

Parallel Domain Decomposition Strategies for Stochastic Elliptic Equations

Part A: Local KL Representations

Andres A. Contreras* Paul Mycek† Olivier P. Le Maître‡
Francesco Rizzi§ Bert Debuschere ¶ Omar M. Knio||**

June 12, 2017

Abstract

This work presents a method to efficiently determine the dominant Karhunen-Loève (KL) modes of a random process with known covariance function. The truncated KL expansion is one of the most common techniques for the approximation of random processes, primarily because it is an optimal representation, in the mean squared error sense, with respect to the number of random variables in the representation. However, finding the KL expansion involves solving integral problems, which tends to be computationally demanding. This work addresses this issue by means of a work-subdivision strategy based on a domain decomposition approach, enabling the efficient computation of a possibly large number of dominant KL modes. Specifically, the computational domain is partitioned into smaller non-overlapping subdomains, over which independent local KL decompositions are performed to generate local bases which are subsequently used to discretize the global modes over the entire domain. The latter are determined by means of a Galerkin projection. The procedure leads to the resolution of a reduced Galerkin problem, whose size is not related to the dimension of the underlying discretization space, but is actually determined by the desired accuracy and the number of subdomains. It can also be easily implemented in parallel. Extensive numerical tests are used to validate the methodology and assess its serial and parallel performance. The resulting expansion is exploited in Part B to accelerate the solution of the stochastic partial differential equations using a Monte-Carlo approach.

Keywords: Domain Decomposition, Stochastic Processes, Karhunen-Loève Expansion

1 Introduction

Stochastic Partial Differential Equations (PDEs), and elliptic ones in particular, are increasingly being used to account for situations involving uncertain or incomplete knowledge of the simulated system, and to perform, for example, inference tasks and sensitivity analyses. Computational approaches for the solution of SPDEs conceptually involve three essential steps: the modeling of the input uncertainty, the solution of the governing equations, and ultimately post-processing the output to characterize the uncertainty. This paper (“Part A”) and its companion (“Part B”) focus on the two first steps respectively. We discuss at present a domain decomposition strategy to approximate random fields using local reduced bases and local coordinates. These developments are motivated by the computational approach proposed in Part B, where the structure of local representations is exploited to accelerate the Monte Carlo sampling of the solution.

*Duke University, 121 Hudson Hall, Box 90287, Durham, NC 27708 (andres.contreras@duke.edu).

†Duke University, 144 Hudson Hall, Box 90300, Durham, NC 27708 (paul.mycek@duke.edu).

‡CNRS, LIMSI, Université Paris Saclay, Orsay, France (olm@limsi.fr).

§Sandia National Laboratories, Livermore, CA. (fnrizzi@sandia.gov).

¶Sandia National Laboratories, Livermore, CA. (bjdebus@sandia.gov).

||Duke University, 144 Hudson Hall, Box 90300, Durham, NC 27708 (omar.knio@duke.edu).

**King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.

Modeling and approximating random processes is an important task in many applications and in particular in uncertainty quantification problems. Focusing on random processes in spatial domains, it is often convenient to perform a discretization of the process in both stochastic and spatial dimensions for computational purposes. One way to achieve this is to rely on the representation of the process in terms of random coordinates in a spatial basis, and proceed with the discretization of the spatial basis functions and random coordinates. An overview of several stochastic-discretization methods for random fields is provided in [20, 26, 25]. Among these methods, a common one in the area of computational stochastic mechanics is the Karhunen-Loève (KL) expansion [16, 12, 11]. A key advantage of the KL expansion is that it is optimal, with respect to the number of random variables involved in the representation, in the mean squared error sense. This is particularly attractive when using stochastic spectral methods, such as Polynomial Chaos expansions [5, 14], in view of analyzing the influence of the process on a model solution; that is, performing uncertainty quantification. Indeed, these functional representation methods exploit heavily the smoothness of the model solution with respect to the stochastic coordinates appearing in the KL decomposition. A challenge, however, is that finding the KL expansion of a stochastic process involves decomposing its covariance function. This leads to solving a Fredholm integral equation of the second kind, for a kernel consisting of the two-point covariance function. Decomposing such kernels is computationally demanding, because large spatial discretization grids are routinely considered when solving partial differential equations. A detailed description of Fredholm integral equations of the second kind is provided in [1], and a comparison of different methods that solve the KL expansion is found in [3].

For some particular covariance kernels, an analytical solution of the Fredholm integral equation is available (see for instance [5]). However, this is not typically the case and numerical procedures are required for its spectral decomposition [7]. Different approaches have been devised in order to make more tractable the computation of the KL expansion. For instance, in [22] a wavelet-Galerkin approach is discussed, which provides localized support leading to sparse matrix equations that can be solved at a reduced cost. Another approach is presented in [24], which relies on fast multipole methods to speed up the computations. A more recent attempt at making the problem more tractable is provided in [3], but still finding the KL expansion remains a challenging problem, specially in situations where a large number of terms are needed or the physical space is multidimensional.

This paper proposes an efficient parallel method for the computation of KL expansions with a potentially large number of terms (broad spectrum). The method is based on a domain decomposition technique and we refer to it as the Domain Decomposition KL (DD-KL) method. The proposed strategy involves partitioning the computational domain into smaller non-overlapping subdomains, over which local KL decomposition problems are solved to generate local bases. Our approach elucidates the exact correlation structure between sets of local coordinates associated to different subdomains and is composed of the following three main stages: (i) solving a *local KL expansion* problem over each subdomain; (ii) using the dominant eigenfunctions from the local expansions to assemble a *reduced eigenvalue problem*; and (iii), solving the reduced eigenvalue problem to obtain the desired (global) KL expansion. Our representation is exploited in Part B to accelerate the solution of elliptic PDEs using a Monte-Carlo procedure, specifically applying the local representations to expand the condensed stochastic problem of the domain decomposition formulation as local Polynomial Chaos (PC) expansions.

In addition to the immediate computational advantages of (i) solving small independent local problems and (ii) solving a low dimensional reduced global one, the method also allows us to efficiently distribute and parallelize most of the computations. Finally, as mentioned above, the method yields a representation in terms of independent stochastic coordinates that is convenient to perform uncertainty quantification tasks and sensitivity analyses, to characterize for instance the impact of the random process on a model solution. Indeed, the expansion in terms of independent random coordinates enables both sampling based approaches (*e.g.*, Monte Carlo) and functional expansions (*e.g.* Polynomial Chaos and low rank approximations [5, 13, 21, 27]). Furthermore, the global random coordinates can be related to small sets of local coordinates, over the subdomains, to reduce the computational complexity of the uncertainty quantification problem based on the KL approximation of the process.

The outline of the paper is as follows. In section 2, we recall the KL expansion and detail the proposed domain decomposition method. Also, in this section, a truncation strategy for

the DD-KL method is established in order to control the error in the resulting approximation. In section 3, numerical results are provided illustrating the effectiveness of the DD-KL method and the error control. Next, section 4 is dedicated to illustrating both the serial behavior of the approach and its parallel scalability and efficiency. Finally, in section 5, some concluding remarks are provided.

2 A Domain Decomposition Method for KL Expansions

In this section we introduce the proposed domain decomposition method to approximate the KL expansion of a stochastic process. We start by introducing several notations used throughout the paper. Consider a probability space $\mathcal{P} = (\Theta, \Sigma_\Theta, \mu_\Theta)$, where Θ is the set of events, Σ_Θ a sigma-algebra over Θ and μ_Θ a probability measure. We denote by $\mathbb{E}[\cdot]$ the expectation operator,

$$\mathbb{E}[U] = \int_{\Theta} U(\theta) d\mu_\Theta(\theta), \quad (1)$$

and $L_2(\Theta)$ the corresponding space of second-order random variables, *i.e.*, such that $\mathbb{E}[U^2] < +\infty$. At the deterministic level, we consider Ω a bounded subset of \mathbb{R}^n (with $n = 1, 2$ or 3) and define $L_2(\Omega)$ the space of square integrable functionals $f : \mathbf{x} \in \Omega \mapsto f(\mathbf{x}) \in \mathbb{R}$. We shall denote by $\|\cdot\|_\Omega$ the norm in $L_2(\Omega)$ induced by the scalar product $\langle \cdot, \cdot \rangle_\Omega$:

$$\forall f \in L_2(\Omega), \quad \|f\|_\Omega^2 = \langle f, f \rangle_\Omega = \int_{\Omega} |f(\mathbf{x})|^2 d\mathbf{x}. \quad (2)$$

Finally, we denote by $L_2(\Omega, \Theta)$ the space of real-valued second-order processes $U : \Omega \times \Theta \rightarrow \mathbb{R}$ such that $U(\cdot, \theta) \in L_2(\Omega)$, $U(\mathbf{x}, \cdot) \in L_2(\Theta)$ and

$$\mathbb{E}[\|U(\cdot, \theta)\|_\Omega^2] < +\infty \Leftrightarrow U \in L_2(\Omega, \Theta). \quad (3)$$

2.1 The KL expansion

Let $U \in L_2(\Omega, \Theta)$ be a centered random process with known covariance $C : \Omega \times \Omega \rightarrow \mathbb{R}$:

$$C(\mathbf{x}, \mathbf{x}') \equiv \mathbb{E}[U(\mathbf{x}, \cdot)U(\mathbf{x}', \cdot)]. \quad (4)$$

The truncated KL approximation, U_N , of a second-order stochastic process U , consists in a N -term expansion where each term is composed of the product of a deterministic function of $L_2(\Omega)$ with a random variable of $L_2(\Theta)$. The KL approximation U_N is defined as to minimize the representation error $U - U_N$ in the $L_2(\Omega, \Theta)$ sense. As covariance functions are symmetric and non-negative, it can be shown that the sought KL expansion is given by

$$U_N(\mathbf{x}, \theta) \equiv \sum_{\alpha=1}^N \sqrt{\lambda_\alpha} \eta_\alpha(\theta) \Phi_\alpha(\mathbf{x}), \quad (5)$$

where the λ_α 's and the Φ_α 's are the (dominant) eigenvalues and associated (normalized) eigenfunctions of the covariance, *i.e.*, the solution to the integral equation

$$\int_{\Omega} C(\mathbf{x}, \mathbf{x}') \Phi(\mathbf{x}') d\mathbf{x}' = \lambda \Phi(\mathbf{x}), \quad \langle \Phi, \Phi \rangle_\Omega = 1. \quad (6)$$

The $\eta_\alpha(\theta)$ are the stochastic coordinates of U , having the property of being orthonormal:

$$\mathbb{E}[\eta_\alpha \eta_\beta] = \begin{cases} 1, & \alpha = \beta, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Owing to the structure of the covariance function, C , the eigenvalues are non-negative and can be ordered in decreasing magnitude, leading to a natural energy criterion for the truncation of the expansion. Classically, N is fixed as to satisfy some error tolerance in the $L_2(\Omega, \Theta)$ norm.

Specifically, one sets N so the following inequality is satisfied for some prescribed error tolerance $0 < \delta < 1$

$$\mathbb{E} \left[\|U - U_N\|_{\Omega}^2 \right] = \mathbb{E} \left[\|U\|_{\Omega}^2 \right] - \sum_{\alpha=1}^N \lambda_{\alpha} \leq \mathbb{E} \left[\|U\|_{\Omega}^2 \right] \delta^2. \quad (8)$$

Note that the process norm can be evaluated from the covariance function, $\mathbb{E} \left[\|U\|_{\Omega}^2 \right] = \int_{\Omega} C(\mathbf{x}, \mathbf{x}) d\mathbf{x}$, or through the whole KL spectrum: $\mathbb{E} \left[\|U\|_{\Omega}^2 \right] = \sum_{\alpha} \lambda_{\alpha}$.

Galerkin approximation In most cases, an analytical solution of (6) is not available, and we have to rely on a numerical method to approximate the eigenvalues and eigenfunctions. To this end, we choose a finite dimensional space \mathcal{V} , consisting of the linear span of a basis $\{v_1(\mathbf{x}), v_2(\mathbf{x}), \dots, v_Q(\mathbf{x})\}$, with basis functions $v_i \in L_2(\Omega)$. Let $\Phi^h(\mathbf{x}) = \sum_{k=1}^Q c_k v_k(\mathbf{x}) \in \mathcal{V}$ be the (finite dimensional) approximation and $\Phi(\mathbf{x})$. Using the approximate into Eq. 6 gives the *residual*

$$r(\mathbf{x}) \equiv \lambda \Phi^h(\mathbf{x}) - \int_{\Omega} C(\mathbf{x}, \mathbf{x}') \Phi^h(\mathbf{x}') d\mathbf{x}' = \sum_{k=1}^Q c_k \left(\lambda v_k(\mathbf{x}) - \int_{\Omega} C(\mathbf{x}, \mathbf{x}') v_k(\mathbf{x}') d\mathbf{x}' \right). \quad (9)$$

In the Galerkin method, the vector of coefficients $\mathbf{c} = (c_1, \dots, c_Q)^T$ is chosen by forcing $r(\mathbf{x})$ to be orthogonal to all functions in \mathcal{V} , *i.e.*:

$$\langle r, u \rangle_{\Omega} = 0, \quad \forall u \in \mathcal{V}. \quad (10)$$

Substituting (9) into (10) and exploiting the structure of \mathcal{V} , the problem can be recast as a generalized eigenvalue problem:

$$[K]\mathbf{c} = \lambda[M]\mathbf{c}, \quad (11)$$

where $[K]$ and $[M]$ are non-negative symmetric matrices of $\mathbb{R}^{Q \times Q}$ with entries

$$[K]_{ij} = \int_{\Omega} \int_{\Omega} C(\mathbf{x}, \mathbf{x}') v_i(\mathbf{x}') v_j(\mathbf{x}) d\mathbf{x}' d\mathbf{x}, \quad [M]_{ij} = \langle v_i, v_j \rangle_{\Omega}. \quad (12)$$

The dimension Q of the discrete generalized eigenvalue problem (11) depends on the dimension of the approximation space \mathcal{V} . If a Finite Element (FE) method is used to discretize the problem, the dimension of the basis is equal to the number of degrees of freedom Q of the FE space, which is determined by the number and type of elements used. Depending on the covariance structure, a very fine discretization (*i.e.*, very large Q) might be required to accurately represent the eigenfunctions and, as a result, solving the eigenvalue problem could become computationally intensive. This prevents the use of direct methods for most practical problems with large Q . As one is generally not interested in the full KL decomposition of U but in the N -dimensional dominant subspace, iterative methods (*e.g.*, sub-space iterations, Arnoldi, Lanczos; see for instance [6]) can be considered as an alternative for the resolution of (11). However, although for typical finite element discretizations the mass matrix $[M]$ is sparse, most covariance functions induce a full matrix $[K]$, with possibly prohibitive memory requirements for its storage. Matrix-free iterative methods and parallel implementation can be used to overcome memory limitations and accelerate computations, but the full character of the operator $[K]$ induces large computational cost and important communication burdens that can severely impact the efficiency of these approaches.

In the following subsection, we introduce an alternative strategy suitable to the parallel computation of the KL decomposition. It uses a domain decomposition approach to conveniently distribute the computational load among several processors and recast (11) in a *reduced eigenvalue problem* having a size dictated by the targeted error tolerance rather than by the dimension of the underlying discretization space, *e.g.* Q .

2.2 Domain Decomposition approach

As illustrated in Figure 1, our approach starts by partitioning the domain Ω into D non-overlapping subdomains:

$$\bar{\Omega} = \overline{\bigcup_{d=1}^D \Omega_d}, \quad \Omega_i \cap \Omega_{j \neq i} = \emptyset. \quad (13)$$

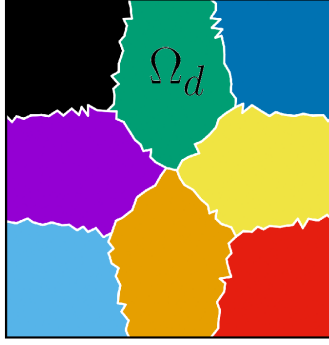


Figure 1: Partitioning of a square domain Ω into $D = 8$ non-overlapping subdomains.

Next, for each subdomain Ω_d , $d = 1, \dots, D$, we introduce the *local eigenmodes* $\tilde{\phi}_\beta^{(d)} : \Omega_d \rightarrow \mathbb{R}$, defined as the solutions of:

$$\int_{\Omega_d} C(\mathbf{x}, \mathbf{x}') \tilde{\phi}_\beta^{(d)}(\mathbf{x}') d\mathbf{x}' = \lambda_\beta^{(d)} \tilde{\phi}_\beta^{(d)}(\mathbf{x}), \quad \|\tilde{\phi}_\beta^{(d)}\|_{\Omega_d} = 1. \quad (14)$$

In (14), we have denoted by $\|\cdot\|_{\Omega_d}$ the natural restriction of the norm in $L_2(\Omega)$ to the subdomain Ω_d . It is seen that the $\tilde{\phi}_\beta^{(d)}$ are the eigenfunctions of the correlation $C(\mathbf{x}, \mathbf{x}')$ restricted to the d -th subdomain. These local eigenfunctions are extended to the global domain Ω by defining

$$\forall \mathbf{x} \in \bar{\Omega}, \phi_\beta^{(d)}(\mathbf{x}) = \begin{cases} \tilde{\phi}_\beta^{(d)}(\mathbf{x}), & \mathbf{x} \in \Omega_d, \\ 0, & \mathbf{x} \notin \Omega_d. \end{cases} \quad (15)$$

We observe that because the $\tilde{\phi}_\beta^{(d)}$ are orthonormal in Ω_d , in light of (15) we have

$$\langle \phi_\beta^{(d)}, \phi_{\beta'}^{(d')} \rangle_\Omega = \begin{cases} 1, & \text{if } d = d' \text{ and } \beta = \beta', \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

For each subdomain Ω_d we retain the $m_d > 0$ dominant eigenpairs according to the criterion discussed later in section 2.3. The D sets of dominant eigenfunctions are collected to form an orthonormal reduced basis \mathcal{B} of $L_2(\Omega)$:

$$\mathcal{B} = \bigcup_{d=1}^D \mathcal{B}_d, \quad \mathcal{B}_d = \{\phi_\beta^{(d)}, \beta = 1, \dots, m_d\}. \quad (17)$$

We denote by $\mathcal{V}_{\mathcal{B}}$ the linear span of \mathcal{B} . We then seek an approximation $\hat{\Phi} \in \mathcal{V}_{\mathcal{B}}$ of the global modes solution of (6), that is

$$\Phi(\mathbf{x}) \approx \hat{\Phi}(\mathbf{x}) = \sum_{d=1}^D \sum_{\beta=1}^{m_d} a_\beta^{(d)} \phi_\beta^{(d)}(\mathbf{x}). \quad (18)$$

We set $\mathbf{a}^{(d)} = (a_1^{(d)}, \dots, a_{m_d}^{(d)})^T$, the vector of the local coordinates of $\hat{\Phi}$ for $\mathbf{x} \in \Omega_d$. Applying the Galerkin method,

$$\forall \phi_\beta^{(d)} \in \mathcal{B} : \left\langle \int_{\Omega} C(\mathbf{x}, \mathbf{x}') \hat{\Phi}(\mathbf{x}') d\mathbf{x}', \phi_\beta^{(d)} \right\rangle_\Omega = \Lambda \langle \hat{\Phi}, \phi_\beta^{(d)} \rangle_\Omega, \quad (19)$$

the approximate eigenfunctions are seen to solve the following discrete eigenvalue problem,

$$\begin{bmatrix} [\hat{K}_{11}] & [\hat{K}_{12}] & \cdots & [\hat{K}_{1D}] \\ [\hat{K}_{21}] & [\hat{K}_{22}] & \cdots & [\hat{K}_{2D}] \\ \vdots & \vdots & \ddots & \vdots \\ [\hat{K}_{D1}] & [\hat{K}_{D2}] & \cdots & [\hat{K}_{DD}] \end{bmatrix} \begin{Bmatrix} \mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} \\ \vdots \\ \mathbf{a}^{(D)} \end{Bmatrix} = \Lambda \begin{Bmatrix} \mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} \\ \vdots \\ \mathbf{a}^{(D)} \end{Bmatrix}, \quad (20)$$

where the block matrices $[\hat{K}_{i,j}] \in \mathbb{R}^{m_i \times m_j}$ have for respective entries,

$$[\hat{K}_{i,j}]_{\alpha,\beta} = \int_{\Omega_i} \int_{\Omega_j} C(\mathbf{x}, \mathbf{x}') \phi_{\alpha}^{(i)}(\mathbf{x}) \phi_{\beta}^{(j)}(\mathbf{x}') d\mathbf{x} d\mathbf{x}', \quad 1 \leq \alpha \leq m_i, 1 \leq \beta \leq m_j. \quad (21)$$

We refer to (20) as the *condensed eigenvalue problem*. The dimension of this problem is

$$n_t = \sum_{d=1}^D m_d = \text{card } \mathcal{B}. \quad (22)$$

It is easily shown that the matrix $[\hat{K}] \in \mathbb{R}^{n_t \times n_t}$ is symmetric and positive definite if the covariance function is such that for all $u \in L_2(\Omega)$,

$$\|u\|_{\Omega} > 0 \Rightarrow \left\langle u, \int_{\Omega} C(\cdot, \mathbf{x}) u(\mathbf{x}) d\mathbf{x} \right\rangle_{\Omega} > 0. \quad (23)$$

This assumption is satisfied for most covariance functions, in particular for the whole Matérn class, and should not be restrictive in practice. Consequently, the n_t eigenvalues Λ_{α} of $[\hat{K}]$ can be ordered with decreasing magnitude as

$$\Lambda_1 \geq \Lambda_2 \geq \dots \geq \Lambda_{n_t} \geq 0. \quad (24)$$

Then, for reasons explained in section 2.3, we can select the smallest \hat{N} , $1 \leq \hat{N} \leq n_t$, such that for a prescribed relative error tolerance $0 \leq \delta \leq 1$ we have

$$\sum_{\alpha=\hat{N}+1}^{n_t} \Lambda_{\alpha} \leq \frac{\delta^2}{2} \sum_{\alpha=1}^{n_t} \Lambda_{\alpha}. \quad (25)$$

The truncated approximation of U is then given by

$$U(\mathbf{x}, \theta) \approx \hat{U}_{\hat{N}}(\mathbf{x}, \theta) \equiv \sum_{\alpha=1}^{\hat{N}} \sqrt{\Lambda_{\alpha}} \hat{\eta}_{\alpha}(\theta) \hat{\Phi}_{\alpha}(\mathbf{x}), \quad (26)$$

where

$$\hat{\Phi}_{\alpha}(\mathbf{x}) = \sum_{d=1}^D \sum_{\beta=1}^{m_d} a_{\alpha,\beta}^{(d)} \phi_{\beta}^{(d)}(\mathbf{x}). \quad (27)$$

is the eigenfunction corresponding to Λ_{α} .

As illustrated in the examples section, n_t is essentially fixed by the requested accuracy and not by the size of the discretization space. In fact, in our simulations, typical values for n_t were small enough to permit the use of direct solvers for the solution of the reduced problem. However, any other type of solver, for instance iterative ones, can be considered for the solution of (20). In the following, we refer to the approximation (26) as the *DD-KL expansion*. Algorithm 1 provides a schematic of the main steps involved in the DD-KL approach. The algorithm highlights the two main steps that can be carried out in parallel, namely the construction of the local bases \mathcal{B}_d and the assembly of the matrix $[\hat{K}]$ of the condensed operator. The actual parallel implementation of the method is further discussed and tested in section 4.

Remark A numerical method is needed to solve the local problems and yield the *local eigenpairs* (λ, ϕ) ; in the present work we use Finite Element methods to discretize the Galerkin weak form of (6). Even though we mentioned in section 2.1 that using this approach could become computationally intractable for large problems, using it to solve the local problems is now a viable option, owing to the much smaller size of the local problems. Specifically, their number of degrees of freedom is approximately divided by D as compared to the global problem. Moreover, the local modes over distinct subdomains can be computed *in parallel* exploiting the independence of the local problems; this allows to effectively distribute the computational load among several processors.

Algorithm 1: Schematic steps of the DD-KL approach.

```

Partition the domain  $\Omega$  into  $D$  subdomains ;
// Parallel loop - Computing local modes
foreach subdomain  $\Omega_d$  do
    Discretize the local integral equation given by (14) to get  $[K^{(d)}]$  and  $[M^{(d)}]$  ;
    Solve the local generalized eigenvalue problem  $[K^{(d)}]\{\tilde{\phi}^{(d)}\} = \lambda^{(d)}[M^{(d)}]\{\tilde{\phi}^{(d)}\}$ 
end foreach

// Parallel loop - Computing entries of reduced problem
foreach subdomain  $\Omega_i$  do
    foreach subdomain  $\Omega_j$  do
        foreach  $1 \leq \alpha \leq m_i$  and  $1 \leq \beta \leq m_j$  do
            Compute  $[\hat{K}_{i,j}]_{\alpha,\beta} = \int_{\Omega_i} \int_{\Omega_j} C(\mathbf{x}, \mathbf{x}') \phi_{\alpha}^{(i)}(\mathbf{x}) \phi_{\beta}^{(j)}(\mathbf{x}') d\mathbf{x} d\mathbf{x}'$ 
        end foreach
    end foreach
end foreach

Assemble and solve the reduced eigenvalue problem ;           /* see equation (20) */
Get approximated global eigenfunctions ;                       /* see equation (18) */

```

Remark The cost of solving the reduced eigenvalue problem is **independent** of the size of the discretization space used for the resolution of the local problem. Instead, it depends on the number of local modes retained in each subdomain, m_d , which add up to n_t ; we expect $n_t = \mathcal{O}(N)$, which is in practice much less than Q . The computational cost and complexity analysis of the method are discussed in section 4 on the basis of computational examples.

Remark Even if the local modes $\phi_{\beta}^{(d)}(\mathbf{x})$'s are all approximated using the same FE method¹ over each subdomain Ω_d , the condensed problem (20) does not ensure that the final approximation belongs to the FE space that would have been built over the whole domain Ω . For instance, a typical situation corresponds to the case of continuous FE approximations over each subdomain Ω_d to compute the $\phi_{\beta}^{(d)}$'s, so the $\hat{\Phi}_{\alpha}$ will generally be only piecewise continuous over the union of the Ω_d as indicated by (18), with jumps across the interfaces between subdomains. Such jumps are generally not a problem, but they can eventually be removed if needed at a post-processing stage, *e.g.* by one of the averaging procedures routinely used in Discontinuous Galerkin methods.

Remark Substituting (18) into (26) we get the following alternative representation of the DD-KL expansion directly in terms of the local eigenmodes

$$\hat{U}_{\hat{N}}(\mathbf{x}, \theta) = \sum_{d=1}^D \left[\sum_{\beta=1}^{m_d} \sqrt{\lambda_{\beta}^{(d)}} \xi_{\beta}^{(d)}(\theta) \phi_{\beta}^{(d)}(\mathbf{x}) \right], \quad \xi_{\beta}^{(d)}(\theta) = \sum_{\alpha=1}^{\hat{N}} \sqrt{\frac{\Lambda_{\alpha}}{\lambda_{\beta}^{(d)}}} a_{\alpha,\beta}^{(d)} \hat{\eta}_{\alpha}(\theta), \quad (28)$$

where the $\{\xi_{\beta}^{(d)}(\theta), \beta = 1, \dots, m_d\}$ are called the local random variables. This expression highlights that for $\mathbf{x} \in \Omega_d$ the process $U(\mathbf{x}, \theta)$ can be approximated using an expansion that depends only on the local coordinates $\xi_{1 \leq \beta \leq m_d}^{(d)}$. In other words, even though the stochastic dimension of the truncated expansion is n_t , at the subdomain level we can characterize the process using a reduced stochastic dimension m_d . In addition, it can be shown that the local coordinates of a subdomain are uncorrelated but that coordinates of distinct subdomains are generally correlated. This representation of the process in terms of local random variables is exploited in the sequel to this paper (Part B) to accelerate the solution of stochastic elliptic PDEs using a Monte-Carlo approach.

¹In fact, distinct subdomains could be treated using different discretization methods.

2.3 Truncation strategy

In this section we address the selection of the parameters governing the proposed method. An obvious question concerns the selection of the number D of subdomains and the actual partition of Ω . In the example section 3.3, we numerically illustrate the robustness of the method with respect to the partitioning of Ω in D subdomains and investigate the effects of varying D . Here, we focus on the selection of the number of local modes m_d and the appropriate truncation of the final DD-KL expansion, for a fixed partition of Ω .

The method introduces two different sources of error in the approximation of U by $\hat{U}_{\hat{N}}$. First, an error is introduced when representing the eigenfunctions of U in the finite dimensional space $\mathcal{V}_{\mathcal{B}}$ built on the local bases, see (18). Hereafter, we shall denote by $U_{\mathcal{B}}$ the projection of U on $\mathcal{V}_{\mathcal{B}}$:

$$U_{\mathcal{B}}(\mathbf{x}, \theta) = \sum_{d=1}^D \sum_{\alpha=1}^{m_d} \sqrt{\lambda_{\alpha}^{(d)}} \eta_{\alpha}^{(d)}(\theta) \phi_{\alpha}^{(d)}(\mathbf{x}). \quad (29)$$

Second, the projected process $U_{\mathcal{B}}$ is further reduced, through the resolution of the reduced problem, to yield the final approximation $\hat{U}_{\hat{N}}$.

Because $U - U_{\mathcal{B}}$ is orthogonal to $U_{\mathcal{B}} - \hat{U}_{\hat{N}}$, the squared norm of the error $U - \hat{U}_{\hat{N}}$ can actually be broken down into two independent parts as follows:

$$\mathbb{E} \left[\|U - \hat{U}_{\hat{N}}\|_{\Omega}^2 \right] = \mathbb{E} \left[\|U - U_{\mathcal{B}}\|_{\Omega}^2 \right] + \mathbb{E} \left[\|U_{\mathcal{B}} - \hat{U}_{\hat{N}}\|_{\Omega}^2 \right]. \quad (30)$$

See appendix A for the derivation. The first term is obtained by adding up the local contributions over the subdomains, which, by construction of the local modes, are given by

$$\epsilon_d^2 \equiv \mathbb{E} \left[\|U - U_{\mathcal{B}}\|_{\Omega_d}^2 \right] = \mathbb{E} \left[\|U\|_{\Omega_d}^2 \right] - \sum_{\alpha=1}^{m_d} \lambda_{\alpha}^{(d)}, \quad \forall d = 1, \dots, D. \quad (31)$$

Then, gathering the local contributions, we end up with

$$\epsilon_{\mathcal{B}}^2 \equiv \mathbb{E} \left[\|U - U_{\mathcal{B}}\|_{\Omega}^2 \right] = \sum_{d=1}^D \epsilon_d^2 = \mathbb{E} \left[\|U\|_{\Omega}^2 \right] - \sum_{d=1}^D \sum_{\alpha=1}^{m_d} \lambda_{\alpha}^{(d)}. \quad (32)$$

Finally, similarly to the classical KL truncation error, the second error contribution is

$$\epsilon_{\mathcal{B}\hat{N}}^2 \equiv \mathbb{E} \left[\|U_{\mathcal{B}} - \hat{U}_{\hat{N}}\|_{\Omega}^2 \right] = \mathbb{E} \left[\|U_{\mathcal{B}}\|_{\Omega}^2 \right] - \sum_{\alpha=1}^{\hat{N}} \Lambda_{\alpha}. \quad (33)$$

Since $\mathbb{E} \left[\|U_{\mathcal{B}}\|_{\Omega}^2 \right] \leq \mathbb{E} \left[\|U\|_{\Omega}^2 \right]$, the overall error can be estimated from:

$$\mathbb{E} \left[\|U - \hat{U}_{\hat{N}}\|_{\Omega}^2 \right] = \epsilon_{\mathcal{B}}^2 + \epsilon_{\mathcal{B}\hat{N}}^2 \leq 2\mathbb{E} \left[\|U\|_{\Omega}^2 \right] - \sum_{d=1}^D \sum_{\alpha=1}^{m_d} \lambda_{\alpha}^{(d)} - \sum_{\alpha=1}^{\hat{N}} \Lambda_{\alpha}. \quad (34)$$

This expression shows that to reduce the error, one needs to jointly increase the size of the local basis over *all* the subdomains *and* increase \hat{N} . Clearly, this suggests the existence of an optimal (in terms of computational efficiency) set of values for m_d and \hat{N} . In this work, in order to achieve an overall relative error $0 \leq \delta \leq 1$, we simply enforce the following error levels. First, regarding the local errors, we require simply that the m_d 's are selected to ensure for each d

$$\epsilon_d^2 = \mathbb{E} \left[\|U\|_{\Omega_d}^2 \right] - \sum_{\alpha=1}^{m_d} \lambda_{\alpha}^{(d)} \leq \mathbb{E} \left[\|U\|_{\Omega_d}^2 \right] \frac{\delta^2}{2}, \quad (35)$$

such that $\epsilon_{\mathcal{B}}^2 \leq \mathbb{E} \left[\|U\|_{\Omega}^2 \right] \delta^2/2$. Then, \hat{N} is selected so that (25) holds, which ensures that

$$\mathbb{E} \left[\|U - \hat{U}_{\hat{N}}\|_{\Omega}^2 \right] \leq \delta^2 \mathbb{E} \left[\|U\|_{\Omega}^2 \right]. \quad (36)$$

Remark We observe that the selection of the m_d 's in (35) balances the local relative (squared) error over the subdomains; this could be further improved by collecting the local eigenvalues for all the subdomains and selecting the n_t dominant ones that ensure $\epsilon_{\mathcal{B}}^2 \leq \delta^2/2$. However, numerical experiments have shown that the local selection based on (35) is quite satisfactory and in addition it maintains a strict independence of the local bases construction.

3 Numerical Examples

In this section we describe the test problem used to validate the domain decomposition approach (Section 3.1), present several numerical results that demonstrate the convergence and the error control (Section 3.2) of the method. Finally, we provide in Section 3.3 a brief analysis of the impact of the domain partitioning on the behavior of the method.

3.1 Test problem

The method is applied to the decomposition of a second-order stochastic process $U(\mathbf{x}, \theta)$ defined over the two-dimensional unit square $\Omega = [0, 1]^2$. For the covariance of the process, we assume the classical stationary squared exponential structure, with characteristic correlation length L and unit variance:

$$C(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|_2^2/L^2\right). \quad (37)$$

We stress that the proposed method is not limited to this specific covariance structure and domain shape. For the spatial discretization of the KL modes, we use piecewise constant Finite Element discretization unless otherwise indicated. This low order FE approximation space is chosen because of the application to stochastic elliptic PDEs we have in mind; in these applications, the stochastic coefficient field modeled by KL expansion is often considered as constant over the finite elements [4]. In any case, we stress that our method perfectly accommodate for higher order FE methods. The spatial mesh consists of a conforming triangulation of Ω into a set Σ of N_e finite elements. Eventually, the set of elements is partitioned into D subsets forming connected non-overlapping subdomains Ω_d , for $d = 1, \dots, D$. For the partitioner, we rely on a k -means geometrical clustering algorithm [17, 15, 18].

The correlation length L has a marked effect on the spectral decay of the KL expansion. Figure 2 shows the magnitude of the ordered eigenvalues of $C(\mathbf{x}, \mathbf{x}')$. From the Figure, we can appreciate the slower decay rate of λ_k for smaller values of L . In fact, for the smallest correlation lengths tested, the leading eigenvalues are seen to be essentially equal with an asymptotic decay that is increasingly delayed as L decreases. The slower the decay rate, the larger the number of terms required in the KL expansion to achieve a desired accuracy. For instance, setting $\delta^2 = 10^{-3}$ in equation (8) we get $N = 3$, $N = 8$, $N = 259$, and $N = 22,356$ for $L = 10$, $L = 1$, $L = 0.1$, and $L = 0.01$, respectively.

The DD-KL approach exploits the fact that the convergence behavior of the KL expansion, for fixed covariance structure, is governed by the magnitude of L relative to the characteristic length of the domain. Indeed, decreasing (resp. increasing) the characteristic extent of the domain has a similar effect as increasing (resp. decreasing) the correlation length. In the DD-KL approach, increasing the number of subdomains allows to reduce the extent of the subdomains, leading to an apparent larger L and a faster spectral decay for the local expansions. This is illustrated in Figure 3, where the spectra of local expansions are shown for different values of the number of subdomains. Note that the Figure reports the spectra for all the D subdomains, so there are D spectra plotted when Ω is partitioned into D subdomains. Here, we observe that for a given number D of subdomains, the local subdomains all have roughly the same extent so the local expansions have a similar decay. Moreover, it is seen that, as expected, the local expansions have spectra which decay faster as D increases.

The variability of the spectra among the D subdomains can be appreciated from Figure 3b. This variability is principally due to the partitioning procedure that generates non identical subdomains, with slightly variable apparent L as a result.

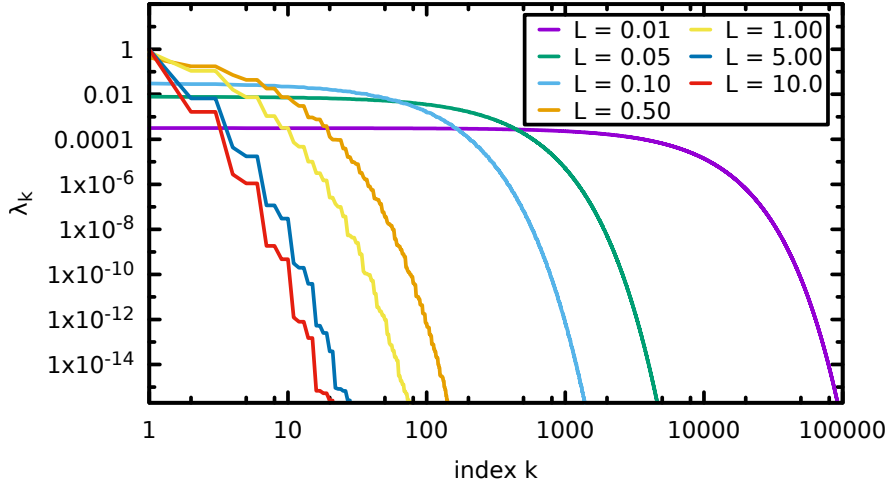


Figure 2: Spectra of the global decomposition for the squared exponential covariance in (37) with different values of the correlation length L as indicated.

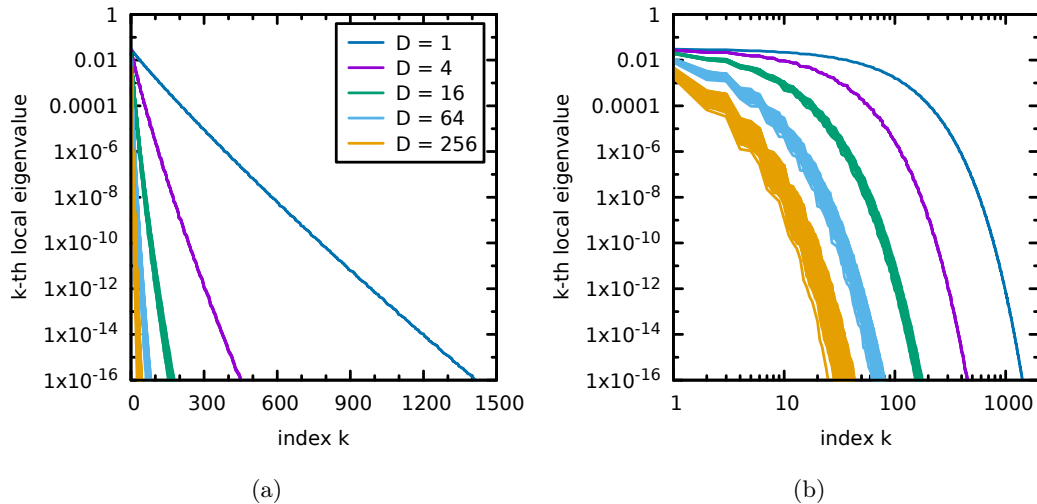


Figure 3: Spectra of local decompositions for the squared Gaussian covariance (37), with $L = 0.1$, and different D as indicated. The plot in (a) uses a linear scale for the x -axis, whereas the plot in (b) uses a logarithmic scale.

3.2 Convergence analysis

We start by demonstrating that the proposed method converges towards the direct decomposition estimates of the dominant or truncated decomposition of U . To this end, we select arbitrarily truncation levels $N > 0$ and check that the DD-KL solution \hat{U}_N converges to the truncated direct solution U_N (*i.e.*, the solution computed without the domain decomposition), as the error tolerance is lowered. In fact, the comparison and convergence analyses are conducted using the spatially discretized solutions, over the same finite element mesh. We compute U_N solving (6) discretized over the finite element mesh, as described in Section 2.1. The same finite element mesh with $N_e = 40,802$ elements and piecewise constant approximation is used for both decompositions. In this case, the differences between \hat{U}_N and U_N result from the DD-KL reduction error only. To measure these differences, we define the following error metrics. First, we consider error in the spectra, computing the normalized ℓ_1 -distance between the N dominant

eigenvalues of the DD-KL solution (Λ_k) and of the direct computation (λ_k) :

$$\epsilon_{\text{spec}} = \frac{\sum_{k=1}^N |\lambda_k - \Lambda_k|}{\sum_{k=1}^N |\lambda_k|}. \quad (38)$$

In addition to the error in spectra, ϵ_{spec} , we quantify the distance between the N -dimensional subspaces of the direct and DD-KL eigen bases. Different error measures can be thought to characterize the distance between subspaces; here we simply rely on the expected squared L_2 -norm of the projection error in the linear span of the eigenvectors $\hat{\Phi}_\alpha(\mathbf{x})$ of the DD-KL approximation. For a generic second order process V , the (relative) subspace error measure $\epsilon_{\text{sub}}^2(V)$ is defined as follows:

$$\epsilon_{\text{sub}}^2(V) = \frac{\mathbb{E} \left[\left\| V(\mathbf{x}, \theta) - \sum_{\alpha=1}^N \langle V(\mathbf{x}, \theta), \hat{\Phi}_\alpha(\mathbf{x}) \rangle_{\Omega} \hat{\Phi}_\alpha(\mathbf{x}) \right\|_{\Omega}^2 \right]}{\mathbb{E} \left[\|V(\mathbf{x}, \theta)\|_{\Omega}^2 \right]}. \quad (39)$$

Note that $\epsilon_{\text{sub}}^2(U_N)$ is therefore the spatially discretized version of the normalized projection error $\epsilon_{\mathcal{B}}$ in (32) for the truncated process U_N . In order to compute the expectation in the previous error definition, the process U is assumed to be Gaussian in what follows.

For N and D fixed, we expect the difference between U_N and \hat{U}_N to decrease as richer and richer local bases are considered. Recalling that the dimension of the local basis for subdomain Ω_d is m_d , we expect $\hat{U}_N \rightarrow U_N$, in the sense of the metrics defined above, when m_d increases for *all* the subdomains. As explained before, the local basis dimension m_d yields a local expansion error ϵ_d given by equation (31). In the following, we choose a positive tolerance parameter δ^* to control the error in the local expansions and we select the smallest $m_d \in \mathbb{N}_+$ satisfying

$$\epsilon_d^2 \leq \mathbb{E} \left[\|U\|_{\Omega_d}^2 \right] \times \delta^*, \quad \forall d = 1, \dots, D. \quad (40)$$

Figures 4 and 5 report the errors ϵ_{spec} and ϵ_{sub}^2 , respectively, as a function of the tolerance parameter δ^* . The two figures show that one can improve the accuracy of the KL-DD expansion and approximate better and better U_N by lowering the tolerance δ^* , that is, by increasing the size of the local bases and accordingly enriching the reduced space. In the figures, a dashed line with unit slope is shown to serve as a visual reference corresponding to $\epsilon_{\text{spec}} = \delta^*$ and $\epsilon_{\text{sub}}^2 = \delta^*$, respectively. We also note that as larger N are considered, δ^* has to decrease consistently, because increasing N requires $n_t = \sum_d m_d \geq N$: a minimal tolerance δ^* is therefore required to induce a reduced basis \mathcal{B} large enough ($\text{card } \mathcal{B} > N$) when selecting the m_d through (40). This explains the different ranges for δ^* considered in the plot, depending on N . Another interesting

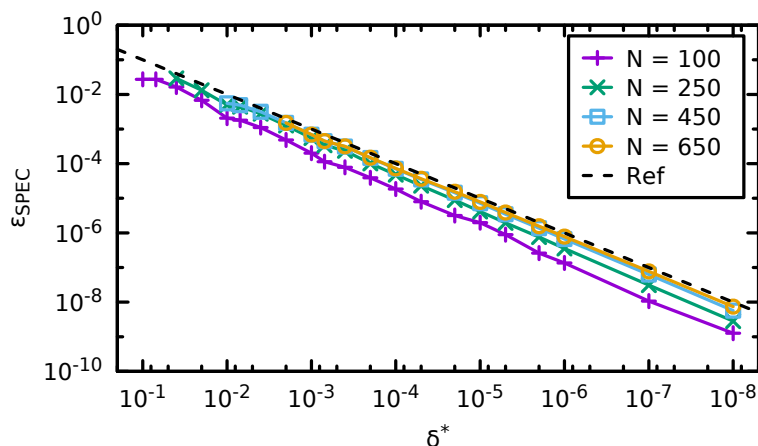


Figure 4: Relative spectrum error ϵ_{spec} , between U_N and \hat{U}_N , as a function of the local tolerance δ^* (see (40)) for different subspace dimensions N as indicated. Computations use $D = 80$ and $L = 0.1$. The dashed line has unit slope.

observation from Figures 4 and 5 is that the two errors decay proportionally with the local

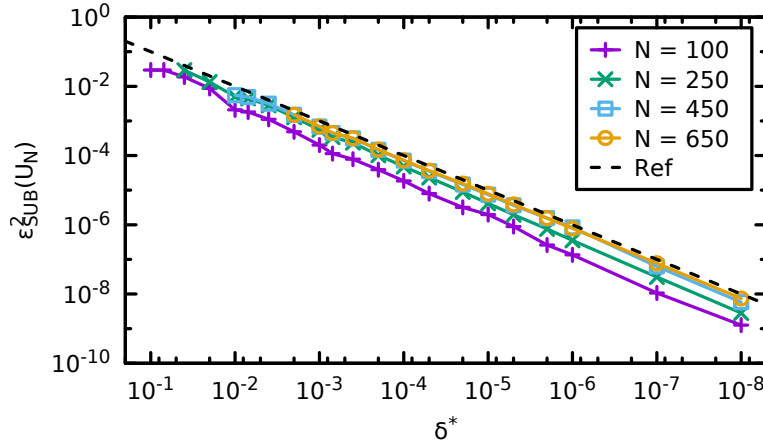


Figure 5: Subspace error $\epsilon_{\text{sub}}^2(U_N)$ in (39) as a function of the local tolerance δ^* (see (40)) for different subspace dimensions N as indicated. Computations use $D = 80$ and $L = 0.1$. The dashed line has unit slope.

tolerance δ^* . This trend demonstrates that the local accuracy fixes the global precision on U_N , for a fixed N and δ^* small enough. Conversely, for a fixed and small enough δ^* , the relative error ϵ_{sub}^2 increases as we demand to approximate more and more modes (increasing N) but always remains less than δ^* . Note that the error is always below the dashed line, which illustrates that we can effectively bound the error in the approximation to be less than any desired value δ^* .

Now that we have established that the truncated DD-KL solution \hat{U}_N converges to the truncated process U_N when the tolerance on the local error is lowered, we proceed to investigate the error between \hat{U}_N and the non-truncated process U . Strictly speaking, we actually look at the differences between the spatially discretized versions of U and \hat{U}_N . Also, to compute efficiently the subspace error $\epsilon_{\text{sub}}^2(U)$ in (39), we need either to directly sample realizations of U or to know its (untruncated) KL expansion. Here, however, we use as a proxy for U a truncated KL expansion of U , obtained with the direct method, with a very large number of modes, typically several times larger than the requested N . In Figure 6, the evolution of $\epsilon_{\text{sub}}^2(U)$ is shown as a function of the local tolerance δ^* , for different values of N . These curves must be compared with the errors with respect to U_N plotted in Figure 5. Contrary to the error with respect to U_N , the error with respect to U is seen to level off and not to continue to decay when the tolerance δ^* is lowered. This behavior reflects the decomposition of the error in (30): when the projection error $\epsilon_{\mathcal{B}}$ is made smaller and smaller, lowering δ^* , the truncation error $\epsilon_{\mathcal{B}\hat{N}}$ becomes dominant such that $U - \hat{U}_N$ can not be reduced further but by increasing N . It confirms that increasing the dimension of the local bases beyond a certain threshold does not provide any appreciable advantages. On the contrary, it requires determining more local modes and solving a larger reduced problem with higher numerical cost as a result. The plots also verify the appropriateness of the strategy described in Section 2.3, since we can observe that for a given N the relative subspace error stagnates after $\delta^* \approx \epsilon_{\text{sub}}^2(U)$.

So far, we have kept the number of subdomains constant in analyzing the error in the DD-KL solution. We now show that the proposed error-control approach is not compromised when we increase the number of subdomains D used to partition Ω . In the following numerical test, we fix a target relative error δ and select the local and global truncation level using equations (25) and (35), to balance the projection ($\epsilon_{\mathcal{B}}^2 \leq \frac{\delta^2}{2}$) and truncation ($\epsilon_{\mathcal{B}\hat{N}}^2 \leq \frac{\delta^2}{2}$) errors. The DD-KL solution is then computed for different numbers D of subdomains and we report in Figure 7 the resulting relative projection error $\epsilon_{\text{sub}}^2(U)$. The curves demonstrate that a relative error less than δ is consistently achieved irrespective of the number of subdomains considered.

The numerical experiments on the convergence of the DD-KL method were repeated using continuous piecewise quadratic finite element discretizations (P_2). We obtained results that are similar to the piecewise constant case and, therefore, they are not reported here. Note, however, that the discretized DD-KL and direct solutions lives in different finite dimensional space for the quadratic approximation, as the direct solution is continuous over the whole domain Ω , when

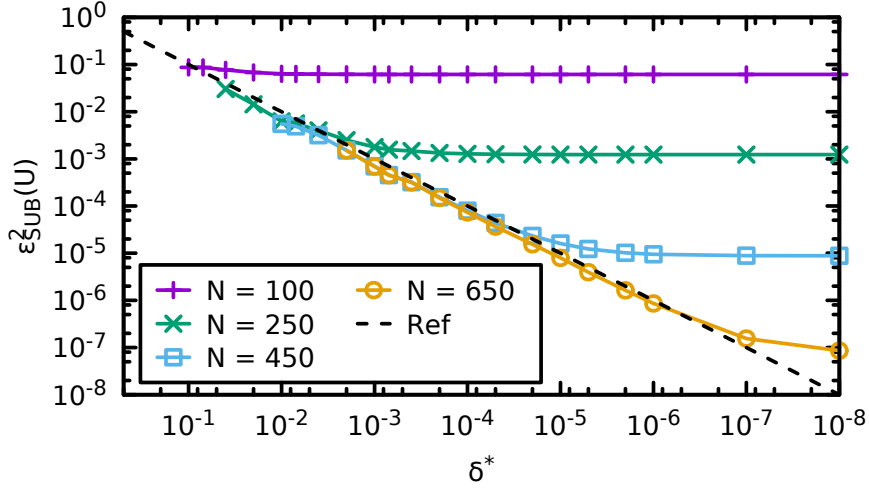


Figure 6: Projection error $\epsilon_{\text{sub}}^2(U)$ with respect to the untruncated process, as a function of tolerance δ^* defining the local truncation through (40). Computations are conducted for $L = 0.1$ using $D = 80$ subdomains and different values of N as indicated. The dashed black line is a reference of unit slope.

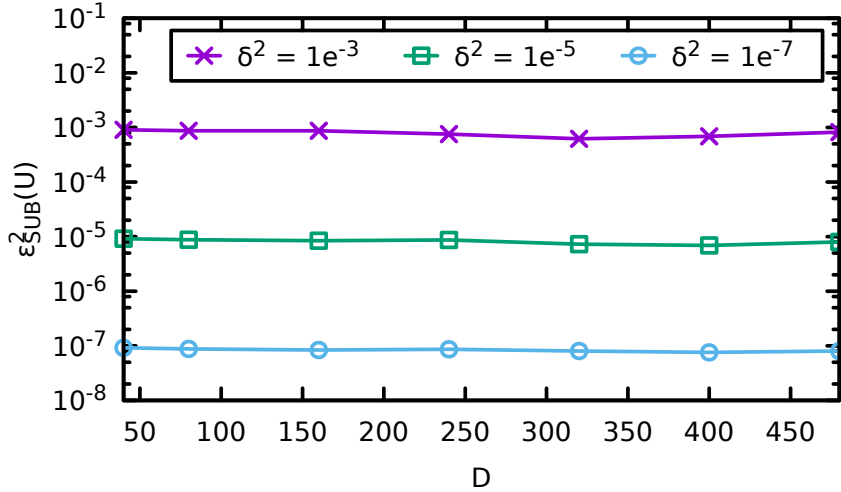


Figure 7: Projection error $\epsilon_{\text{sub}}^2(U)$ as a function of the number of subdomains D and for different targeted relative accuracy δ as indicated.

the DD-KL solution is only continuous over the subdomains Ω_d , with possibly discontinuities across the boundaries between subdomains. Our numerical tests indicates that the DD-KL approach remains convergent for finite element discretization with order greater than 0, and is not compromised as D increases. Of course, changing the order of the FE method affects the error in the computed eigenmodes of the local problems, such that a coarser mesh with a lower number of degrees of freedom Q could be considered for the P_2 FE discretization, compared to the P_1 or P_0 discretizations, while maintaining a comparable accuracy in the final approximations. The present work is not concerned with the question of determining the optimal FE order and mesh size to achieve a prescribe accuracy at a minimal computational cost. This question is left as a subject to future studies.

3.3 Effect of domain partitioning

We have just seen that changing the number of subdomains does not affect the accuracy of the final DD-KL expansion, provided that the truncations of the local and final expansions are properly set. However, the number of subdomains is obviously expected to affect the computational cost, as the DD-KL method was motivated by the complexity reduction in the first place. The analysis of the computational efficiency of DD-KL method is delayed to the next section, where a parallel implementation is presented. In this section, we focus on the behavior of the local and reduced bases when D varies and depending on the type of partitioning of Ω considered.

We first assume that Ω is partitioned into D subdomains having similar size. Following the discussion above, the change in apparent L is proportional to $D^{1/n}$, for n spatial dimensions. In addition to the reduction of the dimensionality of the discretized local problems, we also expect the size of the local bases to decrease for larger D . However, the spectra shown in Figure 3a indicate that the reduction in the average (over the D subdomains) number of local modes, \bar{m} , becomes marginally insignificant as D increases. In any cases, we have the lower bound $m_d \geq 1$, indicating that the behavior of the method for very large D (compared to N) can be problematic. To clarify this point, we report in Table 1 the evolution of the reduced problem size (n_t), average local basis dimension (\bar{m}) and standard deviation of m_d (denoted σ_{m_d}), when increasing the numbers of subdomains. The results reported correspond to the previous example, with a target accuracy $\delta^2 = 2 \times 10^{-3}$. We observe that as D increases the dimension n_t of the reduced problem increases. In fact, it is expected that asymptotically we would have $n_t = \mathcal{O}(Q)$, where Q is the number of degrees of freedom. This claim is supported by the evolutions of the averaged local basis dimension \bar{m} which is seen to initially drop quickly and then continues to drop at a lower rate. For the range of D investigated the smallest value we reached was $\bar{m} = 3$, but if D continues to increase we will eventually converge to $\bar{m} = 1$. Similarly, the standard deviation of the local basis dimension m_d is reported to decay monotonically to zero. Since, for a fixed target accuracy δ we have roughly fixed number of term N in the expansion, the evolution of n_t with D indicates the existence of an optimal number of subdomains, balancing the complexity reduction in the local problems with the progressive increase in the size n_t of the reduced problem. Determining this optimum is not obvious as it depends on implementation (*e.g.* how many local problems can be solved in parallel) and on the numerical method used to solve the reduced problem. Regardless of these considerations, we remark that, over the range of values for D shown in Table 1, the KL decomposition problem is recast in a reduced one that has a dimension ranging from $n_t = 431$ to 3,840, indicating a significant complexity reduction compared to the direct approach. For example, using intermediate value $D = 160$, the DD-KL approach involves the solution of 160 independent eigen problems (possibly in parallel) with size roughly 40,802/160 = 255 and the resolution of the reduced problem with size $n_t = 983$; these have to be compared with the direct approach in which one has to solve a single eigen problem with dimension 40,802.

D	n_t	$\bar{m} \pm \sigma_{m_d}$
20	431	21.55 \pm 1.43
40	542	13.55 \pm 0.59
80	741	9.26 \pm 0.56
160	983	6.14 \pm 0.35
320	1682	5.26 \pm 0.44
640	2,306	3.60 \pm 0.53
1280	3,840	3.00 \pm 0.00

Table 1: Progression of n_t for different values of D with $\delta^2 = 2 \times 10^{-3}$ and $L = 0.1$.

It should be clear at this point that the efficiency of the DD-KL method depends on the behavior of the local problems when the domain is partitioned into smaller ones. It is then important to investigate the impact of the partitioning method on this behavior. There are several approaches available to decompose Ω into D non-overlapping subdomains Ω_d . We briefly mention two popular families: graph partitioning and k -means clustering. The graph partitioning methods convert the mesh into a dual graph, in which the vertices are the mesh elements, and edges link

neighboring elements. The graph is then partitioned as to balance the number of vertices in each partition, while minimizing the number of edges that straddle different partitions. The k -means clustering [17, 8, 9, 15] consists in partitioning data points into k clusters, where each data point is assigned to the cluster whose mean (or centroid) is the nearest. In the context of domain decomposition, the data points to be partitioned correspond to the mesh element centroids or nodes. In this section, we rely on the k -means clustering method and consider different distance functions to control the geometrical properties of the subdomains Ω_d . Specifically, we consider an anisotropic Euclidean distance,

$$\Delta_d(\mathbf{x}, \mathbf{x}') = \sqrt{(x - x')^2 + \rho(y - y')^2}, \quad (41)$$

to control with $\rho > 0$ the aspect-ratio of the Ω_d ; and with $R > 1$ control the dispersion of the subdomain sizes by using coefficients r_d , drawn uniformly in $[1, R]$, to scale the distance to the d -th centroid. Figure 8 shows several partitions of Ω into $D = 20$ subdomains using different values of ρ and R . It is seen that as $\rho > 1$ increases (from left to right) the subdomains are more and more stretched horizontally. Further, for $R = 1$ (top row) the subdomains have all similar size (surface), in contrast to the case with $R = 10$ (bottom row) where a significant dispersion of the subdomains size is seen.

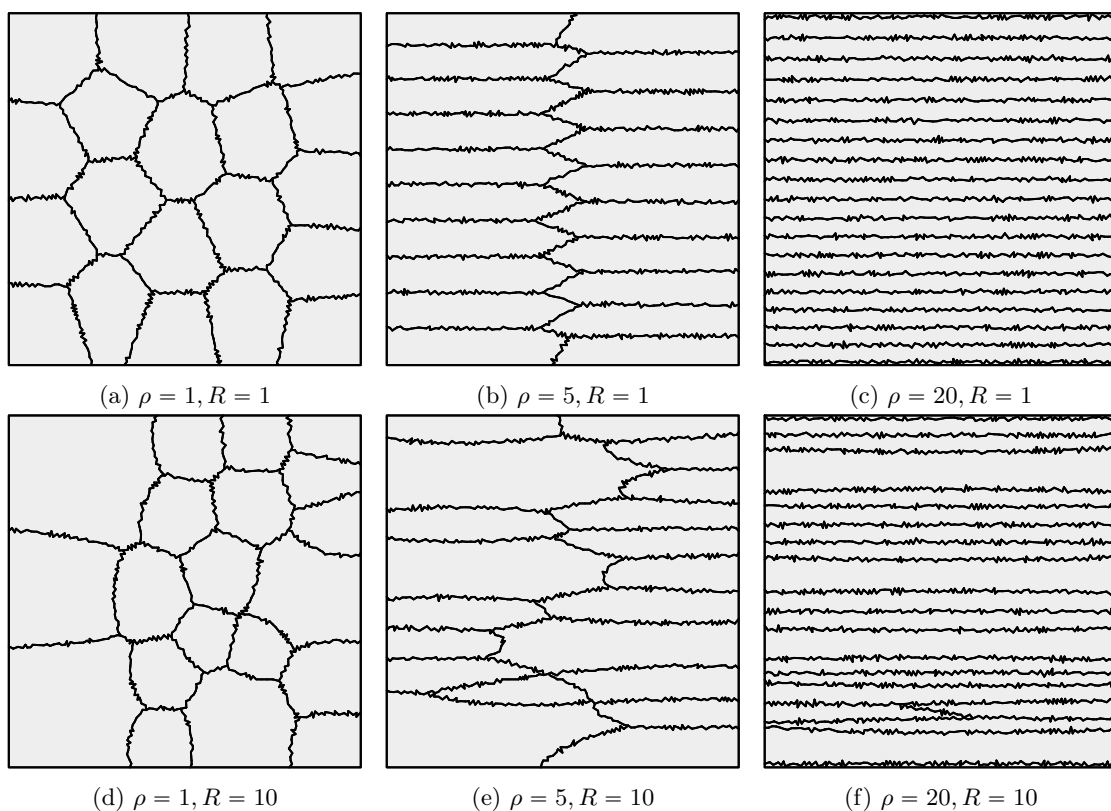


Figure 8: Partitioning of $\Omega = (0,1)^2$ into $D = 20$ subdomains with the k -means clustering method, using different parameter values of ρ and R as indicated.

Before investigating the effects of the partition on the DD-KL method, we first check in Figure 9 that changing the geometry of the subdomains does not affect the accuracy of the method and that the selection of the local and global truncations remains appropriate. The plots confirm that, for all partitions tested, the method achieves the desired accuracy set with $\delta^2 = 2 \times 10^{-3}$.

Plotted in Figure 10 are the evolutions of the reduced problem size, n_t , as a function of the anisotropy parameter ρ , for different values of R and D . Consistently with the results reported in Table 1, n_t increases with D when R and ρ are held fixed. In addition, an effect related to the aspect-ratio of the subdomains is evidenced in Figure 10. Specifically, the size of the reduced problem is adversely affected for stretched subdomains. This effect was anticipated from the

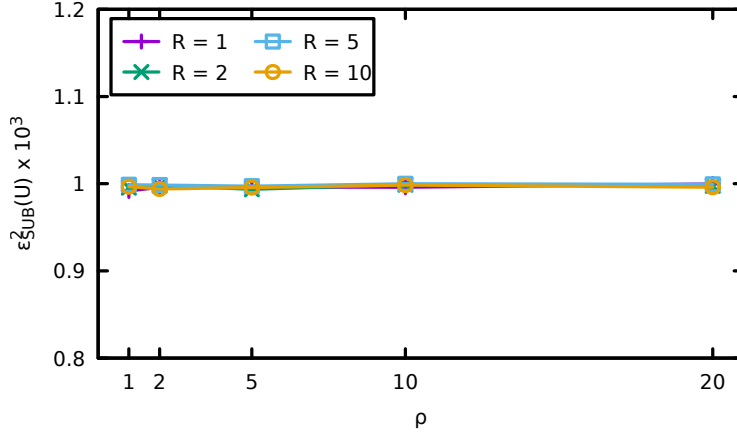


Figure 9: $\epsilon_{\text{sub}}^2(U)$ versus ρ for a target accuracy $\delta^2 = 2 \times 10^{-3}$. The domain is partitioned into $D = 80$ subdomains using the k -means clustering with parameters ρ and R as indicated.

trends and behaviors reported above: the most effective reduction of the local bases is expected for subdomains that have well balanced size in all directions. Incidentally, this behavior means that, in the case of anisotropic covariance functions, one would have interest in designing a partition of Ω that fits with the principal directions of C . We remark, however, that the effect of stretched subdomains is not so severe, with a reported increase of less than 50% in the reduced problem size when going from 1:1 to 1:20 aspect-ratios. Finally, we observe from Figure 10 that increasing the dispersion of the subdomains size (increasing R) has a negligible effect on n_t . This can be explained by compensation effects between larger m_d for large subdomains and lower m_d for smaller ones. Having n_t roughly independent of R does not necessarily translate into a constant computational cost. In fact, having subdomains with very different sizes could induce severe load balancing issues for the parallel resolution of the local problems. On the positive side, being able to tune the geometric size of the subdomains, without impacting much n_t , means that one can eventually adapt the partition in the case of a non-uniform spatial discretization.

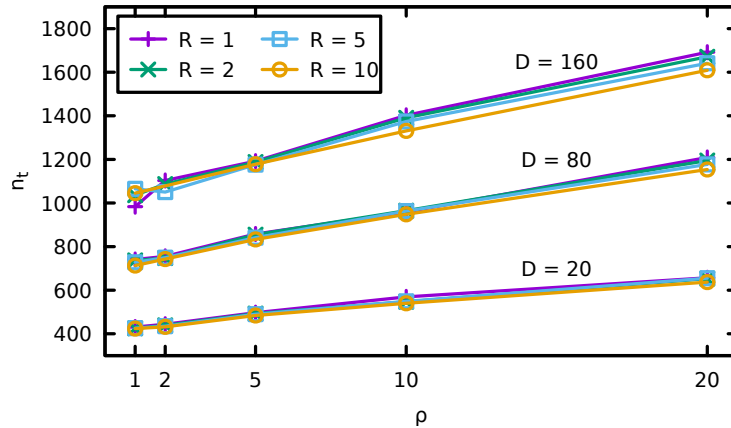


Figure 10: n_t versus ρ for a target accuracy $\delta^2 = 2 \times 10^{-3}$. Curves are generated for different values of D and R as indicated.

4 Performance Analysis

This section is dedicated to assessing the serial behavior of our approach, as well as its parallel scalability and efficiency.

4.1 Serial behavior

We first investigate the influence of the number of subdomains D on the computational time of the method in a serial implementation. For the reasons discussed later in Section 4.2, we only report the behavior of the local solves (see Eq. (14)) and the assembly of the reduced problem, *i.e.*, the filling of the block matrices (see Eq. (20)). Figure 11 depicts the dependence of the computational times of these stages on the number of subdomains. We also compare the case of a discretization with either piecewise constant (P_0) and piecewise quadratic (P_2) finite elements in Figures 11a and 11b, respectively. The computational times are reported in arbitrary units.

The first observation that can be drawn from these figures is that the two computational times decrease as the number of subdomains increases, with a faster decay for the local solves than the assembly of reduced problem, and the same behavior is reported for the two finite element discretizations. Due to the faster decay of the local solves CPU time, the reduced problem assembly time becomes dominant as D increases, and the combined time becomes quickly dominated by the matrix fill.

It is easy to understand that solving the local problems becomes cheaper as the number of subdomains increases: as D increases, the subdomains become smaller, and thus the local problems involving fewer unknowns are cheaper to solve. In the tests presented, a direct solver was used for the local problem, so the computational time reduction is very significant. Another important advantage of the proposed approach stems from the independence of the local solves. While solving the global generalized eigenvalue problem (11) would require the evaluation and possibly the storage of the full stiffness matrix $[K]$ in (12), the local problems only involve stiffness matrices $[K^{(d)}]$ that correspond to pairs of points $(\mathbf{x}, \mathbf{x}')$ belonging to a same subdomain Ω_d . In other words, the set of local matrices $[K^{(d)}]$ simply correspond to the diagonal, subdomain-based blocks of the global matrix $[K]$. This can result in significant CPU time and memory savings as compared to solving the global problem (11).

The reason for the reduction of the computational time for the reduced problem assembly is less obvious, but can be explained as follows. A given block $[\hat{K}_{i,j}]$ has $m_i \times m_j$ entries, each one requiring an integration over $\Omega_i \times \Omega_j$, with complexity proportional to $\text{Ne}_i \times \text{Ne}_j$ (Ne_d being the number of degrees of freedom of the approximations in Ω_d). If the subdomains have roughly the same numbers of local modes ($m_i \approx m_j \approx \bar{m}$) and unknowns ($\text{Ne}_i \approx \text{Ne}_j \approx \text{Ne}/D$), and given that there are $D(D+1)/2$ such blocks to be computed (accounting for the symmetric structure), the complexity in assembling the reduced matrix $[\hat{K}]$ can be estimated to be proportional to $\text{Ne}^2 \bar{m}^2 \times (D+1)/D$. In Section 3, we have seen that the average number of local modes $\bar{m} = n_t/D$ tends to decrease with D (see Table 1), although n_t increases. Consequently, the complexity of the reduced problem assembly reduces with D though it is seen to progressively level-off for very large number of subdomains. Eventually, as m_d can not be less than 1, increasing further D would have a negative impact on the overall efficiency, because though the assembling stage may not be penalized, the resolution cost of the reduced problem would continuously increase.

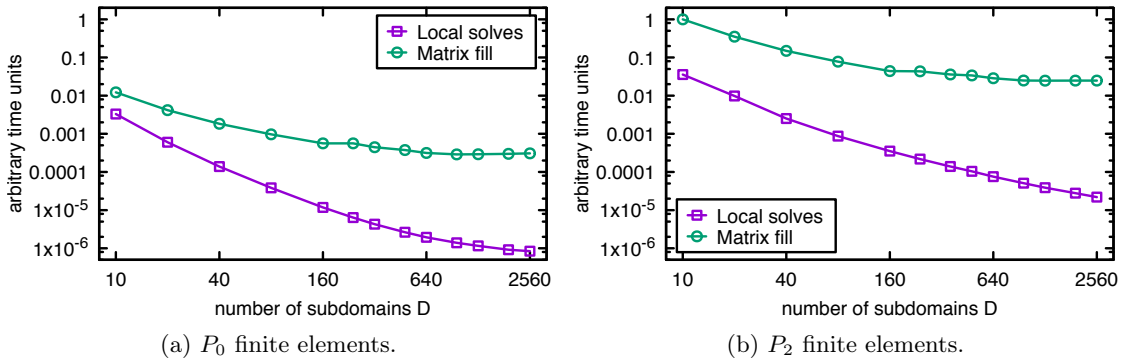


Figure 11: Computational times of local solves and reduced problem assembly versus D . Computations are performed on a fixed mesh with $\text{Ne} = 16,441$ elements. The correlation length is $L = 0.1$ and the target accuracy is $\delta^2 = 10^{-3}$.

Finally, from Figures 11a and 11b it is found that the P_2 discretization is more expensive, in

terms of computational times, regardless of the number of subdomains. This is due to a higher number of unknowns for the P_2 discretization compared to the P_0 case. But a fair comparison between the two discretizations should not only account for the computational times but also for the spatial discretization error on the computed modes.

4.2 Parallel implementation

Our domain decomposition approach naturally lends itself to parallel computation. In addition to the reduced complexity of solving smaller problems, as reported above, further gain can be expected through parallelization. From the presentation of the method given in Section 2 and summarized in Algorithm 1, one can identify the three main stages of the approach. First, local generalized eigenvalue problems are solved at the subdomain level (see Eq. (14)); second, the stiffness matrix of the reduced (regular) eigenvalue problem is constructed (see Eq. (20)); and last, this reduced problem is solved. We shall focus here on the first two stages, which represent the core of our approach. Regarding the last stage, let us simply mention that libraries exist to efficiently deal with the parallel solution of reduced problem, for example, *PARPACK* (Parallel ARPACK) [19], *SLEPc* (based on *PETSc*) [10, 23], or the *Anasazi* package of *Trilinos* [2].

In what follows, we rely on a Message Passing Interface (MPI) approach to parallelize the solution of the local problems and the assembly of the reduced problem. We shall denote by N_{MPI} the number of MPI processes; we then split the set of subdomains into N_{MPI} subsets. Each process p holds the data corresponding to the D_p subdomains in the p -th subset. In a static *a priori* load-balancing approach, provided that the number of degrees of freedom are roughly constant from one subdomain to another, we would like the number of subdomains D_p handled by process p to be evenly distributed among the processes, that is $D_p = D/N_{\text{MPI}}$ for all p . In practice, to tackle the case where N_{MPI} does not divide D , we distribute the subdomains among the processes through

$$\forall p = 1, \dots, N_{\text{MPI}}, \quad D_p = \lfloor D/N_{\text{MPI}} \rfloor + \begin{cases} 1 & \text{if } p \leq (D \bmod N_{\text{MPI}}), \\ 0 & \text{otherwise,} \end{cases} \quad (42)$$

where $\lfloor \cdot \rfloor$ is the floor function. Such a distribution among processes is illustrated in Figure 12a, in a case where N_{MPI} does not divide D .

4.2.1 Parallel resolution of the local problems

In the first stage, the local problems are solved; they consist in the Fredholm equations defined by Eq. (14), which after discretization lead to the generalized eigenvalue problems (see Algorithm 1):

$$[K^{(d)}]\{\tilde{\phi}_\beta^{(d)}\} = \lambda_\beta^{(d)}[M^{(d)}]\{\tilde{\phi}_\beta^{(d)}\}. \quad (43)$$

It is clear that these problems are independent for each subdomain, and that only local data is needed to construct the stiffness and mass matrices $[K^{(d)}]$ and $[M^{(d)}]$. The parallel implementation of the local problems is thus trivial, as no communication is required between the corresponding MPI processes. Similarly, the local eigenvalues $\lambda_\beta^{(d)}$ and local eigenvectors $\tilde{\phi}_\beta^{(d)}$, for $\beta = 1, \dots, m_d$, can be conveniently stored locally in a distributed memory architecture. Because there are D_p subdomains held by process p , and one single local eigenvalue problem to be solved per subdomain, process p is thus in charge of solving D_p problems. Provided that the number of degrees of freedom are roughly constant from one subdomain to another, we could expect the workload to be approximately balanced among the MPI processes and thus lead to good parallel performance.

4.2.2 Construction of the reduced eigenvalue problem

The second stage differs from the previous one in many ways. Perhaps the most relevant difference is the need to access non-local data. Indeed, as can be observed from Eq. (20), the block-matrices $[\hat{K}_{i,j}]$ involve data that belong to subdomains Ω_i and Ω_j . For diagonal blocks ($i = j$), or for blocks corresponding to subdomains that are handled by the same process p , no MPI communication is required, as all the needed data is held by the same process. On the contrary,

for other blocks, data have to be exchanged between different MPI processes, namely the local modes $\tilde{\phi}_\beta^{(d)}$, as well as local mesh information for discretizing the integrals. This is handled by resorting to an all-to-all communication, specifically to the MPI function `MPI_Allgatherv`.

Another important difference lies in the fact that the total number of blocks, $N_K \equiv D^2$, grows quadratically with the number of subdomains D . The classical way of distributing the workload would be for each process p to handle the construction of the D_p block-rows corresponding to its subdomains. In other words, process p would compute $[\hat{K}_{i,j}]$ for all $j = 1, \dots, D$ and for all $i \in \mathcal{I}_p$, where \mathcal{I}_p denotes the set of subdomain indices handled by process p . Here, we further exploit the characteristic block-symmetric structure of the reduced problem matrix inherited from the symmetry of the covariance functions C . Specifically, we compute only the $N_K^\Delta \equiv D(D+1)/2$ upper (or lower) triangular blocks of the matrix. The computation of these N_K^Δ blocks is distributed among the N_{MPI} processes so that process p is in charge of $N_{K,p}^\Delta$ blocks. The distribution of the blocks on the different processes is again handled using Eq. (42), substituting D and D_p with N_K^Δ and $N_{K,p}^\Delta$, respectively. The distribution among processes is illustrated in Fig. 12b, in a case where N_{MPI} does not divide N_K^Δ .

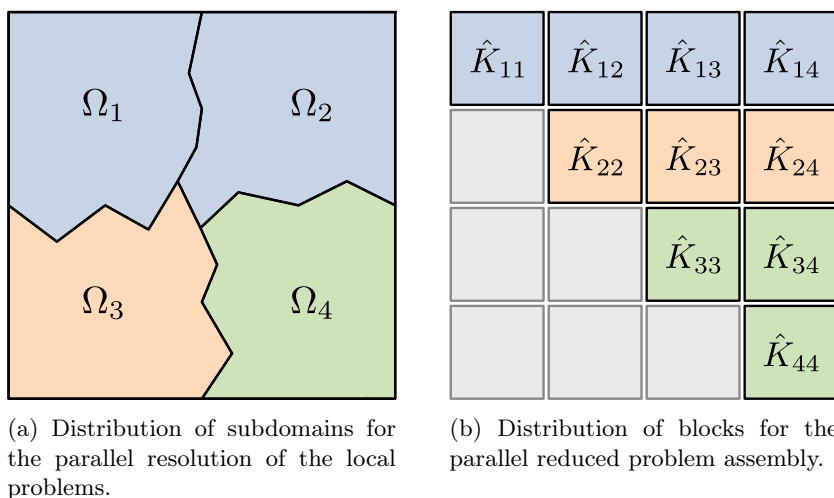


Figure 12: Example of distribution among MPI processes of the subdomains (Fig. 12a) and block computation for the reduced problem assembly (Fig. 12b). Case of $D = 4$ subdomains and $N_{\text{MPI}} = 3$ MPI processes. The processes $p = 1, 2, 3$ are colored in blue, orange and green respectively.

The cost of computing each block matrix $[\hat{K}_{i,j}]$ depends both on the number of degrees of freedom in subdomains Ω_i and Ω_j and on the number of local modes m_i and m_j retained for these subdomains. Provided that these are balanced, the workload of constructing the reduced problem should also be balanced among the processes and thus lead to good parallel performance. More general situations could be considered, in particular in the case of non uniform mesh and non-isotropic covariance structure, by introducing more advanced load-balancing strategies. This will be addressed in follow-on work.

4.3 Parallel efficiency

We now turn to investigate the efficiency of the parallel implementations of the local problem solves and the assembly of the reduced problem matrix. Specifically, we measure the scalability with the number of processes of performing these two tasks in parallel, including the MPI communication times involved. The scalability is characterized using two quantities, the parallel speedup S and the parallel efficiency E . The speedup and efficiency are reported as functions of the number N_{MPI} of MPI processes considered. The two measures are defined by

$$S(N_{\text{MPI}}) \equiv T(1)/T(N_{\text{MPI}}), \quad E(N_{\text{MPI}}) \equiv 100 \times S(N_{\text{MPI}})/N_{\text{MPI}} \quad (44)$$

where $T(N_{\text{MPI}})$ is the measured CPU times for the execution of the tasks using N_{MPI} processes. As the smallest number of processes tested is $N_{\text{MPI}} = 16$, the definitions of speedup and efficiency are actually based on the approximation $T(1) \approx 16T(16)$, assuming a perfect speedup for 16 processes. The tests were carried out on a parallel Blue Gene machine, fixing a constant number of MPI processes per computational node. For simplicity, we only report here the case with a single MPI process per node, such that the parallel runs use as many nodes as processes.

For a fixed finite element mesh with $N_e = 40,802$, we tested three different partitions of the domain, considering $D \in \{256; 512; 768\}$. A finer mesh with $N_e = 81,753$ elements, is also tested with $D = 256$ subdomains. The problem size being kept constant as the number of processes is increased, the reported speedup and efficiency correspond to a strong scaling experiment.

The results are reported in Fig. 13. Globally, the speedup and efficiency evolve satisfactorily as N_{MPI} is increased, especially considering that no fine tuning of the parallel implementation has been performed. For the largest number of subdomains tested, $D = 512$ and $D = 768$, the parallel efficiency decreases slowly with N_{MPI} , down to approximately 85% for 256 processes. This is due to the fact that the local problems are relatively small, owing to the large number of subdomains, making the MPI communication time more significant compared to the actual computation time. This trend becomes more pronounced as N_{MPI} increases.

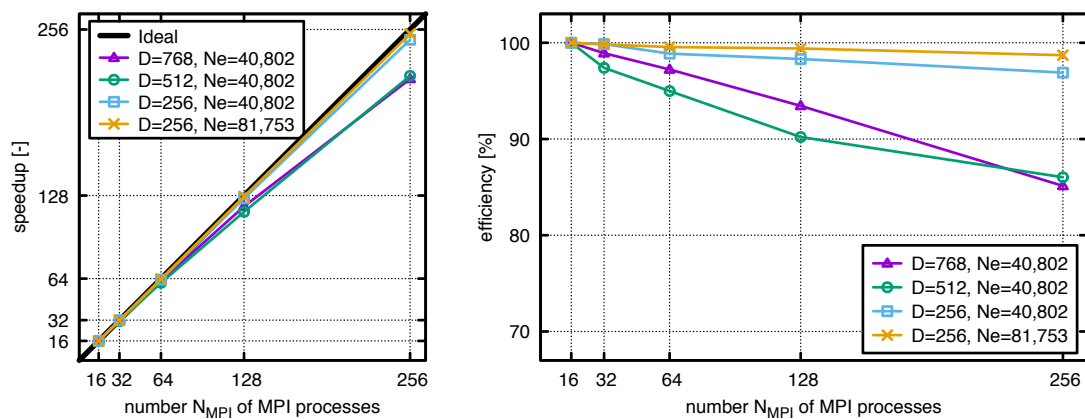


Figure 13: Speedup (left) and efficiency (right) versus the number of MPI processes. Plotted are curves for different values of D and N_e as indicated. The ideal scaling law is shown on the left plot using a solid black line. The correlation length is $L = 0.1$ and the energy criterion for the local modes is $\delta^2/2 = 10^{-3}$.

For $D = 256$, the actual computational time is large enough compared to the communication time, even for $N_{\text{MPI}} = 256$ processes. Consequently, the parallel efficiency remains close to ideal when N_{MPI} increases, with a reported efficiency of about 97% with 256 processes. This effect is confirmed considering a finer finite mesh with $N_e = 81,753$ elements for the same D (256). In this case, a higher efficiency of nearly 99% is measured for $N_{\text{MPI}} = 256$.

These tests demonstrate that the DD-KL approach is efficient and scalable. Not only does it benefit from cost reduction owing to the domain decomposition that breaks the problem in a set of small independent subproblems, but it also lends itself to natural parallel processing, involving limited communication, and thus enabling the efficient computation of the KL decomposition for large problems.

5 Conclusions

In this paper, we presented an efficient method to solve large-scale Kl decomposition problems, based on a work-subdivision strategy. Specifically, the computational domain is partitioned into smaller non-overlapping subdomains, over which local Kl decomposition problems are solved to generate local bases. A criterion has been proposed, and numerically demonstrated, for the selection of the local bases' dimensions in order to ensure a prescribed accuracy. The global

KL decomposition problem is subsequently reformulated by means of Galerkin projection in the subspace of the local modes. This procedure leads us to solve a reduced global problem, whose size is not related to the dimension of the underlying discretization space, but depends on the requested accuracy and the number of subdomains. The low dimensionality of the reduced problem enables efficient solution methods, including direct ones. But the approach is in fact flexible and can accommodate any type of eigenvalue solvers (direct, iterative, . . .), both for the computing the local KL bases and solving of the reduced problem. In addition, although we focused on the problem of performing KL decomposition, it should be noted that the approach could be used in many different contexts and extended to determine the dominant subspace associated to more general integral operators.

The method was illustrated on the approximation of stochastic processes defined over the two-dimensional unit square. A squared exponential covariance structure was assumed, with short correlation lengths yielding slowly decaying spectra. Our numerical experiments demonstrated that the approach provides a fine control of the approximation error, and is very robust with respect to the number of subdomains. It was also observed that the cost of solving the local problems reduces with the number D of subdomains, while the size and cost of solving the reduced problem increases with D . This suggests the existence of an optimal value for D , balancing the decreasing complexity of the local KL problems and the increasing size of the reduced problem. In any case, the range of values of D over which the approach remains effective is large such that selecting the optimal D is not critical. The sensitivity of the method to the shape of the subdomains was also explored. It was found that, in the case of isotropic covariance, it is more effective to use partitions with subdomains having similar size in all directions, rather than having geometrically highly stretched subdomains. In any case, the influence of the subdomains geometry is not too pronounced, such that the method would be able to effectively accommodate complicated situations (in the covariance structure, geometry of the domain, adapted finite element mesh, . . .) without having to construct a dedicated partitioner. This is particularly important in view of reusing standard libraries for partitioning the domain.

Beside the immediate computational advantages of breaking the scale size problem into smaller ones, the proposed approach naturally lends itself to parallel implementation. In fact, the parallelization of the local decompositions over individual subdomains is trivial, as local problems are fully independent from one subdomain to another, and the error criterion to select the local basis is also completely local. The parallel assembly of the reduced problem involves, on the contrary, more significant communication between processes, with a less obvious parallelization as a result. The parallel implementation, relying on the MPI framework, has been tested showing an excellent scaling up to 256 processes, provided that the initial global problem is large enough.

The proposed approach allows us to effectively generate approximated samples of a Gaussian process, using (26). Thus, as discussed in the introduction, the method is therefore an alternative to other existing methods to generate realizations of Gaussian stochastic processes with prescribed covariance structure. In fact, the solution of the reduced problem exhibits the correlation structure between local variables, such that one can also restrict the sampling to subsets of subdomains. Such a sampler is advantageous, for instance, to solve stochastic partial differential equations problems, in particular stochastic elliptic problems as considered in the second part of the paper.

Acknowledgments

This work was supported by the US Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC0010540. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. Support of the King Abdullah University of Science and Technology is also acknowledged. The parallel experiments were carried out on the IBM Blue Gene/Q machine of IDRIS (CNRS, Orsay) and the authors would like to thank J. Chergui for helpful discussions and his valuable assistance in carrying out the parallel experiments.

A Derivation of the Error Expression

We have an initial process $U(\mathbf{x}, \theta) \in L_2(\Omega, \Theta)$, its projection $U_{\mathcal{B}}(\mathbf{x}, \theta) \in \mathcal{V}_{\mathcal{B}} \times L_2(\Theta)$ and finally $\hat{U}(\mathbf{x}, \theta) \in \mathcal{V}_{\mathcal{B}} \times L_2(\Theta)$ the truncated KL expansion of $U_{\mathcal{B}}$ using $\hat{N} \leq n_t$ modes. Our concern is to derive an expression for the error $U - \hat{U}$ in the $L_2(\Omega, \Theta)$ -norm. By definition this is the square root of

$$\mathbb{E} \left[\left\| U - \hat{U} \right\|_{\Omega}^2 \right] = \mathbb{E} \left[\left\langle U - \hat{U}, U - \hat{U} \right\rangle_{\Omega} \right].$$

We have

$$\begin{aligned} \left\langle U - \hat{U}, U - \hat{U} \right\rangle_{\Omega} &= \left\langle (U - U_{\mathcal{B}}) + (U_{\mathcal{B}} - \hat{U}), (U - U_{\mathcal{B}}) + (U_{\mathcal{B}} - \hat{U}) \right\rangle_{\Omega} \\ &= \left\langle (U - U_{\mathcal{B}}), (U - U_{\mathcal{B}}) + (U_{\mathcal{B}} - \hat{U}) \right\rangle_{\Omega} + \left\langle (U_{\mathcal{B}} - \hat{U}), (U - U_{\mathcal{B}}) + (U_{\mathcal{B}} - \hat{U}) \right\rangle_{\Omega} \\ &= \|U - U_{\mathcal{B}}\|_{\Omega}^2 + \|U_{\mathcal{B}} - \hat{U}\|_{\Omega}^2 + 2 \left\langle (U - U_{\mathcal{B}}), (U_{\mathcal{B}} - \hat{U}) \right\rangle_{\Omega}. \end{aligned} \quad (45)$$

However the last term is zero as $(U_{\mathcal{B}} - \hat{U}) \in \mathcal{V}_{\mathcal{B}} \times L_2(\Theta)$ while $(U - U_{\mathcal{B}})$ is orthogonal to $\mathcal{V}_{\mathcal{B}} \times L_2(\Theta)$. Then it follows that

$$\mathbb{E} \left[\left\| U - \hat{U} \right\|_{\Omega}^2 \right] = \mathbb{E} \left[\|U - U_{\mathcal{B}}\|_{\Omega}^2 + \|U_{\mathcal{B}} - \hat{U}\|_{\Omega}^2 \right] = \mathbb{E} \left[\|U - U_{\mathcal{B}}\|_{\Omega}^2 \right] + \mathbb{E} \left[\|U_{\mathcal{B}} - \hat{U}\|_{\Omega}^2 \right].$$

References

- [1] K. E. Atkinson. *The numerical solution of integral equations of the second kind*, volume 4. Cambridge university press, 1997.
- [2] C. G. Baker, U. L. Hetmaniuk, R. B. Lehoucq, and H. K. Thornquist. Anasazi software for the numerical solution of large-scale eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 36(3):13, 2009.
- [3] W. Betz, I. Papaioannou, and D. Straub. Numerical methods for the discretization of random fields by means of the karhunen-loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 271:109–129, 2014.
- [4] O. Ditlevsen and N. Tarp-Johansen. Choice of input fields in stochastic finite-elements. *Probabilistic Engineering Mechanics*, 14:63–72, 1999.
- [5] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.
- [6] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3 edition, 1996.
- [7] R. Gutiérrez, J. C. Ruiz, and M. J. Valderrama. On the numerical expansion of a second order stochastic process. *Applied stochastic models and data analysis*, 8(2):67–77, 1992.
- [8] J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.
- [9] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [10] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, 31(3):351–362, 2005.
- [11] M. Kac and A. Siebert. An explicit representation of a stationary gaussian process. *Ann. Math. Stat.*, 18:438–442, 1947.

- [12] K. Karhunen. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Amer. Acad. Sci., Fennicade, Ser. A, I*, 37:3–79, 1947.
- [13] O. Le Maître, M. Reagan, H. Najm, R. Ghanem, and O. Knio. A stochastic projection method for fluid flow. ii. random process. *J. Comput. Physics*, 181:9–44, 2002.
- [14] O. P. Le Maître and O. M. Knio. *Spectral methods for uncertainty quantification*. Scientific Computation. Springer, New York, NY, 2010.
- [15] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar 1982.
- [16] M. Loève. Fonctions aléatoires du second ordre. In P. Lévy, editor, *Processus Stochastique et mouvement Brownien*. Gauthier Villars, 1948.
- [17] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.
- [18] Y. M. Marzouk and A. F. Ghoniem. K-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical n-body simulations. *Journal of Computational Physics*, 207(2):493–528, 2005.
- [19] K. J. Maschhoff and D. C. Sorensen. P_ARPACK: An efficient portable large scale eigenvalue package for distributed memory parallel architectures. In *Applied Parallel Computing Industrial Computation and Optimization*, pages 478–486. Springer, 1996.
- [20] H. G. Matthies, C. E. Brenner, C. G. Bucher, and C. G. Soares. Uncertainties in probabilistic numerical analysis of structures and solids-stochastic finite elements. *Structural safety*, 19(3):283–336, 1997.
- [21] 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):4521–4537, 2007.
- [22] K. Phoon, S. Huang, and S. Quek. Implementation of karhunen–loève expansion for simulation using a wavelet-galerkin scheme. *Probabilistic Engineering Mechanics*, 17(3):293–303, 2002.
- [23] J. E. Roman, C. Campos, E. Romero, and A. Tomas. SLEPc users manual. Technical Report DSIC-II/24/02 - Revision 3.6, Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, 2015.
- [24] C. Schwab and R. A. Todor. Karhunen–loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122, 2006.
- [25] G. Stefanou. The stochastic finite element method: past, present and future. *Computer Methods in Applied Mechanics and Engineering*, 198(9):1031–1051, 2009.
- [26] B. Sudret and A. Der Kiureghian. *Stochastic finite element methods and reliability: a state-of-the-art report*. Department of Civil and Environmental Engineering, University of California Berkeley, CA, 2000.
- [27] L. Tamellini, O. Le Maître, and A. Nouy. Model reduction based on proper generalized decomposition for stochastic steady incompressible navier stokes equations. *SIAM J. Scientific Computing*, 36(3):1089–1117, 2014.