{\kappa}$, unless the polynomial degree is large enough. For problems with a large variance of $\boldsymbol{\kappa}$, the stochastic influence matrices have positive eigenvalues that can vary over a large range and become close to zero (note that interior subdomains have only Positive *Semi* Definite influence matrices, with a.s. a zero eigenvalue corresponding to constant boundary values). The PC approximation of eigenpairs getting close to zero is challenging because of the oscillatory character of the polynomials that tends to induce spurious negative eigenvalues in $[\widetilde{\mathbf{S}}]^{(d)}$. In these challenging situations, having p large enough to guaranty with high enough probability the positivity of the stochastic preconditioner can be prohibitively expensive, and a more robust approach is in order. We propose to proceed with an appropriate factorized PC representation.

3.4. Factorization of local stochastic operators

The direct projection of the local influence operators can not always ensure for all samples the positivity (or semi-definiteness) of the PC approximation $[\widetilde{\mathbf{S}}]^{(d)}$. To remedy this issue, we propose a PC approach based on a factorization of $[\widehat{\mathbf{S}}]^{(d)}$ before the projection. For simplicity of the exposition, in the rest of the subsection, we do not make explicit the operators' dependencies on the random event.

3.4.1. Cholesky-type factorizations

As the local influence operator $[\widehat{\mathbf{S}}]^{(d)}$ is symmetric and Positive Semi Definite, we start with its rank revealing Cholesky decomposition,

$$[\widehat{\mathbf{S}}]^{(d)} = [\mathbf{L}][\mathbf{D}][\mathbf{L}]^\top, \tag{35}$$

where $[\mathbf{L}]$ is a lower unit triangular matrix and $[\mathbf{D}]$ is a non-negative diagonal matrix. From this decomposition, we define the first factorization of $[\widehat{\mathbf{S}}]^{(d)}$ as

$$[\widehat{\mathbf{S}}]^{(d)} = [\mathbf{H}]^{(d)}{[\mathbf{H}]^{(d)}}^\top, \quad [\mathbf{H}]^{(d)} = [\mathbf{L}][\Delta], \tag{36}$$

where $[\Delta] \doteq [\mathbf{D}]^{\frac{1}{2}}$ uses the non-negative squareroots of the entries of $[\mathbf{D}]$.

One could think of constructing the PC expansion $[\widetilde{\mathbf{H}}]^{(d)}$ of the stochastic factor $[\mathbf{H}]^{(d)}$ using the NI projection method introduced before. Then, using the product of this PC expansion with its transpose, following (36), we would approximate $[\widehat{\mathbf{S}}]^{(d)}$ by

$$[\widetilde{\mathbf{S}}]^{(d)} = [\widetilde{\mathbf{H}}]^{(d)}{[\widetilde{\mathbf{H}}]^{(d)}}^\top, \tag{37}$$

which would be almost surely non-negative for all PC basis. Unfortunately, the convergence with p of the PC approximation in (37) to $[\widehat{\mathbf{S}}]^{(d)}$ can be compromised or even impossible in practice. The origin of the lack of convergence is the non-uniqueness of the Cholesky decomposition. As a result, it is delicate to define consistent deterministic Cholesky factors for all the quadrature nodes. As an example, if $[\mathbf{L}]$ is a stochastic

factor, then $\boldsymbol{\alpha}(\theta)[\mathbf{L}]$ is also a factor for *any* random variable $\boldsymbol{\alpha}$ taking value in $\{-1,+1\}$. Depending on the particular choice of $\boldsymbol{\alpha}$, the projection of $\boldsymbol{\alpha}(\theta)[\mathbf{L}][\Delta]$ may be extremely challenging. Without appropriate treatment, the factors evaluated at the nodes $\mathrm{y}_q^{(d)}$ can correspond to arbitrarily non-smooth $\boldsymbol{\alpha}$, which may compromise the PC convergence. This situation is similar to the problem faced in approximating parametric dependencies of stochastic operators eigenpairs [46].

3.4.2. Orthogonal factorization

An important observation is that the influence matrices are symmetric, and thus, admit an orthogonal factorization given by

$$[\mathbf{S}]^{(d)} = [\mathbf{Q}][\mathbf{D}][\mathbf{Q}]^\top \tag{38}$$

where $[\mathbf{D}]$ is a non-negative diagonal matrix with the eigenvalues of $[\mathbf{S}]^{(d)}$ and the columns of the orthogonal matrix $[\mathbf{Q}]$ are the stochastic eigenvectors of $[\mathbf{S}]^{(d)}$. Denoting again by $[\Delta]$ the diagonal matrix of (non-negative) squareroots of $[\mathbf{D}]$, the factor

$$[\mathbf{H}]^{(d)} = [\mathbf{Q}][\Delta], \tag{39}$$

leads to a valid decomposition $\widehat{[\mathbf{S}]}^{(d)} = [\mathbf{H}]^{(d)}[\mathbf{H}]^{(d)\top}$. However, as for the Cholesky decomposition, the eigenvectors are not uniquely defined, in particular when some eigenvalues have multiplicity larger than 1. Further, the ordering of the eigenmodes using the magnitude of the eigenvalues may not be stable when $\boldsymbol{\kappa}$ is sampled (crossing of eigenbranches [46]). However, we can easily overcome this issue by defining an alternative factorization as

$$\widehat{[\mathbf{S}]}^{(d)} = [\mathbf{H}]^{(d)}[\mathbf{H}]^{(d)}, \quad [\mathbf{H}]^{(d)} \doteq [\mathbf{Q}][\Delta][\mathbf{Q}]^\top. \tag{40}$$

With this definition, $[\mathbf{H}]^{(d)}$ is invariant to the particular choice of eigenvectors, and therefore possesses a convergent PC expansion. This PC expansion is written as

$$\widetilde{[\mathbf{H}]}^{(d)}(\theta) = \sum_{\alpha\in\mathcal{B}^{(d)}} [\mathrm{H}]_\alpha^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}}(\boldsymbol{\xi}^{(d)}), \tag{41}$$

where the PC coefficients $[\mathrm{H}]_\alpha^{(d)}$ are computed by quadrature, using

$$[\mathrm{H}]_\alpha^{(d)} = \sum_{q=1}^{N_{\mathrm{Q}}^{(d)}} [\mathrm{H}]_q^{(d)} \boldsymbol{\Psi}_\alpha^{(d)}(\mathrm{y}_q) w_q^{(d)}. \tag{42}$$

In (42), the factors $[\mathrm{H}]_q^{(d)}$ of the quadrature nodes are defined as

$$[\mathrm{H}]_q^{(d)} = [\mathrm{Q}]_q[\Delta]_q[\mathrm{Q}]_q^\top, \tag{43}$$

where $[\mathrm{Q}]_q$ and $[\Delta]_q$ are obtained from the decomposition of $\widehat{[\mathrm{S}]}_q^{(d)}$ defined above for the direct projection. As a consequence, the overhead of the factorized approach, compared to the direct one, amounts to the factorization of the deterministic influence matrices at all quadrature nodes of all subdomains. In practice, the cost of these factorizations is only a fraction of the cost of computing $\widehat{[\mathrm{S}]}_q^{(d)}$. Algorithm 1 summarizes the procedure to obtain the PC approximation of the factor for a given subdomain. Since the subdomains share no information, the computation of the local PC expansions is possible in parallel.

Finally, the PC approximation of the influence operator reads

$$\widetilde{[\mathbf{S}]}^{(d)}(\theta) = \left(\widetilde{[\mathbf{H}]}^{(d)}(\theta)\right)^2 \doteq \left(\sum_{\alpha\in\mathcal{B}^{(d)}} [\mathrm{H}]_\alpha^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}}\left(\boldsymbol{\xi}^{(d)}(\theta)\right)\right)^2, \tag{44}$$

Algorithm 1 Set PC expansion $[\widetilde{\mathbf{H}}]^{(d)}$

procedure COMPUTE-$[\widetilde{\mathbf{H}}]^{(d)}$(KL decomposition of $\boldsymbol{\kappa}^{(d)}$, PC basis)
  Set quadrature nodes and weights;
  for all PC modes α **do**
    set $[\mathrm{H}]_\alpha^{(d)} = [0]$; ▷ Initialization of the PC modes
  end for
  for $q = 1, \ldots, N_{\mathrm{Q}}^{(d)}$ **do** ▷ Loop over quadrature nodes
    Set $\hat{\kappa}^{(d)}$ for $\boldsymbol{\xi}^{(d)} = \mathrm{y}_q$; ▷ Set coefficient, see (26)
    Compute $[\widehat{\mathrm{S}}]_q^{(d)}$; ▷ Set the influence matrix
    Solve $[\widehat{\mathrm{S}}]_q^{(d)} = [\mathrm{Q}][\mathrm{D}][\mathrm{Q}]^\top$; ▷ Decompose the influence matrix
    Set $[\mathrm{H}]_q^{(d)} = [\mathrm{Q}][\Delta][\mathrm{Q}]^\top$; ▷ Set the factor, see (43)
    for all PC mode α **do**
      $[\mathrm{H}]_\alpha^{(d)} \leftarrow [\mathrm{H}]_\alpha^{(d)} + [\mathrm{H}]_q^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}}(\mathrm{y}_q) w_q$; ▷ Update PC modes, see (42)
    end for
  end for
  return $\{[\mathrm{H}]_\alpha^{(d)}\}$; ▷ Return the PC modes
end procedure

where $\boldsymbol{\xi}^{(d)}(\theta)$ is given by (34), while the corresponding preconditioner is expressed as

$$[\widetilde{\mathbf{S}}](\theta) = \sum_{d=1}^{D} [\mathrm{R}]^{(d)} \left(\sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}]_\alpha^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}} \left(\boldsymbol{\xi}^{(d)}(\theta) \right) \right)^2 \left([\mathrm{R}]^{(d)} \right)^\top . \tag{45}$$

Hereafter, we call the preconditioner in (45) the Factorized PC (FPC) preconditioner and the corresponding CG method the Factorized PCG (FPCG) method.

3.5. Sampling and Preconditioning

Whence the PC expansions of the local operators $[\widetilde{\mathbf{H}}]^{(d)}(\theta)$ constituting the stochastic preconditioner have been set for all subdomains, in a preprocessing stage, the sampling stage can start. The sampling procedure involves, for each sample, $\kappa^{(m)}(x)$, two main steps: the set-up and the resolution. In the set-up step, one goes through the subdomains to a) construct the local operator of the sampled elliptic problem (12), b) compute the projection on the local KL basis (34) to get the realization of the local random variables $\boldsymbol{\xi}^{(d)}$, and c) use these values to evaluate the PC surrogate of the influence operators from (44). We observe that tasks a) to c) are independent and involve no exchange of information between the subdomains, allowing for straightforward parallelization strategies. After completion of the first step, the local influence operators of the subdomains can be assembled to form the preconditioner of the realization, denoted by $[\widetilde{\mathrm{S}}]^{(m)}$, following (45), and the resolution step is engaged. The Schur system (19) corresponding to the coefficient $\kappa^{(m)}$ is solved iteratively, in a matrix-free approach, with the PCG algorithm and using the preconditioner $[\widetilde{\mathrm{S}}]^{(m)}$. Algorithm 2 summarizes the workflow for the resolution of one sample.

Algorithm 2 involves a procedure PCG (see line 10) that solves the reduced problem with the FPCG. It returns the solution u_Γ^k satisfying the tolerance criterion specified by the argument tol. Within the iterations, the PCG algorithm updates the solution, residual, and conjugated directions (see for instance [35]), until the convergence criterion is met, that is when $\|\mathrm{r}^k\|/\|\mathrm{b}_{\mathrm{S}}\| < \mathrm{tol}$. Algorithm 2 does not show the computation of the system's right-hand-side b_{S}; this computation involves local solves, following (19), and is performed in the initial loop over the subdomains in parallel with the evaluation of $[\widetilde{\mathrm{S}}]$. Each iteration requires the application of the Schur operator and the resolution of a preconditioning problem (computation of $[\widetilde{\mathrm{S}}]^{-1}\mathrm{r}^k$). The Schur operator is applied in a matrix-free approach, leading to the resolution of local problems, possibly in parallel.

Algorithm 2 Procedure to compute one solution sample with the FPCG method

1: **procedure** FPCG-SOLVE(Sample $\kappa^{(m)}$, tolerance tol, initial guess u^0)
2: Set $[\widetilde{\mathrm{S}}] = [0]$; ▷ Initialize Preconditioner
3: **for** $d = 1, \ldots, D$ **do** ▷ Loop over subdomains
4: Set local problem (12);
5: Set $\boldsymbol{\xi}^{(d)}$ by local projection; ▷ see (34)
6: Set $[\mathrm{H}^{(d)}] = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}]_\alpha^{(d)} \boldsymbol{\Psi}_\alpha^{N_{\mathrm{KL}}^{(d)}} \left(\boldsymbol{\xi}^{(d)}(\theta)\right)$; ▷ Realization of factor (41)
7: Set $[\widetilde{\mathrm{S}}] \leftarrow [\widetilde{\mathrm{S}}] + [\mathrm{R}]^{(d)} [\mathrm{H}^{(d)}] \left([\mathrm{R}]^{(d)}\right)^\top$; ▷ Update Preconditioner;
8: **end for**
9: Set $[\widetilde{\mathrm{S}}]^{-1}$ ▷ Inversion of $[\widetilde{\mathrm{S}}]$
10: Set $\mathrm{u}_\Gamma = \mathrm{PCG}(\mathrm{u}_\Gamma^0, [\widetilde{\mathrm{S}}]^{-1}, \mathrm{tol})$; ▷ Do PCG solve
11: **Return** u_Γ; ▷ Return solution
12: **end procedure**

The resolution of these local problems can rely on standard solvers for deterministic elliptic problems. For the spatial meshes and numbers of subdomains considered in this work, we were able to assemble the local operators and store their Cholesky factorization. Thus, the local problems at each iteration are solved by means of direct Cholesky factorization. However, if the local problems are too large, an iterative method can be used instead. Concerning the preconditioning problem, the PCG algorithm's implementation classically involves an initial factorization of the preconditioner for an efficient application during the iterations. Here, this step is made explicit in line 9 of Algorithm 2. To avoid confusion with the factorized form the local influence operators of the FPC preconditioner, we prefer to label this step the inversion of the preconditioner. In this work, we exploit the SPD nature of the FPC preconditioner to compute its Cholesky decomposition, rather than its inverse. Note that other preconditioners, *e.g.* the median-based $[\overline{\mathrm{S}}]$ and DPC preconditioner, can be substituted in the call to PCG in the algorithm. However, the median-based preconditioner does not need to be "inverted" for each sample, and an LU decomposition is applied in the case of the DPC preconditioner as its positivity is not guaranteed.

Algorithm 2 is called multiple times by the sampler that generates the sequence of realizations of the coefficient $\kappa^{(m)}(x)$ and treats the solution samples $\mathrm{u}^{(m)}$ to derive the QoI and estimate their statistics.

4. Numerical tests

In this section, we numerically investigate the performance of the different preconditioning strategies: median-based, DPC, and FPC. As a test problem, we considers the stochastic elliptic equation in a two-dimensional unit square $\Omega = [0,1] \times [0,1]$. Unless specified otherwise, the FE discretization uses 16,441 triangular elements, with similar diameters, supporting piecewise continuous quadratic approximation (standard P_2 elements). Note that other FE methods (*e.g.* P_1) can be used without affecting the conclusions of the numerical experiment, since the proposed stochastic preconditioner relies on PC approximations of the discrete Schur system.

The spatial discretization has 33,150 unknowns nodal values for the full problem. For the discretization of the coefficient $\boldsymbol{\kappa}$, we employ an element-wise constant approximation. The nominal partition of Ω has $D = 100$ subdomains, leading to a Schur system (22) with size $N_\Gamma = 3{,}389$. The left plot of Fig. 1 shows the reference FE mesh and its partition into subdomains; the right plot shows a realization of the log-normal field $\boldsymbol{\kappa}$ for a covariance with parameters $\gamma = 1.2$, $\sigma^2 = 1$ and $\ell_c = 0.05$.

We measure the performance of a preconditioner by the number of PCG iterations needed to achieve the solution of the Schur system within a prescribed tolerance (tol) on the residual divided by the system's right-hand-side. All numerical experiments presented in the paper use a fixed tolerance of $\mathrm{tol} = 10^{-8}$ on the relative residual norm. We shall consider the MPCG method (median-based) as the reference and define the preconditioner's acceleration $\boldsymbol{\rho}$ as the ratio of the number of iterations needed to converge from the same

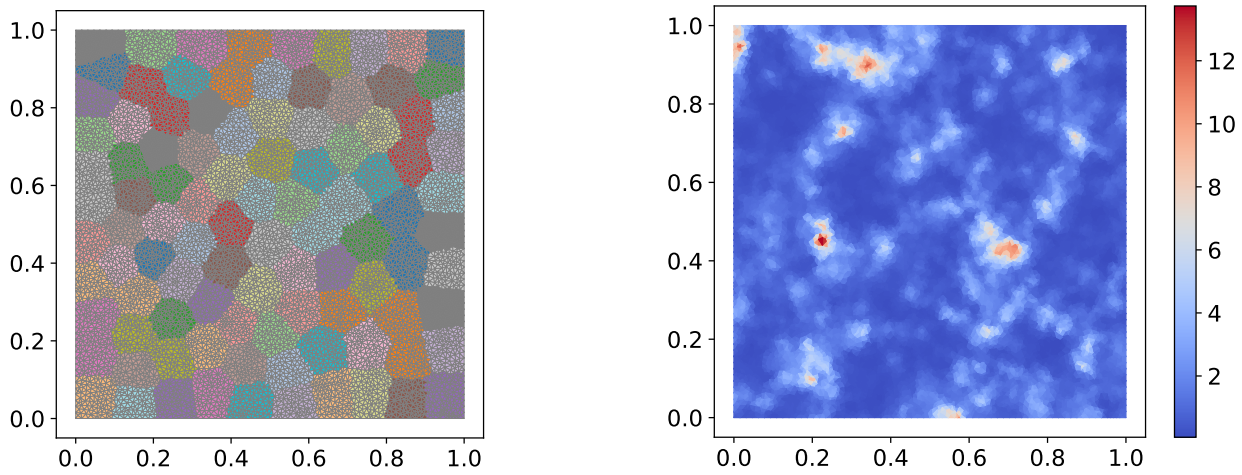


Figure 1: Finite Element mesh and partition of Ω in $D = 100$ subdomains (left), and a realization of $\boldsymbol{\kappa}$ for $\gamma = 1.2$, $\ell_c = 0.05$ and $\sigma^2 = 1$ (right).

initial guess $\mathbf{u}_\Gamma = 0$:

$$\boldsymbol{\rho} \doteq \frac{\#\text{ MPCG iterations}}{\#\text{ DPCG or FPCG iterations}}. \tag{46}$$

Since the number of iterations to converge depends on random samples of $\boldsymbol{\kappa}$, $\boldsymbol{\rho}$ is a random variable. A ratio greater than one means a higher efficiency relative to the median preconditioner.

4.1. MPCG method

We start by illustrating the degradation of the performance of the MPCG method when the median coefficient is not a good representative of all the samples.

Figure 2 reports the averaged number of MPCG iterations for a stochastic field $\boldsymbol{\kappa}$ with $\gamma = 1.2$ and different correlation lengths and variances of its log. The computations use a partition in $D = 100$ subdomains. A total of 1,000 samples are computed to estimate the averaged number of iterations to converge. For $\ell_c = 0.05$, we additionally represent the range of number of iterations using boxplots. Each box encompasses 50% of the samples and has a line at the median value. The whiskers cover 24.65% more samples each; finally, on each side, the 0.35% outliers are shown. This representation will be used consistently throughout the rest of the paper. For low variance values, the median field $\bar{\kappa}$ is representative of most realizations of $\boldsymbol{\kappa}$ and, on average, the median preconditioner achieves the solution in roughly 12 iterations; the sample variability is also low. When the variance increases, the sampled fields depart more and more from $\bar{\kappa}$, and the averaged number of MPCG iterations increases. This effect is more pronounced for short correlation length because the short-scale variations are then proportionally more significant such that the spatially-constant coefficient $\bar{\kappa}$ of the deterministic preconditioner is not representative and these situations are not properly handled. On the contrary, when $\ell_c \gg 1$ the sampled fields have small spatial variations, such that $\kappa^{(m)} \approx c\bar{\kappa}$ for some $c \in \mathbb{R}_+$, and the median preconditioner remains effective. Further, the number of iterations differs significantly from one sample to another when ℓ_c is small, as denoted by the significant extent of the whiskers and the spread of the outliers.

4.2. DPCG method

We now turn to the DPC preconditioner defined by equations (31)-(33). Compared to the median-based preconditioner, the DPC preconditioner allows for a better representation of the sampled fields, but it is not guaranteed to be SPD. In this section, we illustrate the behavior of the DPCG method, for a log-normal field with roughness $\gamma = 2$ (*i.e.* a smooth field), correlation length $\ell_c = 0.05$, and $D = 100$ subdomains. To

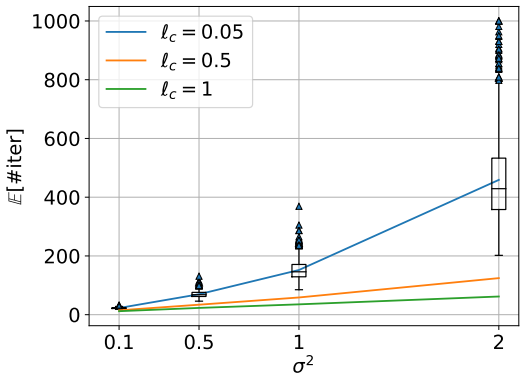


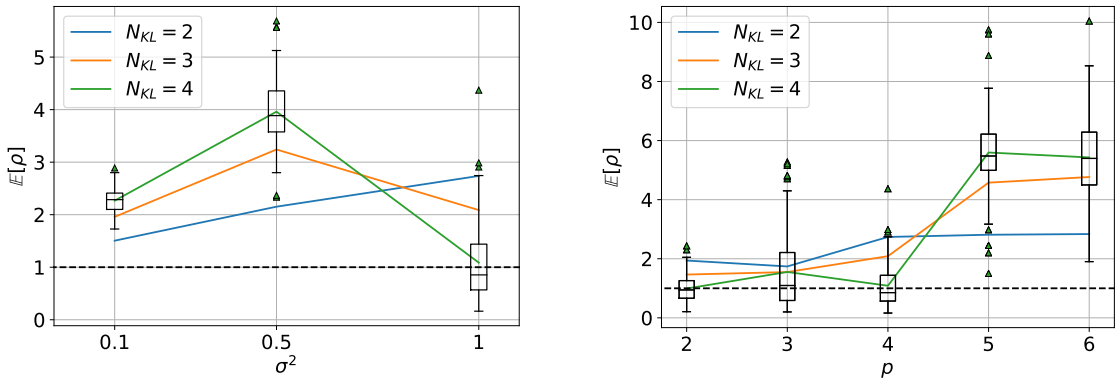
Figure 2: Average number of iterations to convergence (and corresponding boxplots for $\ell_c = 0.05$) in the MPCG method for different variances and correlation lengths. Case of $\gamma = 1.2$ with $D = 100$ subdomains.

simplify the analysis, the number of local random variables in the approximation of $\boldsymbol{\kappa}$ is fixed to N_{KL} for all subdomains.

Figure 3 reports the evolution of the acceleration $\boldsymbol{\rho}$ of the DPCG method for different variances σ^2, number N_{KL} of random variables, and truncation order p of the PC expansion with total-degree truncation. The average acceleration is estimated using 100 random samples of $\boldsymbol{\kappa}$.

In Fig. 3a, where $p = 4$, we observe that for a very low variance $\sigma^2 = 0.1$, the DPCG method needs roughly 1.5 to 2.5 times (depending on N_{KL}) fewer iterations to converge than the MPCG method. When the variance increases to $\sigma^2 = 0.5$, the average acceleration increases to roughly 2 to 4 times. However, when $\sigma^2 = 1$, the average acceleration decays to reach $\rho \approx 1$ for $N_{\mathrm{KL}} = 4$. The plot also shows that, for large σ^2, the acceleration deteriorates with N_{KL}. This behavior of the DPCG acceleration is explained as follows. When σ^2 is small, the influence matrices have limited variability, and their non-trivial eigen-pairs remain away from zero. As a result, they have accurate direct PC expansions for $p = 4$, ensuring samples of the DPC preconditioner are SPD with high probability. When σ^2 increases, the influence matrices have increasing variability and their lowest non-trivial eigenvalues get closer to zero with higher variability. Unless the polynomial degree is increased, the direct PC expansions of the influence matrices lose positivity because of the oscillatory character of the polynomial approximation. The loss of positivity adversely impacts the average acceleration. Anticipating some coclusions drawn from the next figure, we note that the number of non-SPD preconditioners is already increasing from $\sigma^2 = 0.1$ to $\sigma^2 = 0.5$. However, the number of non-SPD preconditioners occurring at $\sigma^2 = 0.5$ is still small. In addition, their negative eigenvalues have very small absolute value. This means that the high acceleration rates provided by the samples with SPD preconditioners can compensate the lower acceleration rates provided by the still few samples with non-SPD preconditioners. Therefore, the impact of the non-SPD preconditioners on the average acceleration rate is low, leading to a misleading increase of the average acceleration curve. The number of samples with non-SPD preconditioners is rapidly dominant as σ^2 increases. Increasing σ^2 also induces a larger sample variability of the acceleration (see whiskers of the boxplots provided for $N_{\mathrm{KL}} = 4$). The PC truncation error is found to become more critical when N_{KL} increases, suggesting that an accurate representation of joint effects between local modes of $\log \boldsymbol{\kappa}$ is crucial to maintain the acceleration level. The importance of the PC truncation error is further investigated in Fig. 3b which reports the average acceleration of the DPCG method for $\sigma^2 = 1$ and different PC orders. As expected, increasing p improves the average acceleration. However, the improvement is slow and demands using a large p when $\sigma^2 > 1$.

Figure 4 provides a more detailed spectral analysis of the DPC preconditioner. Figure 4a shows, for $N_{\mathrm{KL}} = 4$, the magnitude of the smallest negative eigenvalue $\lambda_{\min}$ in 100 samples of the preconditioner, using different degrees p and total order truncation. Out of the 100 samples, we have respectively $50, 90, 97, 17, 27$ DPC preconditioners with at least one negative eigenvalues when $p = 2, 3, 4, 5, 6$ respectively. In the range

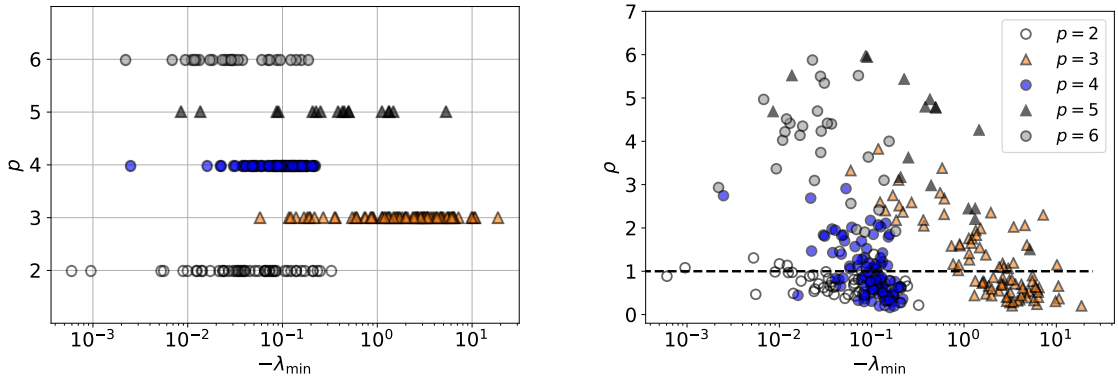


(a) Acceleration as a function of σ^2 ($p = 4$).

(b) Acceleration as a function of p ($\sigma^2 = 1$).

Figure 3: Average acceleration of the DPCG method (and corresponding boxplots for $N_{\mathrm{KL}} = 4$) for different variances σ^2, PC order p, and number of local random variables N_{KL}. Total-degree PC basis. $\gamma = 2$, $\ell_c = 0.05$ and $D = 100$.

of degrees tested, it is seen that the magnitude of the smallest negative eigenvalues is generally larger for even degrees. When the degree increases, the range of negative eigenvalues does not reduce much but their number (probability of occurrence) does reduce. Next, Fig. 4b shows the acceleration $\boldsymbol{\rho}$ of the DPCG method plotted against the lowest negative eigenvalues. A correlation between the magnitude of the lowest negative eigenvalue and the acceleration is visible when the PC degree p is odd. The trend is less clear for even degrees, but the plot indicates that the acceleration can be significantly degraded even when the smallest negative eigenvalues is not far from 0.



(a) Smallest negative eigenvalues for different PC orders.

(b) Acceleration vs. smallest negative eigenvalue.

Figure 4: Spectral analysis of 100 DPC preconditioners for total-order truncation, with $N_{\mathrm{KL}} = 4$.

To better understand the role of the PC truncation error, we compare in Fig. 5 the average acceleration of the DPC method for the partial degree, total degree and hyperbolic-cross truncations of the local PC bases. For a fair comparison, the average acceleration is reported as a function of local basis dimension $J^{(d)}$. The results correspond to $N_{\mathrm{KL}} = 3$ and the previous stochastic field with $\sigma^2 = 1$, $\gamma = 2$ and $\ell_c = 0.05$. It is seen that all truncation methods seem to converge to the same averaged acceleration, $\mathbb{E}[\boldsymbol{\rho}] = 5$, although at different rates. Specifically, the hyperbolic-cross truncation seems the least effective, while the total order truncation exhibits a non-monotonous behavior with odd/even degree effects, similar to the non-monotonous convergence reported in [34]. For comparable local basis dimensions $J^{(d)}$, the acceleration of the hyperbolic-

cross truncation is clearly less than for the two other truncation methods, indicating the importance of the interaction terms compared to univariate effects. Indeed, for a similar basis dimension, the hyperbolic-cross truncation incorporate much higher univariate degree polynomials, at the expense of multivariate polynomials. For instance, in Fig. 5, the hyperbolic-cross truncation goes up to $p = 20$ while the for the partial degree truncation it is limited to $p = 5$. Thus, the reported accelerations illustrate the inherent lack of robustness of the DPC method which, to ensure the positivity of the preconditioner, requires an accurate representation of most interactions between local KL modes. Consequently, aggressive truncations strategies (*e.g.* hyperbolic-cross), which typically disregard high-order interactions, are not suitable. This fact makes the DPCG method computationally demanding to achieve all the potential of the local stochastic approximation of $\boldsymbol{\kappa}$. Rather than tailoring PC bases to ensure correct DPC behavior, it is preferable to preserve the flexibility of arbitrary PC truncation strategies and to construct almost surely SPD stochastic preconditioners.

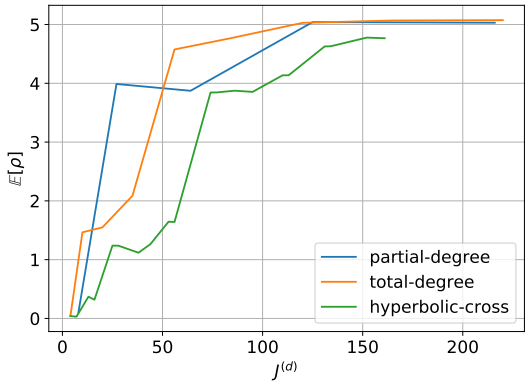


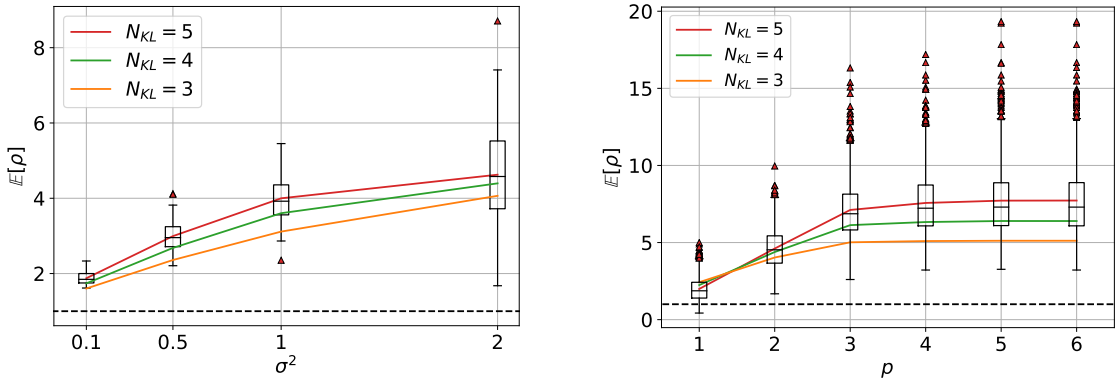
Figure 5: Average acceleration of the DPCG method as a function of the local PC bases dimension $J^{(d)}$ and for different PC truncations as indicated. Case of $N_{\mathrm{KL}} = 3$ and $\sigma^2 = 1$, $\gamma = 2$ and $\ell_c = 0.05$.

4.3. FPCG method

We now consider the FPCG method. We set $D = 100$, $\gamma = 1.2$ and $\ell_c = 0.05$. Note that the value of γ is less than in the previous section, so the problem is more demanding.

Figure 6 reports the acceleration of the FPCG method for different variances σ^2, and local discretization parameters N_{KL} and p. Figure 6a shows the effect of the variance σ^2 on the acceleration for total order truncation with $p = 2$. A significant improvement of the acceleration with σ^2 is reported, together with an increase of the sample variability. The PC degree has been halved and the range of σ^2 doubled compared to the case shown in Fig. 3a. The average acceleration of the FPCG method remains grater than 3 for $\sigma^2 > 1$, and is much less significantly impacted compared to the DPCG case. Figure 6b confirms that increasing p improves the acceleration, until the local KL truncation error on $\log \boldsymbol{\kappa}$ becomes dominant and prevents further improvement of the acceleration. In addition, Figure 6b shows that the spread of the acceleration remains finite when the PC error is negligible, confirming that the sample variability of $\boldsymbol{\rho}$ is mostly controlled by N_{KL}, and not p. We finally remark that, in our experiments, the FPCG method always yields an acceleration $\boldsymbol{\rho} > 1$, meaning that the FPCG method always does better than the MPCG method, even for low orders $p = 1$. Note that the case $p = 0$ formally corresponds to a deterministic preconditioning with the mean of the Schur system; it does not exactly coincide with the MPCG method that uses the Schur system associated to the median field, but the two methods are expected to achieve the same performance ($\boldsymbol{\rho} = 1$).

To complete the comparison with the DPCG method, Fig. 7 reports the average acceleration as a function of the local PC basis dimension using the total degree and hyperbolic-cross truncations with different degrees p and fixed $N_{\mathrm{KL}} = 3$. First, the FPCG acceleration is seen to achieve the asymptotic acceleration for much



(a) Acceleration as a function of σ^2 ($p = 2$).

(b) Acceleration as a function of p ($\sigma^2 = 2$).

Figure 6: Average acceleration of the FPCG method, with corresponding boxplots for $N_{\text{KL}} = 5$. Case of $\gamma = 1.2$, $\ell_c = 0.05$, $D = 100$.

lower dimensional bases (degree) compared to the case of DPCG method shown in Fig. 5. This much more satisfying behavior is attributed to the built-in characteristic of the FPC preconditioner that does not consume PC degrees to ensure positivity. Further, the acceleration of the FPCG method is much less sensitive to the truncation method, therefore enabling alternative bases construction and offering flexibility in the PC approximation method. In the present work, the PC expansion being determined using fixed isotropic quadrature rules, the results presented in the rest of the paper will be based on the total degree truncation. However, more advanced approximation techniques (*e.g.* sparse approximation, low rank approximation, ...) can be considered with the FPCG method, now that the positivity issues are resolved.

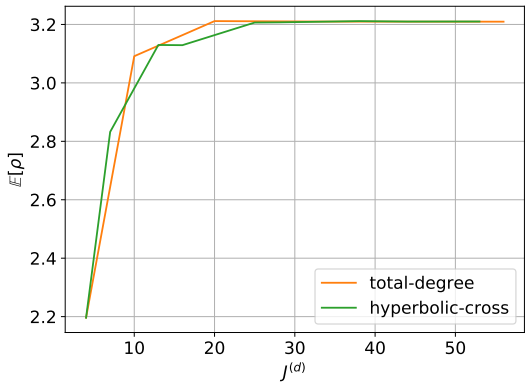
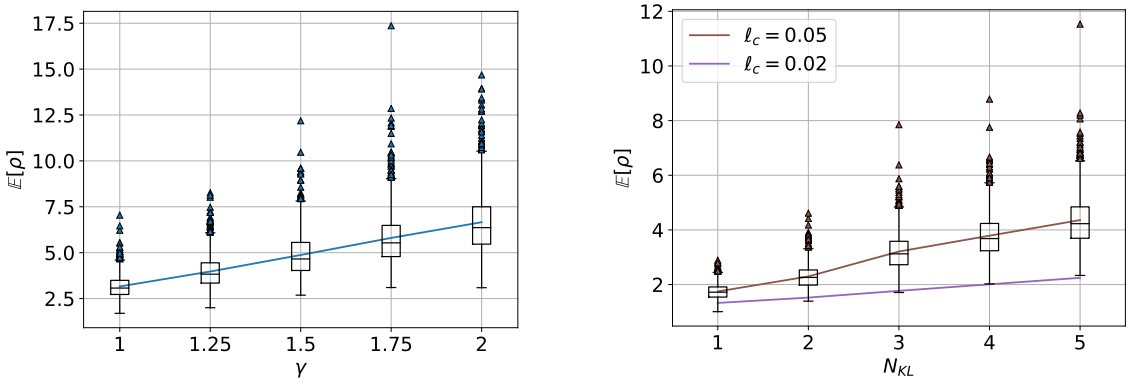


Figure 7: Average acceleration of the FPCG method as a function of the local basis dimension $J^{(d)}$ and hyperbolic-cross and total degree PC truncations. Case of $N_{\text{KL}} = 3$ and $\sigma^2 = 1$.

The analysis of the FPCG method continues with Fig. 8a, which shows the dependence of the FPCG acceleration on the roughness of the log-normal fields. The variance and correlation length are fixed to $\sigma^2 = 1$ and $\ell_c = 0.05$ while the local KL dimension is set to $N_{\text{KL}} = 4$ and the PC order is $p = 4$ (total degree truncation). The plot indicates that the acceleration improves as the field becomes smoother (*i.e.* γ increases). This behavior is expected since, for fixed N_{KL} and variance σ^2, the (local) KL truncation error reduces for increasing γ (see Appendix A). To further appreciate the effect of the KL truncation error, Fig. 8b shows the acceleration of increasing N_{KL} when $\gamma = 1.2$ and the variance as in Fig. 8a. Cases of $\ell_c = 0.02$ and $\ell_c = 0.05$ are reported. Consistently with the behavior of the KL truncation error, the FPCG

acceleration improves with N_{KL} for the two correlation lengths, and the acceleration is the largest for the largest ℓ_c. In addition, the gap between the accelerations for $\ell_c = 0.02$ and $\ell_c = 0.05$ increases with N_{KL}, reflecting the higher convergence rate of the local KL expansion for the largest ℓ_c (see Appendix A). Also, for $\ell_c = 0.05$, the improvement of the acceleration seems to slow down for the largest tested values of N_{KL}; this is explained by the emergence of the PC truncation error contribution which becomes more noticeable as the KL truncation error reduces. These experiments confirm that the efficiency of the FPCG methods improves with the accuracy of the local approximation of the stochastic coefficient $\boldsymbol{\kappa}$, controlled by N_{KL}, and of the PC expansion of the influence operators' factor, controlled by p. These two parameters of the FPC preconditioner should be selected jointly to balance the KL and PC truncation errors. In any case, one key feature of the FPCG method is that the preconditioner remains effective and achieves a significant acceleration even for low values of p and N_{KL}.



(a) Acceleration with γ ($N_{\mathrm{KL}} = 4, \ell_c = 0.05$).

(b) Acceleration with N_{KL} ($\gamma = 1.2$).

Figure 8: Average acceleration of the FPCG method (and corresponding boxplots for $\ell_c = 0.05$) with the roughness parameter γ (left) and local KL truncation N_{KL} (right). Other parameters are $\sigma^2 = 1$ and $p = 4$.

4.4. Influence of the number of subdomains

4.4.1. Fixed number of local KL modes

The previous experiments have demonstrated that increasing N_{KL} provides a higher acceleration of the FPCG method, compared to the reference MPCG method, by improving the local representation of $\boldsymbol{\kappa}$ over the subdomains. However, the computation cost and the memory requirement to store the preconditioner's factors increase quickly with both the PC degree p and number N_{KL} of local random variables. Therefore one cannot consider arbitrarily large values for N_{KL}. As explained in Appendix A, the convergence rate of the KL expansions depends on the covariance function, through its parameters σ^2, γ, and ℓ_c. For fixed covariance parameters, the convergence rate of the local KL expansion over a subdomain depends in fact on the *apparent correlation length* over the subdomain, $\ell_{\mathrm{loc}} \doteq \ell_c / \operatorname{diam}(\Omega^{(d)})$, where $\operatorname{diam}(\Omega^{(d)})$ is the diameter of $\Omega^{(d)}$. Therefore, considering smaller subdomains with the same value of N_{KL} results in lower local KL error. In the case of sub-domains with balanced sizes, their diameters will be $\operatorname{diam}(\Omega^{(d)}) \sim D^{1/n}$ such that $\ell_{\mathrm{loc}} \sim \mathcal{O}(D^{-1/n})$ (recall that n is the number of spatial dimensions).

Figure 9 illustrates the improvement of the acceleration achieved when increasing D, keeping all other parameters fixed. This numerical experiment uses a stochastic coefficient $\boldsymbol{\kappa}$ with $\sigma^2 = 1$, $\gamma = 1.2$ and $\ell_c = 0.05$ (left plot) and $\ell_c = 0.02$ (right plot). The numerical parameters of the FPCG methods are $N_{\mathrm{KL}} = 4$ and $p = 4$. It is seen that, as expected, the average acceleration improves with D, even though the samples variability of $\boldsymbol{\rho}$ increases too, as denoted by the extents of the whiskers. However, the whiskers mostly extend to the high acceleration side denoting samples of highly effective preconditioners.

For a fixed truncation order p, the PC truncation error will be dominant for large D and one could expect the acceleration to stagnate at some point. Such a stagnation is not visible, for the range of values

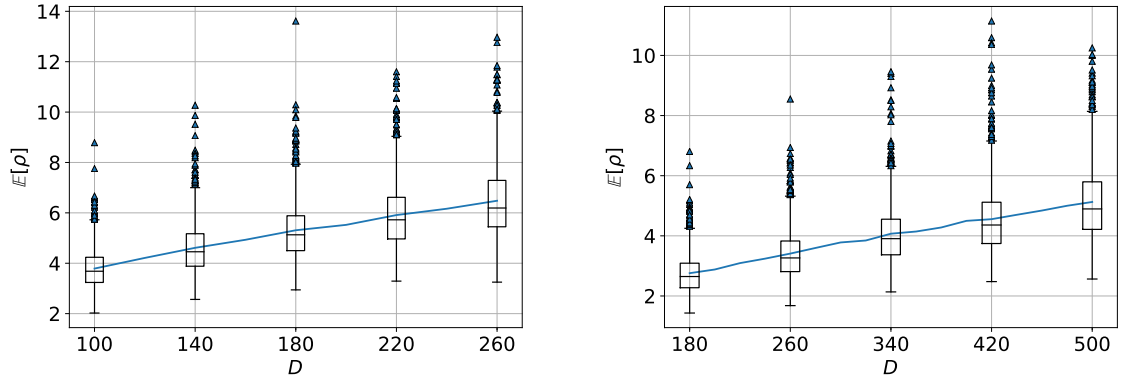


Figure 9: Average acceleration and corresponding boxplots of the FPCG method as a function of the number D of subdomains. Stochastic field $\boldsymbol{\kappa}$ with $\sigma^2 = 1$, $\gamma = 1.2$ and $\ell_c = 0.05$ (left) and $\ell_c = 0.02$ (right). Other parameters are $p = 4$ and $N_{\mathrm{KL}} = 4$.

for D shown in Figure 9. This is explained by the constant increase with D of the number of iterations to convergence in the MPCG method, caused by the increasing size of the Schur complement. In contrast, the number of FPCG iterations to convergence continuously decreases with D, as shown below in the case of a local adaptation of N_{KL} (see Fig. 12), so the acceleration improves with D. Further, although the KL truncation errors become small for large enough D when N_{KL} is fixed, reducing the PC order p is not an effective way to reduce the computational complexity without relying on adaptive PC basis selection. Such a procedure could be for instance an anisotropic PC truncation. The PC order needed to achieve a given accuracy is asymptotically related to the variance of $\boldsymbol{\kappa}$, and not on the apparent correlation length ℓ_{loc}. From this observation, we conclude that the number of local random variables N_{KL} in the local KL expansions is the main parameter controlling the efficiency and mitigating the computational complexity of the FPCG method. This aspect is investigated in the following.

4.4.2. Adapting the local KL approximations

Let us consider a fixed PC order p ensuring a limited PC truncation error on the approximation of the influence operators. A fine control of the KL truncation error can be achieved by adapting the number of local KL modes, $N_{\mathrm{KL}}^{(d)}$, in each subdomain. Specifically, in our settings, the fraction of energy R_{KL} of the Gaussian field accounted for by the KL expansions is

$$R_{\mathrm{KL}} = \frac{\sum_{d=1}^{D} \sum_{i=1}^{N_{\mathrm{KL}}^{(d)}} \lambda_i^{(d)}}{\sigma^2 |\Omega|}, \tag{47}$$

where $|\Omega|$ is the measure of the domain ($|\Omega| = 1$ in our case) and the $\lambda_i^{(d)}$ are the eigenvalues of the local KL expansion of the Gaussian field over $\Omega^{(d)}$ (see (25)). Tuning all the $N_{\mathrm{KL}}^{(d)}$ to obtain a prescribed value for R_{KL} is not convenient. It is easier to rely on a local criterion and set $N_{\mathrm{KL}}^{(d)}$ in each of the subdomains accordingly. Let us denote by $\tau \in (0, 1)$ the local tolerance on the KL error; we define $N_{\mathrm{KL}}^{(d)}$ as the smallest positive integer such that

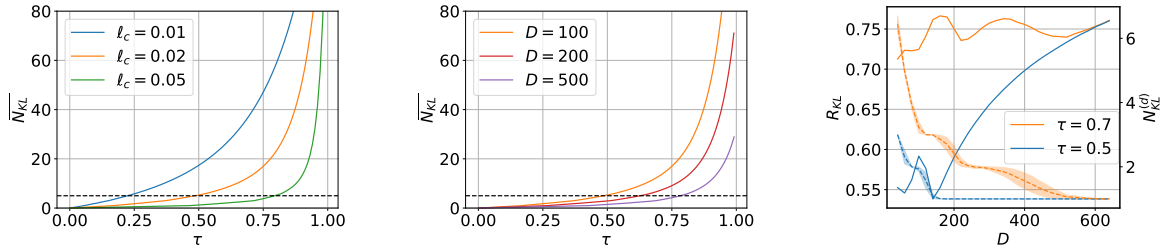
$$\sum_{i=1}^{N_{\mathrm{KL}}^{(d)}} \lambda_i^{(d)} \geq \tau \sigma^2 |\Omega^{(d)}|. \tag{48}$$

One can easily check that (48) implies $R_{\mathrm{KL}}(\tau) \geq \tau$. In other words, $1 - \tau$ is an upper bound for the relative KL error. In the case of subdomains with roughly equal diameters, $N_{\mathrm{KL}}^{(d)}$ satisfying (48) does not vary much

from one subdomain to another and we introduce the average number of local modes

$$\overline{N}_{\mathrm{KL}} = \frac{1}{D}\sum_{d=1}^{D} N_{\mathrm{KL}}^{(d)}.$$

Figure 10a presents the evolution of $\overline{N}_{\mathrm{KL}}$ as a function of the local KL tolerance τ in the case of a field $\boldsymbol{\kappa}$ with parameter $\gamma = 1.2$, different correlation lengths, and a partition in $D = 100$ subdomains (these results are independent of σ^2). It is seen that for a local tolerance of $\tau = 0.6$ one needs roughly 2 local modes (on average) per subdomain when $\ell_c = 0.05$, while 25 modes are necessary when $\ell_c = 0.01$. For the stochastic field with $\ell_c = 0.02$, Fig. 10b shows the evolution of $\overline{N}_{\mathrm{KL}}$ with τ for numbers of subdomains $D = 100$, 200 and 500. When D increases from 100 to 500, $\overline{N}_{\mathrm{KL}}$ to satisfy (48) with $\tau = 0.6$ decreases from 7 to 3, owing to the reduction of the apparent correlation length ℓ_{loc}. Fig. 10c reports the resulting fraction of energy $R_{\mathrm{KL}}(\tau)$ and the number of local modes $N_{\mathrm{KL}}^{(d)}$ for $\tau = 0.7$ and 0.5. Here $\gamma = 1.2$ and $\ell_c = 0.05$. It is seen that for all D the fraction of energy (solid line) remains greater than τ. However, the behavior for the two τ are quite different. For $\tau = 0.5$ the average value of $N_{\mathrm{KL}}^{(d)}$ quickly drops to one (left axis) and exhibits a low variability between subdomains (the extent of the shaded areas correspond to RMS value of $N_{\mathrm{KL}}^{(d)}$). After $N_{\mathrm{KL}}^{(d)}$ reaches 1, around $D \approx 150$, R_{KL} starts to increase monotonically to attain values significantly higher than the lower bound $\tau = 0.5$: we have $R_{\mathrm{KL}}(0.5) = 0.75$ for $D = 600$. In contrast, when a higher precision on the local KL approximation is required, setting $\tau = 0.7$, $\overline{N}_{\mathrm{KL}}$ decreases at a slower pace, has slightly higher RMS values, and reaches one at $D \approx 600$. Therefore, R_{KL} remains higher but close to $\tau = 0.7$ over the range of D presented. For $D > 600$, $R_{\mathrm{KL}}(\tau)$ would continue to increase as the KL approximations with just one mode per subdomains will become more and more accurate as the subdomains size decreases. Eventually, there will be just one element per subdomain and the KL approximation will be an "exact" element-wise constant approximation of $\boldsymbol{\kappa}$.

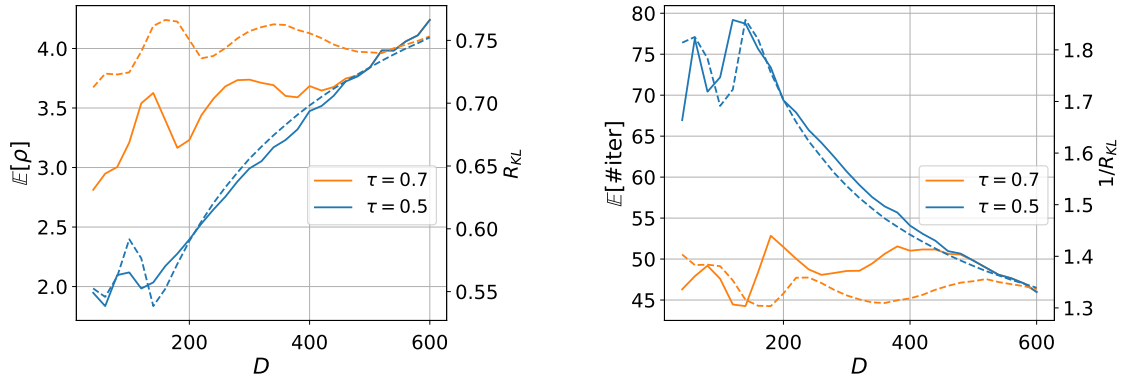


(a) Different ℓ_c ($D = 100$). (b) Different D ($\ell_c = 0.02$). (c) R_{KL} with D ($\ell_c = 0.05$).

Figure 10: Local KL adaptation: average number of local modes $\overline{N}_{\mathrm{KL}}$ as a function of the tolerance τ on the local truncation for different correlation lengths and numbers of subdomains (left and middle plots); fraction of energy R_{KL} (solid lines) and number of local modes $N_{\mathrm{KL}}^{(d)}$ (dashed lines and shaded areas) as functions of the number of subdomains (right plot). The dashed lines correspond to the average $\overline{N}_{\mathrm{KL}}$, while the shaded areas represent the RMS deviation of $N_{\mathrm{KL}}^{(d)}$ around that mean. Case of $\gamma = 1.2$.

We now return to the analysis of the efficiency of the FPC preconditioner. We fix the stochastic field parameters to $\sigma^2 = 1$, $\gamma = 1.2$, and $\ell_c = 0.05$. The PC order is set to $p = 4$ with total degree truncation, and we use the previous two tolerances $\tau = 0.7$ and 0.5 to adapt the local KL expansions. Figure 11a reports the resulting average FPCG acceleration $\mathbb{E}[\boldsymbol{\rho}]$ (solid lines, left axis) and fraction of energy R_{KL} (dashed lines, right axis) as functions of the number of subdomains D. The results for $\tau = 0.5$ show a continuous improvement of the acceleration for $D > 100$. This constant improvement is not surprising as we have just seen that this value of τ leads quickly to $N_{\mathrm{KL}}^{(d)} = 1$ (see Fig. 10c) and, subsequently, to a continuous reduction of the KL error for $D > 100$. In this regime, the FPC preconditioner gets closer and closer to the exact stochastic Schur complement of the discrete problem, up to PC errors that are not too significant for variance level $\sigma^2 = 1$. The parallel evolutions of R_{KL} and $\mathbb{E}[\boldsymbol{\rho}]$ are also evident in Fig. 11a for $\tau = 0.5$.

The case of $\tau = 0.7$ is more complex. First, a detailed inspection of the results for $\tau = 0.7$ reveals correspondences between the variations with D of R_{KL} and the fluctuations around the global trend of $\mathbb{E}[\boldsymbol{\rho}]$, as expected. However, there is no clear improvement of the trend in R_{KL} to explain the continuous improvement of the acceleration with D. A possible explanation is a decreasing PC error when $N_{\mathrm{KL}}^{(d)}$ decreases, because of fewer high-order interactions between modes to be accounted for. However, previous experiments with N_{KL} fixed for all subdomains have demonstrated that, in the present situation, the PC truncation has a limited impact and cannot explain the improvement of the FPCG acceleration. An alternative explanation concerns the definition of the acceleration: it could be that $\mathbb{E}[\boldsymbol{\rho}]$ increases with D because of the degradation of the performance of the MPCG method. This explanation is supported by the results of Fig. 11b, which reports the average number of iterations to converge in the FPCG method. It is seen that for $\tau = 0.5$, the number of iterations decreases continuously with D, while it remains essentially constant when $\tau = 0.7$. In addition, the number of FPCG iterations are seen to follow closely the evolutions of $1/R_{\mathrm{KL}}$ (dashed lines). This finding means that the average cost of solving the sampled elliptic problem is controlled by the KL error and is independent of the size of the Schur complement. In other words, the FPCG method is scalable with D.



(a) $\mathbb{E}[\boldsymbol{\rho}]$ and R_{KL} with D.

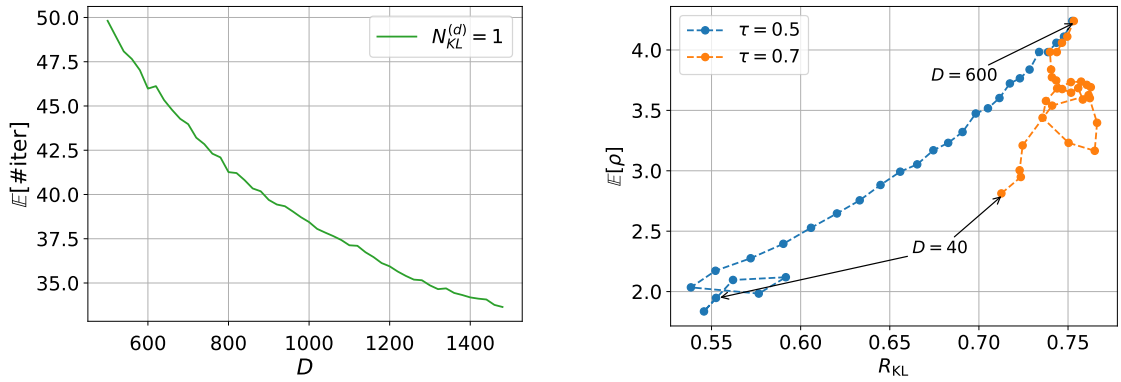
(b) # of FPCG iterations with D.

Figure 11: Performance of the FPCG method with the number of subdomains using local adaptation of $N_{\mathrm{KL}}^{(d)}$. Solid lines represent $\mathbb{E}[\boldsymbol{\rho}]$ (left) and $\mathbb{E}[\#\mathrm{iter}]$ (right), while dashed lines represent R_{KL} (left) and $1/R_{\mathrm{KL}}$ (right). Parameters are $p = 4$, $\gamma = 1.2$, $\sigma^2 = 1$ and $\ell_c = 0.05$.

For $\tau = 0.7$, Fig. 12a shows that increasing the number of subdomains after $D = 600$, such that $N_{\mathrm{KL}}^{(d)} = 1$ for all d, yields the same behavior as for $\tau = 0.5$ and $D > 100$: a continuous decay of the number of iterations to convergence and therefore an improvement of the acceleration. The differences in the two regimes, before and after reaching $\overline{N}_{\mathrm{KL}} = 1$, are illustrated in Fig. 12b. The plot shows the average acceleration as a function of the fraction of the energy R_{KL}. The results correspond to $D \in [40, 600]$. For $\tau = 0.7$, the correlation between the acceleration ($\mathbb{E}[\boldsymbol{\rho}]$) and the fraction of energy (R_{KL}) is not trivial before D is large enough to have $N_{\mathrm{KL}}^{(d)} = 1$. On the contrary, for $\tau = 0.5$ the relation between the two quantities is clear.

4.5. Complexity analysis

A relevant question concerns the selection of the number D of subdomains and other numerical parameters of the FPC preconditioner, namely, the PC order p and, in the case of local adaptation, the tolerance τ for the local KL truncation. Choosing these parameters partly depends on the problem, through the geometry of the domain and the properties of $\boldsymbol{\kappa}$, and its spatial discretization. Another consideration concerns the balance between the cost of constructing the stochastic preconditioner and the resulting acceleration achieved in the sampling stage. For instance, increasing the discretization parameters p and τ results in a more costly construction that will be beneficial only if the computational savings during the sampling stage are large enough. As a starting point, one can assume that sufficiently many samples will be computed in



(a) Average # of FPCG iterations with D ($\tau = 0.7$)

(b) Average FPCG acceleration *vs.* R_{KL}.

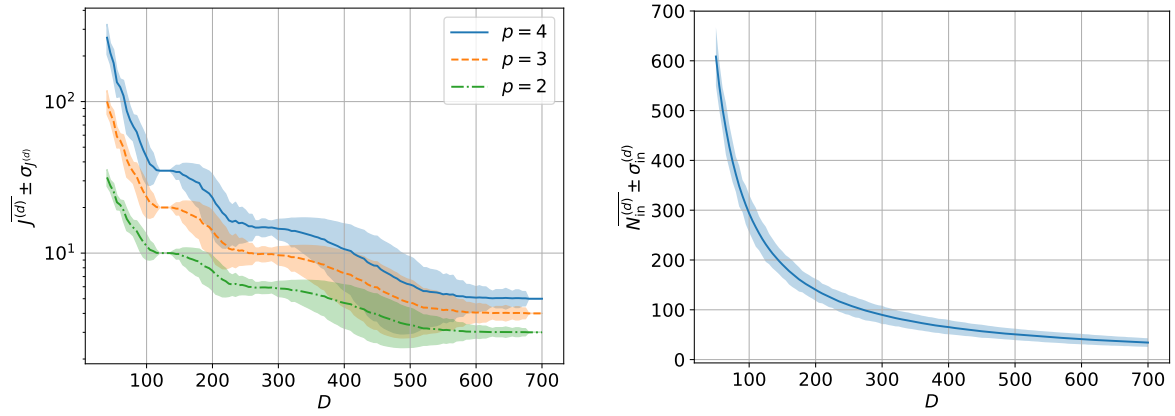
Figure 12: Performance of the FPCG method. Left: $\tau = 0.7$ and $D > 600$; Right: $\tau = 0.5$ and 0.7, $D \in [40, 600]$. Other parameters are $p = 4$, $\gamma = 1.2$, $\sigma^2 = 1$ and $\ell_c = 0.05$.

the sampling stage to payback any improvement of the net FPC efficiency. In other words, if the increase in the construction cost factorizes over sufficiently many samples, it can be considered negligible.

Following this line of reasoning, we still have to consider separately the two regimes discussed in the previous sections: the first regime where D and τ are such that $N_{\text{KL}}^{(d)} > 1$, and the second regime where D and τ are such that $N_{\text{KL}}^{(d)} = 1$. We have seen that the average number of FPCG iterations is mostly controlled by $R_{\text{KL}}(\tau)$, which does not change much with D in the first regime. Therefore, from the sampling point of view, there is no clear interest in increasing D in this regime. However, changing D in this regime does impact the construction cost and the computational complexity of the FPC preconditioner, as analyzed hereafter. When the second regime is attained, the number of PCG iterations decreases with D suggesting that larger D are always beneficial. Of course, this conclusion does not account for the possible divergence of the preconditioner's evaluation cost and memory requirements for its storage, nor for the cost of its application within the PCG iterations. In the rest of the section we attempt to address some of these questions by providing elements on the evolution of the FPC preconditioner complexity with the numerical parameters.

We start by reporting in Fig. 13 the evolution with D of the FPC preconditioner's complexity. Figure 13a shows the average value and RMS bounds of the size of the local PC bases for PC orders $p = 2$ to 4 and a tolerance $\tau = 0.7$. In this example, we relied on the total degree truncation and a stochastic field $\boldsymbol{\kappa}$ characterized again by $\sigma^2 = 1$, $\ell_c = 0.05$, and $\gamma = 1.2$. The decay of $J^{(d)}$ with D is very fast when D is in the first regime, because of the dependence of $J^{(d)}$ on $N_{\text{KL}}^{(d)}$, specifically $J^{(d)} = (p + N_{\text{KL}}^{(d)})!/p!N_{\text{KL}}^{(d)}!$. Increasing ℓ_{loc} through smaller subdomains allows for a reduction of $N_{\text{KL}}^{(d)}$ (see Fig. 10c) that in turns brings a drastic reduction of the size of the local PC bases. Obviously, having a smaller PC basis requires less computational efforts to compute the expansion coefficients of the factorized influence operators. For the (non-optimal) fully tensorized quadrature method implemented in this work, the reduction in the number of influence problem to be solved, $(p+1)^{N_{\text{KL}}^{(d)}}$, achieved through the reduction of $N_{\text{KL}}^{(d)}$ is impressive; even for a linear dependence of the number of influence problems to be solved with the size of the PC bases, as in a regression approach, the complexity reduction would still be huge. Further, not only does the number of influence problem to solve to determine the PC expansion decrease, but the size of the individual influence problems reduces too. This reduction is illustrated in Fig. 13b which reports the evolution with D of the average size of the local finite element problems and the corresponding RMS bounds. Also recall that local influence operators of different subdomains can be computed in parallel.

Then, as D increases, the computational complexity of the PC expansion of the factorized influence operator drops. However, as D increases, there are more and more PC expansions to store in order to



(a) Size $J^{(d)}$ of the PC bases with D

(b) Size $\mathrm{N}_{\mathrm{in}}^{(d)}$ of the local problems with D.

Figure 13: Evolutions of the size $J^{(d)}$ of the local PC bases (for degrees 2 to 4) (left) and of the size $\mathrm{N}_{\mathrm{in}}^{(d)}$ of local FE influence problems (right) as functions of the number of subdomains. The shaded areas represent the RMS bounds around the average. Case of $\tau = 0.7$ and $\boldsymbol{\kappa}$ with $\ell_c = 0.05$, $\sigma^2 = 1$, and $\gamma = 1.2$.

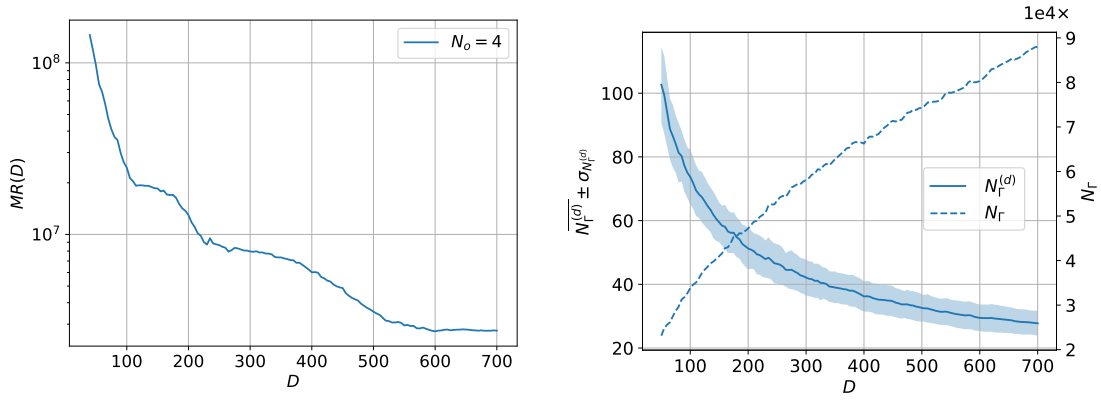
subsequently assemble the realizations of the FPC preconditioner. The memory requirement is therefore a concern. Since the PC coefficients of the stochastic influence operator $[\widetilde{\mathbf{H}}]^{(d)}(\theta)$ are matrices with size $\mathrm{N}_\Gamma^{(d)} \times \mathrm{N}_\Gamma^{(d)}$ (see (41)), the total memory requirement MR to store the PC expansions of all the local influence operators is

$$\mathrm{MR}(D) = \sum_{d=1}^{D} \left(\mathrm{N}_\Gamma^{(d)}\right)^2 J^{(d)}. \tag{49}$$

The evolution of $\mathrm{MR}(D)$ is reported in Figure 14a for our example, with $p = 4$ (evolutions are similar for lower orders). It is seen that the memory requirement reduces with D before it plateaus. This trend is explained by the evolution of $J^{(d)}$, shown in Fig. 13a, which also levels off after $D \approx 500$, and by the joint decrease of $\mathrm{N}_\Gamma^{(d)}$ on D. Even once the size of the PC bases has leveled off, the memory requirement does not increase but remains constant. The dependence of $\mathrm{N}_\Gamma^{(d)}$ on D can be appreciated from Fig. 14b which shows the average value and RMS bounds of number of unknown boundary points in the local problems (solid lines, left axis). We observe that the partitioning procedure employed in this work, a simple k-means algorithm applied on the coordinates of the FE centers, produces subdomains with well-balanced numbers of boundary nodes, owing to the uniformity of the global mesh (see Fig. 1); for more complex discretizations, *e.g.* adapted ones, it could be necessary to rely on more advanced partitioning procedures.

4.6. Discussion

The brief complexity analysis proposed above suggests that it is desirable to use a large number of subdomains to a) reduce the local KL dimension, which induces a subsequent reduction in size of the local PC bases, b) reduce the size of the local influence operators and the size of the local problems involved in their determination, and c) minimize the overall memory requirement for the FPC preconditioner storage. In addition, if the apparent local correlation length ℓ_{loc} can be sufficiently reduced, an additional overall reduction of the number of iterations in the FPCG method can be achieved. However, this rationale does not consider the inherent cost of applying the preconditioner during the sampling stage. Figure 14b presents the evolution of the total number of boundary unknowns, *i.e.*, the size N_Γ of the Schur complement, which grows asymptotically linearly with D. This adverse evolution presents the main limitation of the proposed FPC method, since the application of the preconditioner requires its "inversion" for each sample (see line 9 of Algorithm 2). Obviously, the inversion (factorization) cost of the preconditioner limits the gain brought by the reduction of the FPCG iterations. For D leading to large N_Γ, the cost of the preconditioner inversion



(a) Memory requirements MR(D).

(b) $\mathrm{N}_\Gamma^{(d)}$ and size N_Γ of the Schur complement.

Figure 14: Evolutions with the number of subdomains of the memory requirement (see (49)) for the FPC preconditioner (left) and local number of boundary nodes $\mathrm{N}_\Gamma^{(d)}$ and total size of the Schur complement (right). The shaded area represents the RMS bounds around the average value.

may even dominate the cost of the MPCG iterations. In these conditions, it is difficult to provide a clear rationale to select D and possibly τ and p to achieve the best performance of the FPCG method in the sampling stage. A possible practical way to proceed would consist in determining D such that the inversion cost of the preconditioner does not exceed a fraction of the average computational cost of solving one sample with the MPCG method. Then, D being fixed, the PC order p can be adjusted to ensure limited PC truncation error (mostly depends on σ^2) while τ can be tuned to balance the PC complexity ($J^{(d)}$ through its dependence on $N_{\mathrm{KL}}^{(d)}$) with the performance (related to $R_{\mathrm{KL}}(\tau)$).

Besides the complexity, one may be also concerned with the computational cost of the FPC preconditioner. The interested reader should refer to [34], where parallel experiments demonstrated the scalability of the PC surrogates construction, thanks to trivial parallelism, as well as the low cost associated to the assembly of approximated [S] for each sample.

5. Conclusions

In this paper, we presented a DD approach to generate samples of the solution of a stochastic elliptic equation with a random coefficient field $\boldsymbol{\kappa}$. Each solution sample is solved in a domain decomposition framework leading to the resolution by a CG method of the Schur system (19) associated with the particular sample $\kappa^{(m)}$ of the stochastic coefficient field.

We proposed to speed up the resolution of the sampled Schur problem using a stochastic preconditioner: a preconditioner that is adapted to individual samples $\kappa^{(m)}$. Our approach must be contrasted with classical methods relying on the same deterministic preconditioner for all the samples, such as the preconditioning with the mean or median Schur operator. In our approach, the stochastic preconditioner is determined in a preprocessing stage and subsequently evaluated during the sampling stage. The preconditioner is composed of local polynomial approximations of the local influence operators (boundary-to-boundary maps) associated with the (non-overlapping) partition of the domain into D subdomains. The construction of the preconditioner presents the advantage of relying on local operators. The localization on the subdomains enables a parallel implementation and, more importantly, a reduced computational complexity. Specifically, the approach exploits the introduction of local random variables to represent the stochastic coefficient over the considered subdomain. One fundamental contribution of the work is the derivation of a factorized approximation of the local influence operators. The factorized form ensures the inherent positivity of the preconditioners' realizations and provides massive robustness and efficiency improvement over more straightforward constructions.

The resulting FPCG method has been tested and compared to alternatives (deterministic median-based preconditioner, direct-PC expansion) on a model problem in two spatial dimensions. The tests empirically demonstrate significant reductions in the number of PCG iterations to convergence. For a stochastic coefficient field with high variance and low correlation, our preconditioner allows us to obtain the solution up to 7 times faster in terms of iterations compared with the reference median preconditioner. The main mechanisms controlling the efficiency of the FPCG method have also been evidenced, together with the influence of the method's numerical parameters. Finally, we proposed a brief complexity analysis of the method to prove that the preconditioner's construction is scalable with the number of subdomains.

Our numerical assessment of the FPCG method has only concerned the reduction of the number of iterations compared to the median-based preconditioner. For the problems tested, this is sufficient because the two approaches have comparable costs per iteration and the overhead of the FPC preconditioner set-up time is not significant. The situation may be different for more demanding problems where the Cholesky factorization of the FPC preconditioner would become more significant or even too costly. It would be interesting to compare the computational cost of the FPC preconditioner with available alternative preconditioners at a global level. Such a study would raise several difficulties concerning selecting and tuning the preconditioner to be compared with the FPCG method. At the moment, it can only be stated that the FPCG method *potentially* performs better than any other preconditioner since it converges to the ideal preconditioner (*i.e.*, the Schur system) while having a computational complexity that scales well with the discretization parameters (in particular the number of subdomains D). Still, much work remains to demonstrate that these promises are achieved in practice. For instance, a complete parallel implementation of the FPC method and substitution of direct solvers are in order before conducting comparison experiments for the typical problem size for which existing libraries are tailored.

Similarly, although the preconditioner's construction scales well with the number of subdomains, the Schur system's size may become an inherent limitation when considering domains with finer spatial discretizations or in higher dimensions. Even if the preconditioner has a low evaluation cost for each sample, solving the preconditioned problem may become too costly compared to the iteration savings, especially if parallel strategies are not available. As a consequence, future work and subsequent developments must focus on these aspects. In particular, it would be interesting to assess the impact of incomplete factorization strategies on the overall performance of the FPCG method and to explore the direct approximation of the inverse of the Schur system operator. The latter option seems very challenging as the inverse of the Schur operator cannot be expressed, a priori, as the sum of subdomain's contributions, a key aspect to achieving low computational complexity in our approach. We are currently exploring the use of local preconditioners to maintain locality, through multi-preconditioning strategies [47, 48].

Acknowledgments

This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

References

[1] R. E. Caflisch, Monte carlo and quasi-monte carlo methods, Acta Numerica 7 (1998) 1–49.
[2] J. Liu, Monte Carlo Strategies in Scientific Computing, Springer Verlag, 2001.
[3] K. A. Cliffe, M. B. Giles, R. Scheichl, A. L. Teckentrup, Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients, Computing and Visualization in Science 14 (1) (2011) 3.
[4] R. G. Ghanem, P. D. Spanos, Stochastic finite elements: a spectral approach, Courier Corporation, 2003.
[5] O. Le Maître, O. M. Knio, Spectral methods for uncertainty quantification: with applications to computational fluid dynamics, Springer Science & Business Media, 2010.
[6] M. K. Deb, I. M. Babuška, J. T. Oden, Solution of stochastic partial differential equations using Galerkin finite element techniques, Computer Methods in Applied Mechanics and Engineering 190 (48) (2001) 6359–6372.
[7] O. P. Le Maître, M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, A stochastic projection method for fluid flow: Ii. random process, Journal of Computational Physics 181 (1) (2002) 9 – 44.
[8] O. Le Maître, O. Knio, B. Debusschere, H. Najm, R. Ghanem, A multigrid solver for two-dimensional stochastic diffusion equations, Computer Methods in Applied Mechanics and Engineering 192 (41–42) (2003) 4723 – 4744.

[9] P. Frauenfelder, C. Schwab, R. A. Todor, Finite elements for elliptic problems with stochastic coefficients, Computer methods in applied mechanics and engineering 194 (2) (2005) 205–228.
[10] O. Knio, O. Le Maître, Uncertainty propagation in CFD using polynomial chaos decomposition, Fluid Dyn. Res. 38 (2006) 616–640.
[11] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal on Numerical Analysis 45 (3) (2007) 1005–1034.
[12] R. T. F. Nobile, C. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM J. Numer. Anal. 46 (5) (2008) 2309–2345.
[13] H. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Ann. Rev. Fluid Mech. 41 (2009) 35–52.
[14] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations, Comput. Methods Appl. Mech. Engrg. 196 (45-48) (2007) 4521–4537.
[15] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: Invariant subspace problem and dedicated algorithms, Computer Methods in Applied Mechanics and Engineering 197 (51–52) (2008) 4718 – 4736.
[16] A. Nouy, O. Le Maître, Generalized spectral decomposition for stochastic nonlinear problems, J. Comput. Phys. 228 (2009) 202–235.
[17] A. Cohen, R. DeVore, C. Schwab, Convergence rates of best n-term Galerkin approximations for a class of elliptic SPDEs, Foundations of Computational Mathematics 10 (6) (2010) 615–646.
[18] J. Beck, R. Tempone, F. Nobile, L. Tamellini, On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods, Mathematical Models and Methods in Applied Sciences 22 (09) (2012) 1250023.
[19] J. Beck, F. Nobile, L. Tamellini, R. Tempone, Convergence of quasi-optimal stochastic galerkin methods for a class of PDEs with random coefficients, Computers & Mathematics with Applications 67 (4) (2014) 732 – 751, high-order Finite Element Approximation for Partial Differential Equations.
[20] L. Tamellini, O. Le Maître, A.Nouy, Model reduction based on Proper Generalized decomposition for stochastic steady incompressible Navier Stokes equations, SIAM J. Scientific Computing 36 (3) (2014) 1089–1117.
[21] A. Chkifa, A. Cohen, C. Schwab, High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs, Foundations of Computational Mathematics 14 (4) (2014) 601–633.
[22] G. Blatman, B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression, Journal of Computational Physics 230 (6) (2011) 2345 – 2367.
[23] M. Papadrakakis, V. Papadopoulos, Robust and efficient methods for stochastic finite element analysis using monte carlo simulation, Computer Methods in Applied Mechanics and Engineering 134 (3-4) (1996) 325–340.
[24] M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection, Combustion and Flame 132 (3) (2003) 545–555.
[25] P. R. Conrad, Y. M. Marzouk, Adaptive Smolyak pseudospectral approximations, SIAM Journal on Scientific Computing 35 (6) (2013) A2643–A2670.
[26] P. G. Constantine, M. S. Eldred, E. T. Phipps, Sparse pseudospectral approximation method, Computer Methods in Applied Mechanics and Engineering 229–232 (2012) 1 – 12.
[27] M. F. Pellissetti, R. G. Ghanem, Iterative solution of systems of linear equations arising in the context of stochastic finite elements, Advances in Engineering Software 31 (8-9) (2000) 607–616.
[28] E. Rosseel, S. Vandewalle, Iterative solvers for the stochastic finite element method, SIAM Journal on Scientific Computing 32 (1) (2010) 372–397.
[29] C. E. Powell, H. C. Elman, Block-diagonal preconditioning for spectral stochastic finite-element systems, IMA Journal of Numerical Analysis 29 (2) (2009) 350–375.
[30] W. Subber, A. Sarkar, Domain decomposition method of stochastic PDEs: a two-level scalable preconditioner, Journal of Physics: Conference Series 341 (1) (2012) 012033.
[31] W. Subber, S. Loisel, Schwarz preconditioners for stochastic elliptic PDEs, Computer Methods in Applied Mechanics and Engineering 272 (2014) 34–57.
[32] W. Subber, A. Sarkar, A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner, Journal of Computational Physics 257 (2014) 298–317.
[33] A. A. Contreras, P. Mycek, O. P. Le Maître, F. Rizzi, B. Debusschere, O. M. Knio, Parallel domain decomposition strategies for stochastic elliptic equations. Part A: Local Karhunen–Loève representations, SIAM Journal on Scientific Computing 40 (4) (2018) C520–C546.
[34] A. A. Contreras, P. Mycek, O. P. Le Maître, F. Rizzi, B. Debusschere, O. M. Knio, Parallel domain decomposition strategies for stochastic elliptic equations: Part B: Accelerated monte carlo sampling with local PC expansions, SIAM Journal on Scientific Computing 40 (4) (2018) C547–C580.
[35] Y. Saad, Iterative methods for sparse linear systems, Vol. 82, siam, 2003.
[36] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, New York, 1999.
[37] A. Toselli, O. Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, Springer-Verlag, Berlin Heidelberg, 2005.
[38] T. P. A. Mathew, Domain decomposition methods for the numerical solution of partial differential equations, no. 61 in Lecture notes in computational science and engineering, Springer, Berlin, 2008.
[39] Y. Chen, J. Jakeman, C. Gittelson, D. Xiu, Local polynomial chaos expansion for linear differential equations with high dimensional random inputs, SIAM Journal on Scientific Computing 37 (1) (2015) A79–A102.

[40] S. Pranesh, D. Ghosh, Addressing the curse of dimensionality in SSFEM using the dependence of eigenvalues in KL expansion on domain size, Computer Methods in Applied Mechanics and Engineering 311 (2016) 457–475.
[41] T. Y. Hou, Q. Li, P. Zhang, Exploring the locally low dimensional structure in solving random elliptic pdes, Multiscale Modeling & Simulation 15 (2) (2017) 661–695.
[42] R. Tipireddy, P. Stinis, A. M. Tartakovsky, Basis adaptation and domain decomposition for steady-state partial differential equations with random coefficients, Journal of Computational Physics 351 (2017) 203–215.
[43] J. Charrier, Strong and weak error estimates for elliptic partial differential equations with random coefficients, SIAM Journal on Numerical Analysis 50 (1) (2012) 216–246.
[44] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (4) (1938) 897–936.
[45] R. H. Cameron, W. T. Martin, The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals, The Annals of Mathematics 48 (2) (1947) 385–392.
[46] M. Salloum, A. Alexanderian, O. P. L. Maître, H. N. Najm, O. M. Knio, Simplified CSP analysis of a stiff stochastic ODE system, Computer Methods in Applied Mechanics and Engineering 217-220 (2012) 121 – 138.
[47] R. Bridson, C. Greif, A multipreconditioned conjugate gradient algorithm, SIAM Journal on Matrix Analysis and Applications 27 (4) (2006) 1056–1068.
[48] N. Spillane, An adaptive multipreconditioned conjugate gradient algorithm, SIAM journal on Scientific Computing 38 (3) (2016) A1896–A1918.
[49] K. Karhunen, Über lineare Methoden in der Wahrscheinlichkeitsrechnung, Vol. 37, Sana, 1947.
[50] P. Lévy, M. Loeve, Processus stochastiques et mouvement brownien, Gauthier-Villars Paris, 1965.

Appendix A. Global KL-expansion

Let $\boldsymbol{G} \in L^2(\Omega \times \Theta)$ be a centered stochastic field with covariance function $C : \Omega \times \Omega \to \mathbb{R}$:

$$C(x, x') \doteq \mathbb{E}[\boldsymbol{G}(x, \theta)\boldsymbol{G}(x', \theta)]. \tag{A.1}$$

The Karhunen-Loève (KL) expansion of $\boldsymbol{G}$ is given by

$$\boldsymbol{G}(x, \theta) = \sum_{i=1}^{+\infty} \sqrt{\lambda_i}\phi_i(x)\boldsymbol{\eta}_i(\theta), \tag{A.2}$$

where the eigenpairs $(\lambda_i, \phi_i(x))$ are the solutions of the eigenvalue problem [49, 50, 5]

$$\int_\Omega C(x, x')\phi_i(x')\, dx' = \lambda_i\phi_i(x), \quad \text{with } \langle \phi_i, \phi_j\rangle_\Omega = \delta_{i,j} \text{ and } \lambda_i \geq \lambda_{i+1}. \tag{A.3}$$

The eigenvalues satisfying (A.3) are non-negative, and the eigenfunctions are normalized according to the spatial norm introduced in Section 2.1. The random variables $\boldsymbol{\eta}_i(\theta)$ are given by

$$\boldsymbol{\eta}_i(\theta) \doteq \frac{1}{\sqrt{\lambda_i}}\langle \boldsymbol{G}(x, \theta), \phi_i(x)\rangle_\Omega. \tag{A.4}$$

Since $\boldsymbol{G}$ has zero mean, the random variables $\boldsymbol{\eta}_i(\theta)$ have zero mean. Further, they form an orthonomal set:

$$\mathbb{E}[\boldsymbol{\eta}_i\boldsymbol{\eta}_j] = \delta_{i,j}.$$

In practice, the KL expansion must be truncated to the first N_{KL} dominant modes to result in

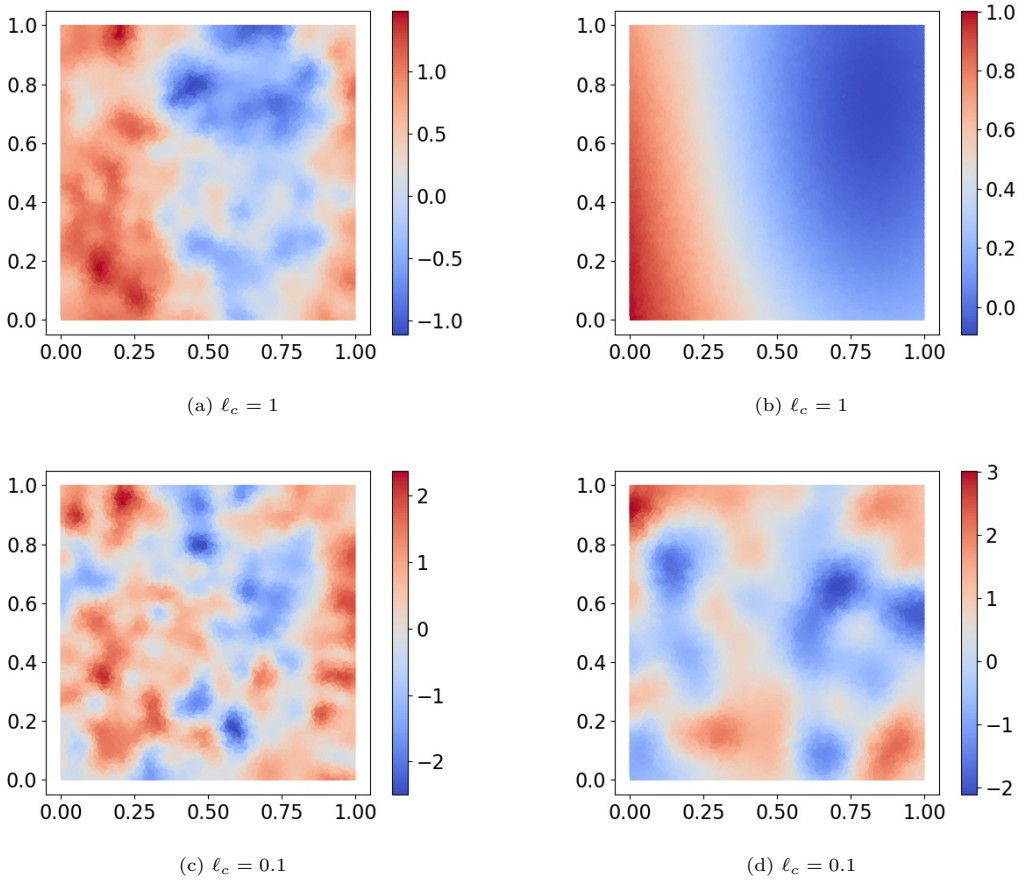
$$\boldsymbol{G}(x, \theta) \approx \widehat{\boldsymbol{G}}(x, \theta) \doteq \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\lambda_i}\phi_i(x)\boldsymbol{\eta}_i(\theta). \tag{A.5}$$

The norm of the KL truncation error $\boldsymbol{G} - \widehat{\boldsymbol{G}}$ is simply given by the sum of the disregarded eigenvalues,

$$\mathrm{E}^2_{N_{\mathrm{KL}}} = \mathbb{E}\Big[\|\boldsymbol{G} - \widehat{\boldsymbol{G}}\|^2\Big] = \sum_{i,j>N_{\mathrm{KL}}} \sqrt{\lambda_i\lambda_j}\langle \phi_i, \phi_j\rangle_\Omega \mathbb{E}[\boldsymbol{\eta}_i\boldsymbol{\eta}_j] = \sum_{i>N_{\mathrm{KL}}} \lambda_i,$$

from which the convergence follows because $\sum_i \lambda_i < \infty$ for a second-order field. In figure A.16 we illustrate the decay of $\sqrt{\lambda_i}$ for different types of covariance functions and correlation lengths.

For physically relevant fields, the frequency content of the eigenfunctions $\phi_i(x)$ increases with the mode index i. The first modes account for the large scale deviations, while the higher order modes represent short scale details of the field. This observation explains why rougher fields typically have low decaying spectra, while highly convergent spectra are characteristic of smooth and highly correlated random fields. Therefore, two random fields with the same norm will demand different truncation, depending on their roughness and correlation properties, to yield the same KL truncation error. To illustrate this point, we show in figure A.15 typical realizations of a Gaussian field in the unit square domain with covariance in (8), with parameter $\gamma = 1$ (left) and $\gamma = 2$ (right), and correlation length $\ell_c = 1$ (top) and $\ell_c = 0.1$ (bottom), and $\sigma^2 = 1$.



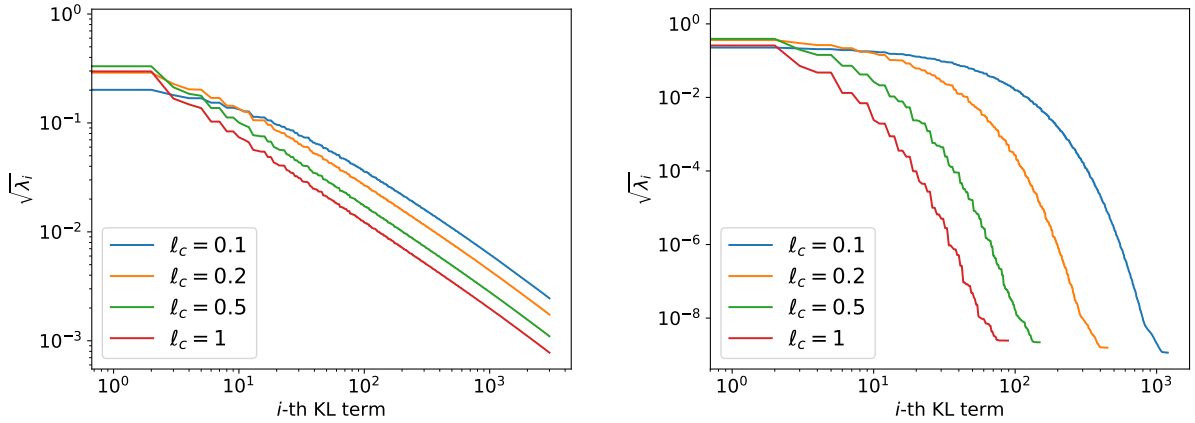
(a) $\ell_c = 1$ (b) $\ell_c = 1$

(c) $\ell_c = 0.1$ (d) $\ell_c = 0.1$

Figure A.15: Sample of the field $\boldsymbol{G}$ for the covariance (8) with parameter $\gamma = 1$ (left), $\gamma = 2$ (right), $\ell_c = 1$ (top), $\ell_c = 0.1$ (bottom) and $\sigma^2 = 1$

Figure A.16 shows the decay of $\sqrt{\lambda_i}$ for different values of the correlation length ℓ_c and $\gamma = 1$ (left) and $\gamma = 2$ (right). When ℓ_c decreases, the energy is distributed over more modes, denoting that short-scale fluctuations are proportionally more significant. We also observe how the roughness impacts the asymptotic decay rates, indicating clearly that correlation function (8) is much more demanding for $\gamma = 1$ than for $\gamma = 2$.

Figure (A.17) illustrates the effect of the roughness on the KL truncation error. It shows the KL approximations of a particular realization of the field with covariance defined in (8), with $\sigma^2 = 1$, $\ell_c = 0.5$, and $\gamma = 1$ (left) and $\gamma = 2$ (right). Plots correspond to using $N_{\mathrm{KL}} = 5$, 20 and 60 (from top to bottom)



(a) Exponential convariance ($\gamma = 1$)

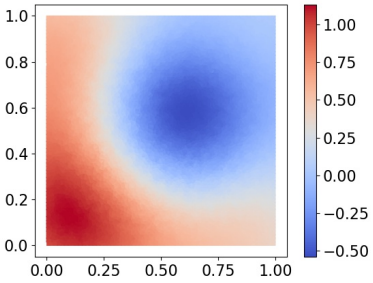
(b) Gaussian convariance ($\gamma = 2$)

Figure A.16: Spectral decay of the KL expansion for the covariance function C in (8) with $\gamma = 1$ (left) and $\gamma = 2$ (right) and several correlation lengths ℓ_c as indicated.

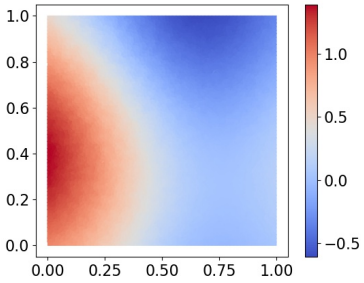
modes in the approximation. The normalized approximation error on this particular realization,

$$\mathrm{e}_{\mathrm{KL}} \doteq \frac{\|\boldsymbol{G}\left(x, \theta^{(m)}\right) - \widetilde{\boldsymbol{G}}\left(x, \theta^{(m)}\right)\|_{\Omega}}{\|\boldsymbol{G}\left(x, \theta^{(m)}\right)\|_{\Omega}}, \tag{A.6}$$

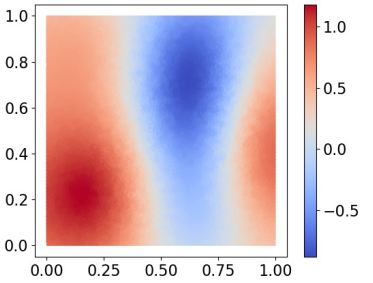
is also indicated. We see that for $\gamma = 1$, the convergence with N_{KL} is slow, reaching a normalized error of roughly 30% for $N_{\mathrm{KL}} = 60$. In contrast, the approximation from the covariance with $\gamma = 2$ quickly converges with a normalized error lower than 10% with only $N_{\mathrm{KL}} = 10$ modes. For more correlated fields (larger ℓ_c), the convergence improves.



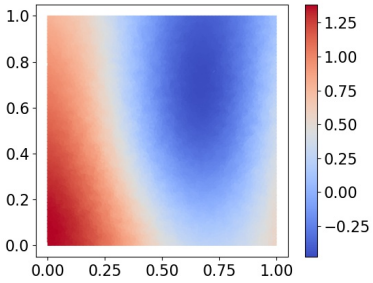
(a) $N_{\mathrm{KL}} = 5$, $\mathrm{e}_{\mathrm{KL}} \approx 0.73$.



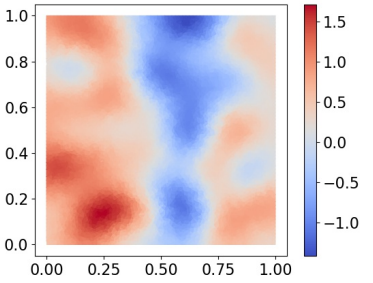
(b) $N_{\mathrm{KL}} = 5$, $\mathrm{e}_{\mathrm{KL}} \approx 0.38$.



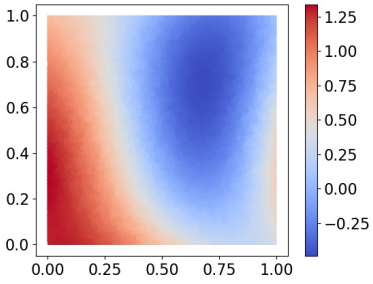
(c) $N_{\mathrm{KL}} = 10$, $\mathrm{e}_{\mathrm{KL}} \approx 0.57$.



(d) $N_{\mathrm{KL}} = 10$, $\mathrm{e}_{\mathrm{KL}} \approx 0.07$.



(e) $N_{\mathrm{KL}} = 60$, $\mathrm{e}_{\mathrm{KL}} \approx 0.34$.



(f) $N_{\mathrm{KL}} = 60$, $\mathrm{e}_{\mathrm{KL}} \approx 6 \times 10^{-6}$.

Figure A.17: Truncated KL expansions for a fixed sample of $\boldsymbol{G}$ with covariance based on a correlation length $\ell_c = 0.5$, variance $\sigma^2 = 1$, roughness parameter $\gamma = 1$ (left), $\gamma = 2$ (right), and using $N_{\mathrm{KL}} = 5$, 10 and 60 (from top to bottom). The corresponding $L^2(\Omega)$ errors e_{KL} are also indicated.