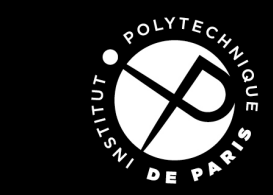
Preconditioning of domain decomposition methods for stochastic elliptic equations

Joao Felicio dos Reis



INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX059

Thèse de doctorat

Preconditioning of Domain Decomposition Methods for Stochastic Elliptic Equations

Thèse de doctorat de l'Institut Polytechnique de Paris

préparée à l'Ecole polytechnique

École doctorale n°574 École Doctorale de Mathématiques Hadamard (EDMH)

Spécialité de doctorat: Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 4 octobre 2021, par

JOÃO F. REIS

Composition du Jury :

M. Frédéric HECHT Professeur émérite, Sorbonne Université (Lab. Jacques-Louis Lions)	Président
M. Anthony NOUY Professeur des universités, École Centrale Nantes (Lab. de Mathématiques Jean Leray)	Rapporteur
M. Luc GIRAUD Directeur de recherche, Inria (HiePACS)	Rapporteur
Mme Nicole SPILLANE Chargée de recherche CNRS, École Polytechnique (CMAP)	Examinateur
M. Paul MYCEK Chargé de recherche, Cerfacs (Équipe Algorithmique Parallèle)	Examinateur
M. Pietro CONGEDO Directeur de recherche, Inria/École Polytechnique (CMAP, Platon)	Directeur de thèse
M. Olivier LE MAîTRE Directeur de recherche CNRS, École Polytechnique (CMAP, Platon)	Co-directeur de thèse

To my mother

Abstract

This thesis presents new numerical methods to efficiently generate solution samples from stochastic elliptic equations with a random coefficient field. Particular focus is on coefficient fields with high variability and low correlation length. Domain Decomposition (DD) methods are often used to accelerate the resolution of the deterministic elliptic equation by dividing the problem into a collection of dependent but less complex ones. The DD methods are also amenable to parallelism. This work concerns the adaptation of some deterministic DD methods to the sampling of stochastic problems.

Classical deterministic DD methods rely on iterative approaches that call for preconditioning strategies to achieve scalability and maintain a high convergence rate when the number of subdomains increases. In our stochastic context, determining a different preconditioner adapted to each sample may be costly, and alternative strategies tailored to the particular task at hand can be more effective.

Each solution sample amounts to solve a reduced linear system for the solution values at subdomains' interfaces, according to a Finite Element discretization. This reduced system is then solved by an iterative method. This thesis proposed three main contributions to the efficient preconditioning, introducing surrogate models of 1) the reduced global operator, 2) local subdomains' contribution to the reduced global operator, and 3) local preconditioners (multi-preconditioning).

The first contribution focuses on the additive Schwarz iterative method and introduces a stochastic preconditioner consisting of a surrogate of the Schwarz system for the unknown boundary values. In a preprocessing stage, a truncated Karhunen-Loève (KL) expansion of the coefficient field and a Polynomial Chaos (PC) expansion of the Schwarz system are com-

puted to form the stochastic preconditioner. Each sample's preconditioner is retrieved through a cheap evaluation of the PC surrogate in the sampling stage. Numerical experiments on a one-dimensional problem illustrate stability and fast convergence of the resulting approach, provided that the number of KL modes and the PC degree are both sufficiently large. In fact, the proposed method converges to the ideal preconditioner (yielding the solution in a single iteration) with the stochastic discretization.

The second contribution extends the previous idea to non-overlapping DD methods by constructing the Schur complement's PC surrogates. We leverage the structure of the Schur problem to better exploit the local character of the DD method. It leads to local PC expansions of the Neumann-Neumann (NN) maps with a small number of local random variables for discretizing the stochastic field over each subdomain. These local PC expansions are computed independently at an offline stage. Then, the PC expansions are evaluated and assembled during the inline sampling stage to form the preconditioner. We propose a decomposition of local operators to ensure that the retrieved preconditioner is almost surely symmetric-positive-definite. Numerical experiments show an average convergence up to 7 times faster than when preconditioning all samples with the Schur system corresponding to the coefficient's median value.

Preconditioning with a Schur system's surrogates requires solving at each iteration a problem with size equal to the number of interface nodes. This may be a limitation for large-scale problems in higher dimensions. Thus, the third contribution concerns a totally local preconditioner: the two-level NN preconditioner. Again, we propose to use local PC surrogates of the NN maps instead of solving the local preconditioning problems for each sample. Numerical experiments empirically show that surrogate-based local preconditioning is nearly as effective as the actual NN maps' computation for each sample. Further, local PC surrogates' use does not compromise the scalability of the local preconditioning associated with a coarse space projection.

Acknowledgements

The completion of this Ph.D. would have not been possible without the support of many people and organizations. I dedicate a few paragraphs to acknowledge everyone that made this journey a successful one.

The work developed in this thesis manuscript has been part of the UTOPIAE international training network, a scholarship funded by the Marie Skłodowska-Curie Actions program within the European Horizon2020 program. I have been provided with financial background above the average, covering important training, conferences, research output, and materials that have greatly improved my Ph.D. experience. I would like to thank all people involved in this program for providing me with this opportunity. A special thanks to my UTOPIAE colleagues Anabel, Bárbara, Christian, Cristian, Danda, Dani, Elisa, Gianluca, Lorenzo, Margarita, Thata, Thomas, and Zeno for all we have been through.

I would also like to dedicate some words to my supervisors, Pietro Congedo and Olivier Le Maître, the two individuals that guided me through this journey with patience and professionalism. In many situations, I had difficulties in surpassing some hurdles, understanding some concepts, in general, making through the cumbersome research maze. It would not have been possible to find my way without the help of my supervisors. Pietro has always been there to guarantee I have the conditions to proceed with my work. Olivier has probably been the most dedicated tutor I ever had. Thank you both for being there.

I would like to thank my friend Giulio, of course, my brother in arms. What a mess we made. Also, a special word to Nikos, the eternal neighbour that turns into a friend always there to support and lift me up. Paris would not have been the same without you two. I also want to thank the Maison du Portugal-André de Gouveia at Cité Internationale Universitaire de Paris, and in

particular, to the Calouste Gulbenkian Foundation, for providing me a home throughout most of my Ph.D. The House also gave me a chance to make more friends for life. Ana, Francisca, and Lucie thank you for all your support and friendship. It was in this House that I met the love of my life, Lilian. The distance did not set us apart and if this Ph.D. is one of the greatest achievements of my career, meeting you was one of the most fortunate events of my life. Sharing my daily life with you is a gift, and your support was essential to finalize this Ph.D.

The infinite support comes from far away, my Home in Setúbal. Even if we have not seen each other for more than 6 months, each reunion looked like we had seen each other the day before. António, Carlos, Morujão, Miguel and Pedro Júlio, Diogo and Alcaria, thank you for being there always, literally.

At last, I would like to thank my family: my mother, my father, and my sister. Not just for the support throughout my Ph.D., but for the 6 years, I have been studying abroad. I cannot tell how hard it must have been to see a son settling down in 4 different countries in such a short period and the months in a row without seeing each other. It has certainly been hard for me too, but as this paragraph comes to an end, I can only tell that the physical distance between us also becomes shorter.

Contents

List of Figures

List of Tables

Chapter 1

Introduction en Français

Les dernières décennies ont montré un développement croissant des ressources informatiques. Ces progrès ont permis aux industries de contourner de nombreuses expériences physiques expansives et complexes pour des simulations numériques qui sont beaucoup moins chères et plus faciles à réaliser.

Afin d'avoir des simulations réalistes, les modèles doivent tenir compte des différentes incertitudes de la physique. Un exemple de phénomènes physiques incertains est la prédiction de l'écoulement des eaux souterraines. La prédiction de l'écoulement des eaux souterraines est pertinente pour plusieurs situations pratiques telles que l'estimation des risques de contamination des sols, le contrôle de la qualité de l'eau potable, l'élimination des déchets radioactifs, etc. Une stratégie typique pour caractériser les paramètres géologiques consiste à utiliser des valeurs moyennes. Cependant, cela a tendance à être sujet à des erreurs puisque, par exemple, la variabilité spatiale de la porosité et de la perméabilité du sol peut être significativement grande. L'alternative consiste à attribuer un modèle probabiliste à ces paramètres. Cette modèle est généralement basée ou une combinaison de données expérimentales avec des connaissances sur les phénomènes physiques. La caractérisation des incertitudes des paramètres du modèle est difficile car les données expérimentales fournies peuvent comporter des erreurs ou être influencées par des facteurs externes au modèle. En plus, l'incertitude associée aux paramètres du modèle se traduit naturellement par sa sortie. Par conséquent, la présence de

propriétés d'incertitude géologique fait de la prévision de l'écoulement des eaux souterraines un problème très difficile. Naturellement, les dépendances aléatoires sont caractéristiques de nombreux autres phénomènes physiques, tels que le transfert de chaleur, les réactions chimiques, l'élasticité, la dynamique des fluides, la combustion, etc. La quantification d'incertitude ("Uncertainty Quantification", UQ) est le domaine qui vise à caractériser à la fois l'incertitude associée aux paramètres du modèle et la façon dont cette incertitude se propage à travers le modèle.

1.1 Sujet de la thèse

Cette thèse se concentre sur le développement de méthodes numériques pour caractériser la propagation de l'incertitude à travers des modèles avec des entrées aléatoires. Cette caractérisation peut se faire en estimant les quantités d'intérêt du modèle en sortie, telles que la moyenne, la variance, les différents types de densités, les quantiles, etc.

Même les méthodes numériques de pointe nécessitent une énorme puissance de calcul pour effectuer cette caractérisation avec précision. À titre d'exemple de l'ampleur de ces simulations, dans [76], les auteurs résolvent un modèle d'eau souterraine avec une porosité et une perméabilité incertaines, nécessitant quelques centaines de simulations numériques de problèmes jusqu'à $4,5$ millions de points de maillage spatial et entre $1\,000$ et $3\,000$ pas de temps. Le temps de calcul total pour effectuer ces simulations variait de 2 à 24 heures sur un cluster avec jusqu'à $19\,200$ unités de calcul.

Une façon d'optimiser les ressources de calcul disponibles consiste à exploiter des architectures parallèles. Cependant, l'implementation de simulations numériques sur des architectures parallèles nécessite naturellement des méthodes numériques adaptées. Les méthodes de décomposition de domaine (DD) sont particulièrement bien adaptées aux simulations parallèles et ont été largement appliquées à une variété de modèles stochastiques pour la caractérisation de l'incertitude. Cette thèse poursuit cet effort et présente une nouvelle méthode DD pour caractériser l'incertitude d'un type particulier de modèle: l'équation elliptique stochastique à coefficients aléatoires.

La caractérisation correcte de l'incertitude associée au type d'équations elliptiques stochas-

tiques considérées dans cette thèse suppose la disponibilité d'un nombre significatif d'échantillons de solutions. Pour cela, le cadre général de la méthode proposée revient à résoudre un grand ensemble d'équations elliptiques déterministes associées à des échantillons (snapshots) du champ de coefficients. Chaque échantillon est résolu par une méthode DD. Une méthode DD revient à scinder chaque problème échantillonné en problèmes locaux indépendants définis sur de petites portions du domaine spatial (sous-domaines). La solution de ces problèmes locaux satisfait certaines conditions de compatibilité inconnues sur l'interface des sous-domaines. Les méthodes DD construisent une séquence de fonctions locales qui convergent vers la solution de ces problèmes locaux via un schéma itératif. Les méthodes DD avancées sont caractérisées par des préconditionneurs qui rendent ce schéma itératif plus efficace.

La plupart des préconditionneurs qui caractérisent les méthodes DD avancées sont conçus pour résoudre un seul problème déterministe. Dans le cadre de pluseurs d'échantillons, la stratégie habituelle consiste à appliquer de manière répétée une méthode DD à chaque problème échantillonné. Cependant, le processus de construction d'un nouveau préconditionneur "à partir de zéro" représente une charge de calcul significative sur la résolution globale de chaque problème. La construction répétée de préconditionneurs devient un processus inefficace. Pour contourner ce problème, cette thèse propose une nouvelle classe de préconditionneurs. L'idée est de construire un metamodèle du préconditionneur qui caractérise votre méthode DD préférée. Ce metamodèle est ensuite évalué en fonction de chaque échantillon et de la réalisation résultante (préconditionneur à base des metamodèles) utilisé pour accélérer le schéma itératif. L'aspect principal de l'approche proposée est que le coût du préconditionnement à base des metamodèles est bien inférieur par rapport aux méthodes traditionnelles, toujours, sans compromettre la performance. Cette thèse présente trois applications du préconditionnement basé sur des metamodèles aux méthodes DD. Chaque application est traitée séparément dans les chapitres 3, 4 et 5.

1.2 Plan et Contributions Principales

Cette thèse présente trois applications d'une approche DD particulièrement adaptée pour résoudre plusieurs problèmes échantillonnés associés à des champs de coefficients très vari-

ables et faiblement corrélés. Chacune de ces applications est présentée séparément dans les chapitres 3, 4 et 5. L'aspect commun de ces trois applications est qu'elles sont caractérisées par une nouvelle classe de préconditionneurs, appelés *préconditionneurs basés sur des metamodèles*.

Cette section commence par une description générale du préconditionnement basé sur des substituts. Ensuite, une liste des principales contributions de cette thèse est aussi présentée.

1.2.1 Idée Générale de Préconditionnement à Base de Metamodèles

Avant d'introduire la nouvelle stratégie DD, rappelons le cadre général DD. Le cadre général DD revient à scinder chaque problème échantillonné en problèmes locaux indépendants définis sur des sous-domaines. La solution de ces problèmes locaux satisfait certaines conditions de compatibilité inconnues sur l'interface des sous-domaines. Les méthodes DD construisent des fonctions locales qui convergent vers la solution de ces problèmes locaux via un schéma itératif. Les méthodes DD avancées sont caractérisées par des préconditionneurs qui rendent ce schéma itératif plus efficace.

L'approche DD introduite dans cette thèse est caractérisée par des préconditionneurs basés sur des metamodèles. La nouvelle approche peut être considérée comme une extension de votre méthode DD préférée. Cette approache est divisée en deux étapes distinctes. L'étape de prétraitement construit le ou les metamodèle(s) du ou des opérateur(s) impliqués dans l'étape de préconditionnement de votre méthode DD préférée. Ensuite, la phase d'échantillonnage utilise ce (ces) metamodèle(s) pour contourner les opérations de préconditionnement coûteuses qui caractérisent la méthode DD originale, pour celles basées sur des metamodèles bon marché. L'aspect principal de la nouvelle approche basée sur les metamodèles est qu'elle est beaucoup moins chère que l'original, tout en ayant des performances similaires.

L'application de méthodes DD caractérisées par des préconditionneurs à base de metamodèles présente un certain nombre d'avantages par rapport aux applications de pointe existantes. Ces avantages sont résumés dans les trois paragraphes suivants.

Premièrement, la méthode DD basée sur des metamodèles est bien mieux adaptée à la résolution d'un grand nombre de problèmes échantillonnés que l'application directe de méthodes

DD communes conçues pour les équations déterministes. En effet, le coût de l'étape de prétraitement est factorisé à travers tous les échantillons. En conséquence, dans le contexte d'un grand nombre d'échantillons, le coût de construction de substitution par échantillon est négligeable. En outre, les solveurs locaux très chers qui caractérisent le préconditionnement dans la méthode DD originale sont contournés par des opérations basées sur des metamodèles moins chères. À condition que les metamodèles soient suffisamment précis, les performances de la méthode DD basée sur les metamodèles seront proches de la méthode DD d'origine.

Un autre avantage du préconditionnement à base de metamodèles est que la précision des échantillons de solution résultants ne dépend pas de la précision des metamodèles utilisés. En effet, chaque metamodèle n'est utilisé que comme préconditionneur, alors que le problème résolu est toujours l'original.

Afin d'atteindre le coût de mise en place répétée d'un nouveau préconditionneur, certaines approches utilisent à la place le même préconditionneur pour tous les échantillons. L'aspect principal de cette approche est que les solveurs locaux associés au préconditionneur peuvent être réutilisés à chaque échantillon sans frais supplémentaires. Le préconditionnement à base de metamodèles est adapté à chaque échantillon, ce qui améliore considérablement ses performances par rapport aux préconditionneurs indépendants de l'échantillon. La différence est particulièrement évidente dans le contexte des champs à forte variance et à faible corrélation. En plus, les metamodèles sont evalueés très efficassement, donc, le coût de chaque préconditionneur adapté a chaque échantillon est très petit.

1.2.2 Applications Explorées dans la Thèse

Cette thèse présente le préconditionnement basé sur des metamodèles à travers trois applications distinctes aux méthodes DD qui sont décrites séparément dans les trois chapitres suivants. Toutes les applications sont basées sur une discrétisation élement finites ("Finite Element", FE) de chaque problème échantillon. La génération de chaque échantillon de solution revient à résoudre un système FE réduit pour les valeurs des solutions sur les interfaces des sous-domaines. En particulier, dans le contexte des sous-domaines qui se superposent soulement dans une interface, la matrice FE réduite associée à ce système est connue sous le nom de

matrice complémentaire de Schur.

Le **Chapitre 3** présente le concept de préconditionnement basé sur des metamodèles. L'étape de prétraitement construit une development en série de Polynomial Chaos (PC) de la matrice réduite en utilisant une série de Karhunen-Loève (KL) tronquée du champ de coefficients. L'étape d'échantillonnage du préconditionneur de chaque échantillon est récupéré grâce à une évaluation très efficiente de la serie de PC. A chaque itération, ce préconditionneur est ensuite utilisé pour obtenir des valeurs approchées de la solution sur l'interface pour accélérer le schéma itératif de la méthode de Schwarz.

Le **Chapitre 4** présente un préconditionneur basé sur un metamodèle pour la méthode des Gradients Conjugués ("Conjugate Gradient", CG) pour résoudre le système du complément de Schur. L'étape de prétraitement construit des metamodèles locales indépendants basés sur séries de PC des composantes locaux du complément de Schur, en utilisant des paramétrisations locales du champ de coefficients. L'étape d'échantillonnage évalue chacun de ces metamodèles locaux en fonction de chaque échantillon. Les réalisations résultantes sont assemblées pour obtenir une matrice de Schur basée sur des metamodèles. Enfin, la méthode CG Preconditionnée ("Preconditioned CG", PCG) utilise la matrice de Schur basée sur des metamodèles résultant comme préconditionneur pour résoudre le système de Schur échantillonné. La stabilité de la méthode PCG est garantie en utilisant une construction non triviale qui produit un préconditionneur symmétrique et défini positif ("Symmetric and Positive-Definite", SPD) pour chaque échantillon au sens presque sûr.

Le **Chapitre 5** présente une méthode de Neumann-Neumann (NN) basée sur des metamodèles. La méthode NN basée sur des metamodèles est caractérisée par un préconditionneur qui effectue des opérations locales basées sur des metamodèles, à la place de la résolution coûteuse des problèmes locaux associés à l'étape de préconditionnement de la méthode NN originale. Le préconditionneur NN basé sur un metamodèle qui en résulte est un préconditionneur parallèle qui s'avère être une alternative au préconditionneur NN d'origine pour la résolution de plusieurs problèmes échantillonnés. Ce préconditionneur est finalement combiné avec différents espaces grossiers pour fournir une méthode *Balancing Neumann Neumann* (BDD) basée sur des metamodèles.

1.3 Résumé des Principales Contributions

Les contributions spécifiques de cette thèse sont énumérées ci-dessous:

- Introduction d'une nouvelle classe de préconditionneurs pour les méthodes de décomposition de domaine appelés préconditionneurs basés sur des metamodèles;
- Introduction d'une nouvelle méthode Schwarz basée sur des metamodèles qui peut fournir des taux d'accélération satisfaisants par rapport à method de Schwarz d'origine dans le contexte d'un grand nombre d'échantillons.
- Introduction d'un préconditionneur SPD basé sur metamodèles pour la méthode du gradient conjugué qui est spécifiquement adapté pour résoudre des problèmes de Schur échantillonnés en grand nombre.
- Introduction d'une méthode Neumann-Neumann stable caracterisée par des préconditionneurs basés sur des metamodèles spécialement conçus pour la résolution d'un grand nombre de problèmes échantillonnés. La combinaison du nouveau préconditionneur avec différents espaces grossiers fournit différentes variantes d'une nouvelle méthode *Balancing Domain Decomposition* basée sur des metamodèles.
- Introduction d'un espace grossier GenEO indépendant de l'échantillon comme alternative à l'espace grossier GenEO [89] original pour la résolution de plusieurs problèmes échantillonnés.
- Certaines parties de ce travail ont été publiées séparément dans trois articles de revue [110, 108, 109].

Chapter 2

Introduction

The last decades have seen an increasing development of computational resources. These advancements have enabled industries to bypass many expensive and complex physical experiments for numerical simulations that are cheaper and easier to perform.

In order to have realistic simulations, models must account for the different uncertainties of physics. An example of uncertain physical phenomena is the prediction of groundwater flow. The prediction of groundwater flow is relevant to several practical situations such as risk estimation of soil contamination, quality control of drinking water, radioactive waste disposal, etc. A typical strategy to characterize the geological parameters is to use average values. However, this tends to be prone to errors since, for example, the spatial variability of the soil's porosity and permeability can be significantly large. The alternative is to assign a probabilistic model to these parameters. This model is usually based on experimental data as well as knowledge of the physical phenomena. The characterization of the uncertainties of model parameters is challenging since the experimental data provided may have errors or be influenced by phenomena that is not accounted in the model. Also, the uncertainty associated with the model parameters naturally translates into its output. Hence, the presence of uncertain geological properties makes the problem of predicting groundwater flow very challenging. Naturally, random dependences are characteristic of many other physical phenomena, such as heat transfer, chemical reactions, elasticity, fluid dynamics, combustion, etc. Uncertainty Quantification (UQ) is the field

that aims at characterizing both the uncertainty associated with the model parameters and the way that this uncertainty propagates through the model in the solution.

2.1 Subject of the Thesis

This thesis focuses on developing numerical methods to characterize the propagation of uncertainty through models with random inputs. Estimating some statistics of the model output's (quantities of interest), such as the mean, the variance, different types of densities or quantiles, represent different ways of making this characterization.

Fine statistical characterizations require significant computational power. As an example of how extensive these simulations can be, in [76] the authors solve a time-dependent groundwater model with uncertain porosity and permeability, requiring a few hundred numerical simulations of problems up to 4.5 million spatial mesh points and between $1,000$ to $3,000$ time-steps. The total computational time to perform these simulations ranged from 2 to 24 hours on a cluster with up to $19,200$ computational units.

One way to optimally use the available computational resources is to exploit parallel architectures. The implementation of numerical simulations on parallel architectures naturally requires appropriate numerical methods, and Domain Decomposition (DD) methods are particularly well suited for parallel simulations. This thesis presents novel DD methods for characterising the uncertainty of a particular type of model: the stochastic elliptic equation with random coefficients.

The correct characterization of the uncertainty associated with the stochastic elliptic equations considered in this thesis assumes the availability of a significant number of solution samples. To this end, the general framework proposed amounts to solve a large set of deterministic elliptic equations associated with samples (snapshots) of the stochastic coefficient field. Each sample is solved by a DD method. A DD method amounts to split each sampled problem into local problems defined on small portions of the spatial domain (subdomains). The solution to these local problems satisfy some unknown compatibility conditions. The DD method builds a sequence of local functions that converge to the solutions of these local problems through an iterative scheme. Advanced DD methods involve preconditioners that make this iterative scheme more efficient.

Most of the preconditioners of DD methods are fitted to particular deterministic problems. A classical strategy to solve multiple sampled problems amounts to apply a DD method to each problem, repeatedly. However, the process of constructing the preconditioner “from scratch” represents a significant computational burden on the overall resolution of each problem. The repeated construction of preconditioners becomes an inefficient process. In order to bypass this issue, this thesis proposes a new class of preconditioners. The idea is to build surrogates of the preconditioner of some DD method. These surrogates are then evaluated according to each sample, and the resulting realization (surrogate-based preconditioner) is used to accelerate the iterative scheme. The key aspect of the proposed approach is that the cost of surrogate-based preconditioning is significantly smaller compared to the straightforward approach while performances are not compromised. This thesis presents three contributions of surrogate-based preconditioning to DD methods. Each contribution is discussed separately through Chapters 3, 4 and 5, respectively

In this chapter we progressively motivate surrogate-based preconditioning. Section 2.2 introduces the stochastic elliptic equation with random coefficients. Section 2.3 describes two distinct classes of numerical methods used to characterize uncertainty: functional representation methods and sampling methods. Sampling schemes substantially benefit from the proposed method for reasons also explained in Section 2.3. Section 2.4 gives a summary of DD methods for the resolution of deterministic elliptic equations. Section 2.5 discusses state-of-the-art applications of DD strategies into solution methods for stochastic elliptic equations. This section includes a discussion of the main limitations of these approaches, which motivate surrogate-based preconditioning. Finally, Section 2.6 gives a general description of surrogate-based preconditioning as well as the main advantages of surrogate-based preconditioning with respect to the state-of-the-art applications previously described. In addition, this section provides a list of the main contributions of this thesis as well as a summary of the contribution of chapters 3, 4 and 5.

2.2 Elliptic Equation with Random Coefficients

In the following, we introduce the elliptic equation with random coefficients. This stochastic equation will be the test model studied through this thesis.

Let Ω be a geometric domain and Θ a set of events. Let $\boldsymbol{u} : \Omega \times \Theta \to \mathbb{R}$ be the solution of the stochastic elliptic partial differential equation given by

$$\nabla \cdot (\boldsymbol{\kappa}(x,\theta)\nabla \boldsymbol{u}(x,\theta)) = -f(x), \quad x \in \Omega \quad \theta \in \Theta \tag{2.1a}$$

$$\boldsymbol{u}(x,\theta) = \boldsymbol{u}_{BC}(x), \quad x \in \partial\Omega, \quad \theta \in \Theta. \tag{2.1b}$$

where $f(x)$ is a deterministic source and $\boldsymbol{u}_{BC}(x)$ denotes the Dirichlet boundary conditions. The elliptic equation (2.1) is a prototype of more complex models arising in many different physical phenomena such as porous media [20, 27, 26, 59, 128, 137], chemical reactions [36], wave scattering [50], heat diffusion [131] and elasticity [135]. The coefficient field $\boldsymbol{\kappa}$ is assumed to be random. The randomness of the solution $\boldsymbol{u}$ is a consequence of its dependence on the random coefficient field. The solution of the elliptic equation (2.1) exists almost surely (a.s.) and is unique provided that the random coefficient field is a.s. bounded from below and above for almost every $x \in \Omega$ [21]. Refer to [125] for a detailed analysis on existence and uniqueness of solutions to this equation.

In this thesis, the characterization of the solution is performed with a particular focus on coefficient fields with large variability and short-scale spatial correlations, as it is often a necessary assumption in many of the applications described above.

2.3 Solution Methods for Stochastic Elliptic Equations

Estimating statistics from solutions of elliptic equations associated with highly variable and low correlated fields is challenging. The techniques used to estimate these quantities are split into functional representation methods [11, 7] and sampling methods [17, 78].

2.3.1 Functional Representation Methods

The idea of functional representation methods is to find a finite representation of the solution by a functional approximation, which hopefully makes the characterization of this uncertainty easier to perform. Since the randomness of the solution is a consequence of the randomness of the coefficient field, a crucial aspect for this representation lies in the representation of the random coefficient field by a finite-dimensional probabilistic model. One way to perform this representation is to parametrize the input data through a finite number of mutually independent random variables. This set of random variables, depending on the event θ, will be denoted as a finite dimensional random vector $\boldsymbol{\xi}(\theta)$. In this sense, we write $\tilde{\boldsymbol{\kappa}}\left(x, \boldsymbol{\xi}(\theta)\right) \approx \boldsymbol{\kappa}(x, \theta)$. A classical parametrization technique is its Karhunen-Loève (KL) expansion [65, 79] (see Appendix B). The field's parametrization enables the construction of a functional representation of the solution in a Fourier-type spectral expansion of the form

$$\boldsymbol{u}(x, \theta) \approx \tilde{\boldsymbol{u}}\left(x, \boldsymbol{\xi}(\theta)\right) = \sum_{\alpha=0}^{J} u_{\alpha}(x) \boldsymbol{\Psi}_{\alpha}\left(\boldsymbol{\xi}(\theta)\right). \tag{2.2}$$

Once this functional is available, statistics of the solution can be obtained using evaluations of the spectral expansion (surrogate-based sampling) or directly retained from the information on this expansion [13, 67, 124].

Polynomial Chaos (PC) expansions [56, 70] represent a class of functional representation methods with the form (2.2) which have been extensively applied to the stochastic elliptic equation [133, 30, 29, 98, 122, 4, 47]. The stochastic basis $\{\boldsymbol{\Psi}_{\alpha}\}$ is set *a priori* as a finite polynomial basis (see generalized Polynomial Chaos [134]). Therefore, the construction of this expansion reduces to the estimation of the deterministic coefficients u_{α}. Methods for computing these coefficients include *intrusive (Galerkin)* methods (see [35, 3, 128, 2, 107]) and *non-intrusive* methods (see spectral projection methods [133, 30, 29, 98, 122], regression methods [14, 13, 124, 1] or collocation methods [4, 47, 131, 9]).

The size of the PC basis (number of terms) depends on the dimension of the random vector $\boldsymbol{\xi}$, and the complexity of the dependence of $\boldsymbol{u}$ on $\boldsymbol{\xi}$. Moreover, the uncertainty associated with highly variable and short-scale correlated fields requires many random variables to represent

the field properly. However, the dimension of the PC expansion (2.2) increases very rapidly with the number of input random variables [97], an effect known as the *curse of dimensionality*. Consequently, the cost of constructing this functional can become prohibitively high. An alternative is to drop the assumption on a pre-set polynomial basis and construct the pairs $(u_\alpha, \boldsymbol{\Psi}_\alpha)$ that yield a low-rank optimal representation of the form (2.2). See the expositions of the Proper Generalized Decomposition method [94, 96, 95] as well as the more general manuscript [11, Part II]. A number of applications to linear and non-linear problems include [96, 23, 42, 68, 106, 80]. The crucial aspect of low-rank representations is that the dimension of this representation only grows linearly with the number of random variables, which makes this approach suitable for problems associated with a high number of random variables.

The major drawback of estimating statistics using functional representations is that the error induced by the solution's surrogate propagates into the resulting (surrogate-based) estimates. In case many (surrogate-based) samples are considered, the error induced by the surrogate may dominate the global error of the estimate, which may compromise the convergence to a meaningful statistical value.

2.3.2 Sampling Methods

The classical Monte Carlo (MC) method estimates statistics using randomly selected solution samples. Each solution sample is the solution of an elliptic problem associated with a randomly generated sample of the coefficient field. For example, the classical MC estimator for the mathematical average of some quantity of interest $\boldsymbol{z}(\boldsymbol{u})$ is given by

$$\mathbb{E}\left[\boldsymbol{z}(\boldsymbol{u})\right] \approx \frac{1}{M} \sum_{m=1}^{M} z\left(u^{(m)}\right), \tag{2.3}$$

where M denotes the number of samples. Each solution sample $u^{(m)}$ is the solution of a deterministic problem of the form (2.1) associated with some deterministic coefficient $\kappa^{(m)}$. The interest of using MC-type sampling methods is that it does not require a representation of the field, and can handle any kind of complexity in $\boldsymbol{\kappa}$.

The main drawback of MC methods is their low rate of convergence. Classically, the error in

the estimator (2.3) will be $\mathcal{O}\left(\sqrt{\frac{\mathbb{V}[z]}{M}}\right)$ such that many samples are needed. Advanced sampling methods are generalizations of the classical MC method that try to accelerate the classical MC convergence. The Multi-Level Monte-Carlo (MLMC) method [57] and Quasi-Monte-Carlo (QMC) methods [38] are popular extensions of the classical approach, which have been applied to the stochastic elliptic equations in [27, 8, 61, 59, 69, 111], for instance.

Sampling-based estimates converge to the exact statistics (up to spatial discretization error) and are an attractive alternative to functional representations. However, the type of fields considered still pose significant challenges to sampling schemes. Indeed, an adequate representation of a sampled field with short scale correlations typically requires very fine spatial meshes that increase the cost associated with each sample. In addition, the large variability of the set of samples slows down the convergence of the sampling scheme. As a result, sampled-based estimators require a large amount of samples, that are expensive to compute, to accurately approximate quantities of interest. Hence, efficient numerical methods that reduce the cost associated with each sample are needed. A popular procedure is to apply a DD method to each sampled (deterministic) problem, and this will be the methodology used in this thesis. The following section gives a summary of general DD methods.

2.4 Domain Decomposition Methods

The idea of DD methods is to split the boundary value problem associated with each coefficient sample into a set of smaller (local) boundary value problems defined on portions of the spatial domain. Each portion of the spatial domain is called a *subdomain*, and the set of all subdomains forms a *partition* that covers the entire spatial domain. Subdomains can either overlap or not, in which case they only share an interface. The key aspect of DD methods is to establish *compatibility conditions* between the local problems so that the resulting local solutions coincide with the global solution over their subdomain. A general overview of DD methods can be found in the classical references [105, 127, 40].

The compatibility conditions depend on the type of problem and partition considered. For second-order elliptic problems, the local solutions should satisfy two continuity relations at the

interfaces: continuity of the unknown values and continuity of fluxes. In the overlapping setting, however, the flux continuity is a consequence of the continuity of the unknown values, which becomes the only necessary and sufficient compatibility condition. Since different compatibility conditions are needed according to different partition type, DD methods are often split into *Overlapping methods* and *Non-overlapping methods*.

Solving an elliptic problem using a DD method has several advantages over a global resolution on the entire spatial domain. One advantage is that DD methods are based on local operations defined over each subdomain. Usually, these local operations can be solved in parallel because they have no data dependencies. Another advantage is that the computational complexity of a DD method is reduced to the workload associated with each subdomain. Local operations on smaller subdomains (or equivalently, on partitions with a large number of subdomains) are often cheaper, which motivates using small-sized partitions. The reduced complexity associated with each subdomain combined with parallel local operations makes the resolution of each sampled problem by a DD method extremely efficient.

2.4.1 Iterative Domain Decomposition Methods

A usual way to obtain local solutions that satisfy the compatibility conditions is using iterative schemes. In general, these iterative schemes amount to sequentially solve at each iteration a set of local problems with prescribed boundary conditions. The convergence rate of the sequence of boundary conditions depends on the procedure used to update them. Classically, the update procedure is local in the sense that it uses information from the local interface nodes. Updating the boundary conditions from the purely local information yields a convergence rate that degrades as the partition of the domain uses more and more subdomains, because the information needs more iterations to be propagated within the domain.

Advanced DD methods use *preconditioners* to provide higher convergence rates. The idea behind preconditioners is to provide more effective updates of the boundary conditions for the local problems at each iterate. However, even with more effective boundary conditions, the convergence of the iterative scheme may still depend on the number of subdomains. A DD method whose convergence is independent of the number of subdomains is called *scalable*. In

order to obtain a scalable method, the action of preconditioning may have to be combined with a separate treatment of the part of solution that deteriorates the convergence. This separate treatment amounts to compute the slow part of the solution, which belong to a low dimensional *coarse space*, by a direct method, and then use the DD iterative scheme to compute the remaining part of the solution. Typically, a projected based strategy is used to keep the slow components of the solution outside the iterative scheme (orthogonal to the coarse space). A review on projection-based strategies can be found in [55, 54].

The key aspect of coarse space construction is defining a set of well-selected basis functions. In fact, for the type of elliptic problem considered, a coarse space spanned by the constant functions, the so-called Nicolaides coarse space [90], is very effective. However, a coarse space spanned by constant functions may not always provide the same robustness for different type of problems, such as problems with discontinuous fields (for details see [89]). Advanced coarse space constructions based on multiscale methods [44], energy minimizing methods [6] or spectral methods [118, 116, 39] are usual choices to achieve scalability for more complex problems. These methods can be used to construct coarse spaces either in overlapping or non-overlapping cases.

The discretization of the elliptic equation yields a (large) system, which the DD methods aim at solving efficiently. Spatial discretization techniques for solving deterministic elliptic problems include Finite Differences [74], Finite Volume [73], Finite Element [25, 45] and Spectral Methods [129]. In this thesis we rely on the standard Finite Element (FE) method, but the developments proposed can be generalised to other discretizations suitable for DD.

2.4.2 Overlapping Methods

Overlapping methods are based on partitions that overlap. At each iteration, the boundary conditions of the subdomains are taken from the internal solutions of the subdomains overlapping the interface. This updating yields a set of local solutions that converge to the global solution. Indeed, if the local solutions constructed at each iterate are continuous in the overlapping region, then the iteration converges to a set of local solutions with also matching fluxes across the interfaces.

An important class of overlapping domain decomposition methods is the so-called class of Schwarz methods. The earliest types of Schwarz methods include the alternating Schwarz method or the additive Schwarz method. The additive Schwarz method (SM) can be implemented in parallel, contrary to the alternating approach. For this reason, the SM has lead to the development of many variants. A complete discussion of both these methods can be found, for example, in [40, Chapter 1] and references therein.

The main issue with the SM is that the convergence is slow. The slow convergence of the SM has motivated the development of several extensions. One way of accelerating the convergence of the SM is to consider local problems with more effective boundary conditions than the simple Dirichlet type. A typical approach is to ensure that fluxes between neighbouring local solutions match at each iteration. Although this is not a necessary condition for convergence, it significantly accelerates the iterative scheme. One of the earliest attempts that proceed in this direction is the so-called P.L. Lions' DD method [75]. Several methods have been developed since then, giving rise to the so-called Optimized Schwarz method [53]. An example of a recent Schwarz variant is the so-called SORAS-GenEO-2 [60], which is the combination between the Optimized Schwarz method (preconditioner) and a spectral coarse space. In this thesis, we restrict ourselves to the standard Preconditioned SM (PSM), to introduce in Chapter 3 the basic concept supporting our surrogate-based preconditioners

2.4.3 Non-overlapping Methods

Non-overlapping methods are based on partitions where the subdomains intersect only at their interfaces. Non-overlapping subdomains are sometimes called *substructures*, and for this reason, non-overlapping methods can be referred to as *iterative substructuring methods*. Examples of non-overlapping methods include Dirichlet-Neumann method, the Neumann-Neumann (NN) method [15], the Finite Element Tearing and Interconnect (FETI) [48] methods or even the non-overlapping formulations of the P.L. Lions' DD method and the Optimized Schwarz method referred above. The NN and FETI algorithms have been given particular attention due to their parallel implementation properties. We refer to [39, Chapter 6] for a description of these two methods. The NN and FETI methods are the dual space formulations of each other, and for the

type of elliptic problems considered in this thesis, they have similar convergence rates.

The great majority of state-of-the-art DD methods arise from the non-overlapping NN or FETI methods. Due to the relation between these two methods, methods based on the NN formulation are often called *primal methods* whereas the ones based on the FETI approach are called *dual methods*. The Balancing Domain Decomposition (BDD)[81] and Balancing Domain Decomposition by Constrains (BDDC) [82] arise from the Neumann-Neumann method. The FETI-DP [49] arises from FETI method. The same relation between NN and FETI exists between BDDC and FETI-DP, i.e., they are dual, and their convergence is similar [83]. .

The discrete FE formulation of the NN and FETI methods amounts to solve a specific linear system for the values of the solution on the interface. This linear system is defined in terms of the Schur complement of the original global FE system. The reduced Schur system is classically solved by an iterative method, such as a PCG method using a specific preconditioner. The introduction of a coarse space amounts to introduce a deflation strategy to the PCG scheme. The resulting iterative scheme is often termed Projected PCG (PPCG) method, and details of this approach will be given in Chapter 5. The BDD and BDDC variants use the same NN preconditioner but different coarse spaces. The coarse space used in BDD is formed using subspaces of the local problems that characterize the original NN method, whereas the coarse space in BDDC is defined by choosing specific degrees of freedom from the stiffness matrix. Another difference is that the BDDC solves a smaller number of local problems per iteration compared to the BDD method. The coarse spaces used for BDDC can be used in combination with FETI to yield the FETI-DP method.

2.5 State-of-the-art DD Strategies for Elliptic SPDEs

In the following, a description of the recent applications of general DD approaches in the context of elliptic stochastic partial differential equations (SPDEs) is given. The discussion is split into applications to functional representation methods and applications to sampling methods. Particular focus is put on the limitations of these applications, which progressively motivate the development of new DD approaches presented in this thesis.

2.5.1 Domain Decomposition in Functional Representation Methods

2.5.1.1 DD methods for the SFEM

The Stochastic Finite Element Method [56] (SFEM) is an intrusive technique to compute the deterministic coefficients of a PC expansion of the elliptic equation solution. The process amounts to plug expansion (2.2) into the elliptic equation (2.1) and devise a linear system for the coefficients according to a Galerkin projection (details in [70, Chapter 4]). The resulting system is often large, and the only way to solve it is by an iterative method. This system is also highly structured which has led to the development of block-type preconditioners [102, 103, 112, 46, 128, 12].

The block structure of this Galerkin system has been adapted to a number of DD methods, where the blocks of the FE stiffness matrix are adapted to the subdomain partition. In [119], the authors proposed an extension of the SM to solve the Galerkin system. In [121], the same authors developed a similar approach using the BDDC method. The work [37] presents an extension of the BDDC method presented in [121].

In the context of highly variable and low correlated fields, the estimation of statistics using PC expansions is challenging. In the particular case of the SFEM method, the limitations are tied to the large dimension of the Galerkin problem because of the need of large PC basis. Adapting the resolution of this system using DD methods is often not enough to overcome this limitation. The memory needed to allocate this problem is also often prohibitively large, even using a parallel DD framework. The alternative is to coarsen either the spatial or the stochastic meshes, leading to inaccurate PC expansions and inaccurate surrogate-based solution samples. As a consequence, the resulting surrogate-based estimates are poor.

2.5.1.2 Local PC expansions

Some DD strategies target the curse of dimensionality of the SFEM observing that the short-scale correlations of the field are easier to represent at a local scale. In other words, the parametrization of the field at the subdomain level requires fewer random variables. Local parametrizations can be exploited to construct a set of independent local surrogates for each local solution. In [22], the authors apply this idea to construct local PC expansions of the so-

lution from local KL expansions. The locality of the PC expansions is achieved using local KL expansions, which in particular use a small number of random variables. In [126], the authors construct local representations of the solutions on the basis adaptation technique of [41], and also exploiting local KL expansions.

The construction of local PC expansions seems to be a promising approach, but there are still aspects that limit this approach. In general, these local strategies amount to force compatibility conditions between the local PC expansions, since each subdomain portion of the solution depends on the entire domain, and their enforcement can be numerically costly. It is also not clear how these kinds of approaches scale with the number of subdomains. It is worth keeping an eye on the further development of approaches based on local solution expansions. However, for very large problems with complex coefficient fields, sampling methods remain the method of choice for the estimation of statistics of stochastic elliptic equations.

2.5.2 Domain Decomposition in Sampling Methods

Due to the cost of functional representation methods in the context of highly variable and low correlation lengths, sampling methods are often a preferable choice to estimate statistics from solutions of stochastic equations. The following discussion presents some applications of DD methods to sampling schemes.

2.5.2.1 Straightforward application of DD methods

One way to obtain solution samples is to apply some DD method to each sampled problem, repeatedly. In order to understand the limitations of this approach and motivate alternative strategies, let us break down the operations involved in the straightforward application of DD methods. A list of the major operations follows:

- Assembling the preconditioner. This operation involves setting the operators involved in the preconditioning step, which may represent a significant computational cost.

- As referred above, the use of a preconditioner alone may not be sufficient to obtain a scalable method. As a result, a combination between a preconditioner and a coarse space

with subsequent projection-based iterations may be require. For the type of elliptic problems considered in this thesis, a coarse space spanned by constant functions is often enough to achieve scalability. In this case, the cost of assembling the coarse space is negligible. However, richer coarse spaces are sample-dependent, and the cost of their construction is non-negligible.

- If projection-based iterations are necessary, the pre-computation of the operators involved has non-negligible cost. These operations may not be totally local, and instead, have the same dimension of the coarse space, stressing the importance of working with low dimensional coarse spaces.
- The iterative scheme. This compasses the operations involved at each iteration, including the preconditioning and the projection steps, based on operators previously set up.

The construction of different operators represent a significant computational burden to the application of the DD method, and alternatives are needed to reduce these set-up costs.

2.5.2.2 Surrogates of operators

DD methods generally amount to solve (iteratively) a reduced system for the unknown interface values. The cost of assembling this system represents a significant computational burden on the resolution of each sample. In the deterministic context, we often prefer not to assemble the condensed system, and instead rely on matrix-free approaches. The situation is different in the context of stochastic elliptic equations, where, in order to bypass the construction of this system, some strategies rely on a surrogate of this linear system. Once the surrogate is available, a surrogate-based linear system can be easily generated according to each sample. In general, the linear systems associated with DD methods satisfy a particular structure formed by independent blocks associated with each subdomain. These blocks are amenable to local functional representations exploiting local parameterisations of the coefficient field. The key aspect of this approach is that the locality provided allows working with a smaller number of random variables, therefore, reducing the curse of dimensionality.

In [32] this surrogate-based approach is applied to a non-overlapping method. The authors form a surrogate of the Schur complement matrix using local PC expansions of its subdomain's

components. The offline stage constructs each PC expansion based on a local KL expansion [31]. This local approach facilitates the construction of accurate local PC expansions. The sampling stage produces a surrogate-based Schur complement system using realizations of surrogates of the Schur subdomain's components. Finally, each sampled Schur system is solved to obtain each solution sample. A similar strategy is proposed in [86].

Another example of this surrogate-based approach concerns multiscale methods. Multiscale FE methods produce an FE basis adapted to each sampled problem. However, the construction of the FE basis is usually an expensive step. In [63], the authors explore a surrogate-based multiscale FE method. This strategy is divided into inline-offline stages. The offline stage builds a reduced-order model of the problem used to construct each element of the FE basis. Then, at each sample, the expensive construction of the FE basis is bypassed by evaluating the reduced-order models. The resulting surrogate-based FE basis is used to compute a solution sample. The reduced-order models depend on local parametrizations of the coefficient field defined for each subdomain, such as local KL expansions.

The common aspect in these type of approaches is that the accuracy of the resulting surrogate-based solution samples is necessarily bounded to the accuracy of the surrogate of the system. If the surrogates are not accurate enough, the resulting estimates will be off the exact statistics, regardless of the number of samples generated.

The only way to retrieve exact solution samples is to work with the original system. Therefore, the focus should turn to the remaining operations that make the straightforward application of DD methods expensive, especially because for large-scale problems, matrix-free approaches are likely to be preferable. An approach that reduces the cost of preconditioning while preserving the original problem is discussed in the following.

2.5.2.3 Sample-independent preconditioners

One of the expensive aspects of the straightforward application of DD methods is that a preconditioner has to be set according to each sample. In order to reduce the cost of repeatedly setting preconditioners, some approaches use the same preconditioner for all samples. The key aspect of this approach is that the local systems of the preconditioner can be re-used for each

sample at no additional cost. Because the preconditioner does not depend on the sample used, this preconditioning strategy will be called *sample-independent preconditioning*.

Naturally, care must be taken to ensure that the selected preconditioner ensures convergence of the iterative scheme for all samples to be computed. Also, the sample-independent preconditioner usually provides a slower average convergence rate compared to sample-adapted preconditioners. In particular, this means that the number of necessary iterations to achieve convergence will be larger than using preconditioners adapted to each sample. In any case, provided that the preconditioner satisfies the necessary conditions for convergence, the application of the resulting sample-independent preconditioner can provide a significant reduction of the cost per iteration. If the number of extra iterations is not too large, the lower preconditioning cost may significantly reduce the overall cost associated with the generation of the solution samples. In [32] (discussed previously), the authors use a Schur complement matrix based on the field's median as a preconditioner to accelerate the resolution of each surrogate-based Schur system by PCG method.

In general, the average convergence rate associated with sample-independent preconditioning depends on the variability of the field. For instance, sample independent preconditioners based on approximations of the field such as particular snapshots or even statistics like the field's median, will be more effective for sampled fields similar to each other and to these approximations. When sampled fields are very different from each other, the proper approximations of the field by a single field's instance is challenging. Therefore, sample-independent preconditioners tend to work only for small varying and highly correlated fields, and in particular, are not suitable for the type of problems considered in this thesis.

2.5.2.4 Recycling methods

Another strategy to speed up sampling is to combine DD methods with recycling methods [115]. The idea of recycling methods is to update the iterative solver used for a particular sampled problem with information coming from previous samples. In theory, the more information the solver can gather, the more effective it will solve the each sampled problem. Recycling strategies have been combined with DD methods to solve stochastic elliptic PDEs, for example, in [64].

Here, the authors combine the Krylov recycling method introduced in [100] with the SM, making the generation of solution more samples more efficient. Contrary to sample-independent preconditioners, recycling strategies have an associated cost per sample. The cost depends on the amount of information transferred from sample to sample. To maintain its effectiveness, the amount of information exchanged is significantly larger when variance is large. As a result, these approaches may have a high cost for the type of fields considered. Although recycling methods are not considered in this thesis, these approaches have promising properties and should be included in future works.

2.5.3 Concluding Remarks

In the context of highly variable and low correlated fields, the estimation of statistics using functional representation methods is challenging. The limitations are tied to the large complexity of the PC expansions. The estimation of statistics based on inaccurate PC expansions leads to inaccurate surrogate-based solution samples. As a consequence, the resulting surrogate-based estimates are poor.

Techniques to reduce the curse of dimensionality include constructing local PC expansions of the solution over each subdomain. The problem is that the local solutions depend on the global spatial domain, making the construction of these PC expansions very challenging. In any case, the approach seems to have potential, and for this reason, it can be exploited in future works. For now, sampling methods remain the method of choice for the type of problems considered.

The straightforward application of DD methods to each sampled problem produces exact solutions, and it is amenable to parallel implementation. However, the successive application of DD methods requires performing many operations representing a significant computational burden and often limits the number of samples generated. Alternative DD methods try to bypass some of these costly operations.

One of the alternatives discussed here amounts to bypass the construction of the original sampled problem through the generation of a surrogate-based one. Another example given was using a surrogate-based of FE basis in the context of multiscale FE methods. DD methods

such as these two approaches produce solution samples whose accuracy is bounded to the surrogate's accuracy. Once again, the construction of accurate surrogates tends to be very challenging as the variability of the field increases, representing a major hurdle to the generation of accurate solution samples.

An alternative that guarantees exact solution samples (up to spatial discretization error) is to apply DD methods characterized by sample-independent preconditioners. The key aspect of these preconditioners is that they can be re-used for each sample with negligible cost per sample. However, this type of approach loses its effectiveness for fields with high variability and low correlation length. The idea of sample-independent preconditioners can be generalized to recycling approaches that can be more effective but with an inevitable much higher cost per sample. Recycling strategies can be a good alternative if the cost associated with each new sample is not too large. However, the proposed method in this thesis tries to solve the problem differently. The following section presents the general idea of the DD method proposed in this thesis.

2.6 Outline of the Thesis

This thesis presents three developments of DD methods designed to solve multiple sampled problems associated with highly variable and low correlated coefficient fields. These developments are presented separately in chapters 3, 4 and 5, respectively, and all are characterized by a new class of preconditioners, called *surrogate-based preconditioners*. In a sentence, a surrogate-based preconditioner is a surrogate of the preconditioner that characterizes some particular DD method.

This section starts with a general description of surrogate-based preconditioning. Then, a list of the main contributions of this thesis is presented. Finally, this section closes with a summary of each chapter.

2.6.1 Surrogate-Based Preconditioning

We first present a general description of surrogate-based preconditioning, including a discussion of the main advantages over state-of-the-art DD applications previously mentioned. Then, we provide a brief description of the three particular developments of surrogate-based preconditioning explored in this thesis.

2.6.1.1 General idea

Let us recall the general DD framework. The general DD framework amounts to split each sampled problem into independent local problems defined on subdomains. The solution to these local problems satisfies some compatibility conditions on the interface. The DD methods build a sequence of local functions that converge to the compatible solution of these local problems through an iterative scheme. Advanced DD methods are characterized by preconditioners that speed-up this iterative scheme.

The DD approaches proposed in this thesis are characterized by surrogate-based preconditioners. These novel approaches are split into two distinct stages. The pre-processing stage constructs surrogate(s) of the operator(s) involved in the preconditioning step of the DD method. Then, the sampling stage uses this (these) surrogate(s) to bypass the expensive preconditioning operations that characterize the original DD method for cheap surrogate-based ones. The key aspect of the new surrogate-based approach is that it has a cheaper iteration cost than the original one, while having a similar convergence rate. The DD methods using surrogate-based preconditioners has many advantages over straightforward DD methods.

First, the surrogate-based DD method is much better suitable for solving many sampled problems than the straightforward application of DD methods designed for deterministic equations. Indeed, the cost of the pre-processing stage is factorized through all samples. As a result, in the context of a large number of samples, the surrogate construction cost per sample is negligible. In addition, cheaper surrogate-based operations bypass the expensive local solvers that characterize preconditioning in the original DD method. Provided that the surrogates are accurate enough, the performance of the surrogate-based DD method will be close to the original DD method.

Second advantage of surrogate-based preconditioning is that the accuracy of the resulting solution samples does not rely on the accuracy of the surrogate used. Indeed, each realization of the surrogate is only used as a preconditioner, while the problem solved is still the original (exact) one.

Finally, the surrogate-based preconditioning aims at being adapted to each sample, making its performance significantly better than sample-independent preconditioners. The difference is particularly evident in the context of fields with large variance and low correlation.

2.6.1.2 Application exploited in the thesis

This thesis presents surrogate-based preconditioning through three separate approaches associated with different DD methods described separately in the next three chapters. All DD methods are based on a FE discretisation of each sample problem, and amount to solve a reduced system for the solution values on the subdomain's interfaces, namely, the Schur complement matrix.

In **Chapter 3** we introduce the concept of surrogate-based preconditioning and apply it to the additive Schwarz method. The pre-processing stage constructs a PC expansion of the reduced matrix using a KL expansion of the coefficient field. The sampling stage evaluates this PC expansion to generate a surrogate-based reduced matrix adapted to each sample. This matrix is then used to accelerate the SM iterative scheme.

In **Chapter 4** we introduce a global surrogate-based preconditioner for the Conjugate Gradient method to solve the Schur complement system. The pre-processing stage constructs independent local PC-based surrogates of the Schur matrix's subdomain components, using local parametrizations of the coefficient field. The sampling stage evaluates each of these local surrogates according to each sample. The resulting realizations are assembled to obtain a surrogate-based Schur matrix. Finally, the PCG method uses the resulting surrogate-based Schur matrix as a preconditioner to solve the sampled Schur system. The stability of the PCG method is guaranteed using a non-trivial surrogate construction that produces a symmetric and positive definite (SPD) preconditioner for every sample in the almost sure sense.

In **Chapter 5** we present a surrogate-based Neumann-Neumann method. The surrogate-

based NN method is characterized by a preconditioner that performs cheap local surrogate-based operations in place of the expensive resolution of local problems associated with the preconditioning step of the original NN method. The resulting surrogate-based NN preconditioner is a parallel and scalable preconditioner that proves to be an alternative to the original NN preconditioner in the context of multiple sampled problems. This preconditioner is eventually combined with different coarse spaces to provide a surrogate-based BDD method.

A more detailed summary of each chapter is provided after listing the main contributions of this thesis.

2.6.2 Summary of the Main Contributions

The specific contributions of this thesis are listed below:

- Introduction of a new class of preconditioners for Domain Decomposition methods in the context of sampled problems called surrogate-based preconditioners;
- Introduction of a prototyped surrogate-based additive Schwarz method (SM). This novel approach can provide satisfactory acceleration rates compared to the original SM in the context of a large number of samples.
- Introduction of an symmetric and positive-definite surrogate-based preconditioner for the Conjugate Gradient method that is specifically suited to solve large number sampled Schur problems.
- Introduction of a stable, parallel and scalable surrogate-based Neumann-Neumann preconditioner for the resolution of a large number of sampled Schur problems. The combination of the novel preconditioner with different coarse spaces provides different variants of a new surrogate-based Balancing Domain Decomposition method.
- Introduction of sample-independent local subspaces for the GenEO coarse space that can substitute the original GenEO local subspaces to solve multiple sampled problems.
- The work of this thesis has been partially published in [110, 108, 109].

2.6.3 Summary of Each Chapter

We now provide an overview of the main chapter's content.

Chapter 3 introduces the surrogate-based preconditioning concept. The idea is to accelerate the SM with a simple surrogate-based preconditioning approach. The pre-processing stage constructs a PC expansion of the matrix of the reduced system for the interface solution values. The PC expansion is based on a truncated KL expansion of the coefficient field. At the sampling stage, a PC-based reduced matrix is recovered thanks to the very efficient evaluation of the PC expansion according to each sample. This matrix is used at each iteration to form a PC-based approximation of the reduced system for the gap between interface values of the solution and the SM iterate's. The key aspect of the approach is that the solution of the PC-based reduced system is used to obtain much more effective Dirichlet boundary conditions for the overlapping local problems, compared to the standard SM method. Numerical experiments of a one-dimensional problem enable a detailed illustration of the different properties related to the PC-based preconditioner. Results empirically show that the PC-based reduced matrix converges to the ideal preconditioner (the exact reduced matrix) by refining the stochastic discretization parameters. Naturally, the ideal preconditioner yields a solution in a single iteration. Therefore, the closer the PC-based preconditioner is to the ideal one, the more effective it becomes. For practical use cases, the approach shows fast convergence provided that the number of KL random variables and the PC orders are large enough. However, two main issues are identified that are problematic in the context of higher variable fields. The first issue is that the preconditioning approach proposed does not guarantee convergence for all samples. In such situations, the classical SM iterate should be used instead. The second limitation is that the number of random variables tends to be too large to obtain fast convergence rates, limiting applications to higher-dimensional problems.

Chapter 4 provides an extension of a surrogate-based preconditioner that solves the two main issues faced in the application of the SM. The DD framework considered is now a non-overlapping partition, and the generation of each sample amounts to solve the so-called Schur complement system by a PCG method. The structure of the Schur problem is used to exploit the local character of the DD setting. To this end, the pre-processing stage computes local

surrogates of the Schur matrix's components associated with each subdomain. Then, at the sampling stage, the local surrogates are evaluated according to each sample, and the resulting realizations assembled to form a surrogate-based Schur matrix according to each sample. The surrogate-based Schur matrix is then used as a preconditioner to solve the Schur system by a PCG method. The local surrogates are based on local PC expansions of a specific decomposition of the Schur matrix's local components. This construction ensures that the sampled preconditioners are SPD, which is a necessary and sufficient condition for the convergence of the PCG method. Moreover, the local PC expansions use few local random variables to discretize the stochastic field on each subdomain thanks to local parametrizations. Numerical experiments on two-dimensional test problems compare the proposed preconditioner with the (sample-independent) median-based preconditioner. Results empirically show that the surrogate-based preconditioner provides average convergence rates up to 7 times faster than the sample-independent preconditioner in large variance and low correlation context. The application of the preconditioner involves the resolution of a linear system whose size is the number of total interface nodes. The global character of this step is what accelerates the propagation of information throughout the global domain, providing effective acceleration rates. However, the number of interface nodes increases with the number of subdomains. Therefore, the resolution of this global system may become a severe limitation in the context of very high-dimensional problems, requiring a local alternative approach in these situations.

Chapter 5 introduces a surrogate-based preconditioning approach totally based on local operations. Similarly to the previous chapter, the DD framework considered is a non-overlapping partition. The starting point is to note that the NN method is equivalent to solving the Schur system using a specific preconditioner: the NN preconditioner. This chapter adapts the surrogate-based preconditioning methodology developed in the preceding chapters and introduces a new surrogate-based preconditioner for the NN method. The key aspect of the new preconditioner is that much cheaper surrogate-based operations bypass the resolution of the local problems that characterize the original NN method. These local surrogate-based operations are based on local PC and KL expansions with limited complexity. Contrary to the previous Schur-based preconditioner, the surrogate-based NN preconditioner acts locally, and the iterative scheme can naturally be implemented in parallel. The local NN maps can be singular such that the precondi-

tioner must be combined with a projection approach to effectively reduce all components of the solution. To this end, the surrogate-based NN preconditioner is balanced with the Nicolaides coarse space to produce a parallel and scalable method. Numerical experiments show that local preconditioning based on local surrogates is almost as effective as the full sampled-based computation of the NN maps, even though the former has a significantly lower application cost. The Nicolaides coarse space is eventually extended to more general GenEO coarse spaces [89]. A GenEO coarse space construction based on the field's median is introduced as a possible alternative to constructing sample dependent local subspaces. A discussion on the adaptation of sample-dependent coarse spaces to the resolution of multiple sampled problems follows. In general, the proposed surrogate-based BDD method proves to be a viable alternative, in terms of number of iterations, for the resolution of multiple sampled problems than the straightforward application of DD methods design to solve deterministic problems. To the author's knowledge, this application represents the first-ever completely adapted, parallel and scalable surrogate-based preconditioning strategy.

Chapter 3

Stochastic Preconditioners for the Addtive Schwarz Method

This chapter describes a first attempt to generate surrogate-based preconditioners. An elementary application to the additive Schwarz method on a one-dimensional problem keeps the level of the discussion fairly simple. Numerical experiments show that this novel preconditioning strategy has potential to be used in higher dimensional problems.

3.1 Introduction

Inspired by works on domain decomposition (DD) methods for elliptic equations [32, 86], we aim at accelerating iterative DD methods for elliptic problems. We start by considering a sample of the coefficient field and the associated discrete problem. We will use finite element (FE) discretization, but the strategy is not restricted to any particular kind of spatial discretization. Then, we consider a partition of the domain into overlapping subdomains, reducing the resulting FE system into a system for the boundary nodes of the subdomains known as the boundary-to-boundary system. This system is the analogous version of the well-known Schur complement system for non-overlapping subdomains [105, 127]. Although the boundary-to-boundary system is significantly smaller than its associated FE system, it is often still too large to be solved by a direct method. Iterative methods [113] are therefore a preferable choice in these situations. In this contribution, we use the additive Schwarz method [75] (SM) to solve this system. As with any other choice of an iterative method, the application of preconditioners to accelerate the SM is essential to obtain accurate solutions within feasible times.

Preconditioners for the SM can essentially be of two types: deterministic or stochastic. Deterministic preconditioners are those based on statistics of the coefficient field, which are constant for all samples. Once constructed, a deterministic preconditioner is re-used for each sample at no additional cost. For this reason, the associated cost of applying this preconditioner is minimal. However, especially for non-smooth fields, sample independent approximations of the coefficient field are usually inaccurate, resulting in ineffective deterministic preconditioners. Classical strategies [103, 119], in the context of the SFEM, consists of selecting the preconditioner associated with a particular coefficient field, such as the mean or median of the stochastic coefficient field. In this work, we introduce stochastic preconditioners adapted to each coefficient sample and therefore sample dependent, resulting in a significant acceleration of the iterative scheme. If the cost of adapting to each sample is sufficiently small, the use of stochastic preconditioners becomes preferable over deterministic ones.

The present work is not restricted to a particular UQ method and can benefit both MC-based and surrogate-based strategies. We target non-smooth parameters, where MC-based estimations require many samples, but accurate surrogate models are also challenging to estimate.

Indeed, in the MC context, the acceleration of the resolution of each problem is clearly beneficial, as it allows a reduction of the cost associated with each sample and, therefore, the possibility of generating more samples and retrieve more accurate estimates. The benefits of the present strategy are also extensible to surrogate methods, in particular, to non-intrusive strategies (such as regression methods [14] and spectral projection methods [99, 107, 29, 30]), which rely on sampled solutions to construct the functional representations. The more samples generated, the more accurate the surrogate is, and consequently, more accurate estimations are obtained.

This chapter is divided into 4 sections. Section 3.2 introduces some mathematical background and the elliptic equation with random coefficients. This section also describes the classic SM method and how it is employed in the MC context. Section 3.3, describes how different types of preconditioning strategies can be applied to the SM. The performance the different preconditioners is demonstrated through numerical experiments in Section 3.4. Finally, we give some concluding remarks and prospective work in Section 3.5.

3.2 The Schwarz Method for the Stochastic Elliptic Equation

We start the exposition by introducing some of the mathematical background needed and the elliptic problem that will be used throughout this work.

3.2.1 Deterministic and Stochastic spaces

Let $\Omega = [0, 1]$ be the geometric domain. Consider the space of squared 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 defined by the scalar product

$$\langle f, g \rangle_\Omega := \int_\Omega f(x)g(x)\, dx, \tag{3.1}$$

with norm denoted by $\| \cdot \|_\Omega$. The space of $L^2(\Omega)$ functions with continuous first-derivative is denoted by $H^1(\Omega)$.

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

as

$$\mathbb{E}\left[\boldsymbol{u}\right] := \int_{\Theta} \boldsymbol{u}(\boldsymbol{\theta})\, d\mu(\boldsymbol{\theta}).$$

The space $L^2(\Theta)$ is again a Hilbert space defined by the scalar product

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\Theta} := \mathbb{E}[\boldsymbol{u}\boldsymbol{v}], \tag{3.2}$$

with norm denoted by $\|\cdot\|_{\Theta}$.

Consider the space of second-order stochastic processes $\boldsymbol{u} : (x, \boldsymbol{\theta}) \in \Omega \times \Theta \mapsto \boldsymbol{u}(x, \boldsymbol{\theta}) \in \mathbb{R}$, denoted by $L^2(\Omega, \Theta)$. This is again a Hilbert space defined by the scalar product,

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\Omega\times\Theta} := \mathbb{E}[\langle \boldsymbol{u}(x, \boldsymbol{\theta})\boldsymbol{v}(x, \boldsymbol{\theta})\rangle_{\Omega}]. \tag{3.3}$$

The space of second-order random fields with squared-integrable first spatial derivatives is denoted by $L^2(H^1(\Omega), \Theta)$.

3.2.2 Stochastic Elliptic Equation

The stochastic elliptic equation we are interested in has the form,

$$\begin{aligned} &\frac{d}{dx}\left[\boldsymbol{\kappa}(x, \boldsymbol{\theta})\frac{d}{dx}\boldsymbol{u}(x, \boldsymbol{\theta})\right] = f(x) \quad x \in \Omega,\ \boldsymbol{\theta} \in \Theta \\ &\boldsymbol{u}(0, \boldsymbol{\theta}) = u_0 \quad \boldsymbol{u}(1, \boldsymbol{\theta}) = u_1 \quad \boldsymbol{\theta} \in \Theta \end{aligned} \tag{3.4}$$

where $f(x)$ is a deterministic source, $\{u_0, u_1\}$ are the (deterministic) Dirichlet boundary condition, and $\boldsymbol{\kappa}(x, \boldsymbol{\theta})$ is the given stochastic coefficient of the problem.

Here, we restrict ourselves to the case of $\boldsymbol{\kappa}$ being a stationary log-normal stochastic process, whose log is a centered Gaussian process with mean μ and covariance function C:

$$\log \boldsymbol{\kappa}(x, \boldsymbol{\theta}) = \boldsymbol{g}(x, \boldsymbol{\theta}), \quad \boldsymbol{g} \sim \mathcal{N}(0, C). \tag{3.5}$$

The covariance of $\boldsymbol{g}$ is defined as,

$$C(x, x') := \sigma^2 \exp\left(-\frac{\|x - x'\|_{\Omega}^{\lambda}}{\lambda \ell_c^{\lambda}}\right), \quad \lambda \in \mathbb{N} \tag{3.6}$$

with variance σ^2 and correlation length ℓ_c. Problem (3.4) is well posed and $\boldsymbol{u}(x, \boldsymbol{\theta}) \in L^2(H^1(\Omega), \Theta)$ for the coefficient field defined above [21].

3.2.3 Monte Carlo Method

Let $\boldsymbol{z}(\boldsymbol{u})$ be some functional representing a statistics of the solution of the elliptic equation (3.4). We are interested in estimating $\mathbb{E}[\boldsymbol{z}]$ by MC. To do so, for a fixed set of events $\theta^{(m)}$, we consider the deterministic coefficient $\kappa^{(m)}(x) := \boldsymbol{\kappa}(x, \theta^{(m)})$. We call the $\kappa^{(m)}$ a sample of $\boldsymbol{\kappa}$, whose associated deterministic problem is given by

$$\begin{aligned} &\frac{d}{dx}\left[\kappa^{(m)}(x)\frac{d}{dx}u^{(m)}(x)\right] = -f(x) \quad x \in \Omega \\ &u^{(m)}(x = x_{\vdash}^{(d)}) = u_0, \quad u^{(m)}(x = x_{\dashv}^{(d)}) = u_1, \end{aligned} \tag{3.7}$$

for $m = 1, \ldots, M$ with $M \gg 1$. Then, the Quantaty of Interest (QoI) is estimated by

$$\mathbb{E}[\boldsymbol{z}(\boldsymbol{u})] \approx \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{z}(u^{(m)}). \tag{3.8}$$

In order to have well-converged statistics, we are required to solve many equations of the form (3.7). Therefore, we need fast and efficient solvers for solving equation (3.7). In this work, we use a particular iterative domain decomposition method to solve equation (3.7): The additive Schwarz Method (SM).

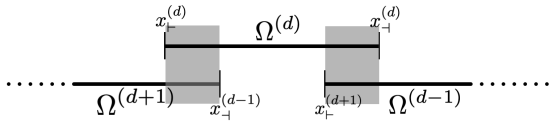


Figure 3.1: One dimensional DD setting with overlapping subdomains.

3.2.4 Schwarz Method

Consider a partition of Ω into D overlapping subdomains defined as $\Omega^{(d)} = [x_{\vdash}^{(d)}, x_{\dashv}^{(d)}]$. The vector of boundary points is defined as

$$\Gamma = [\Gamma^{(1)}, ..., \Gamma^{(D)}] \in \mathbb{R}^{N_\Gamma}, \text{ with } \Gamma^{(d)} = \{x_{\vdash}^{(d)}, x_{\dashv}^{(d)}\}, \tag{3.9}$$

where $N_\Gamma = 2D$ is the number of boundary points (including the edges of Ω). A sketch of this setting is given in Figure 3.1.

When not necessary, we will drop the superscript sample index m in the following for simplicity of notation. However, one should keep in mind that the following developments are sample dependent, just like in the previous section.

The SM is used to find the solution to the global problem (3.7) with corresponding local problems,

$$\mathcal{P}^{(d)}\left(\mathrm{u}_{\Gamma^{(d)}}, f, \kappa\right) := \begin{cases} \dfrac{d}{dx}\left[\kappa(x)\dfrac{d}{dx}w^{(d)}(x)\right] = -f(x), \quad x \in \left[x_{\vdash}^{(d)}, x_{\dashv}^{(d)}\right] \\ w^{(d)}(x = x_{\vdash}^{(d)}) = u_{\vdash}^{(d)}, \quad w^{(d)}(x = x_{\dashv}^{(d)}) = u_{\dashv}^{(d)} \end{cases} \tag{3.10}$$

where,

$$\mathrm{u}_{\Gamma^{(d)}} := [u_{\vdash}^{(d)}, u_{\dashv}^{(d)}].$$

The vector of solution values on the global interface is denoted by

$$\mathrm{u}_\Gamma := [\mathrm{u}_{\Gamma^{(1)}}, ..., \mathrm{u}_{\Gamma^{(D)}}]. \tag{3.11}$$

The SM can be recast as the fixed point iteration

$$\mathrm{u}_\Gamma^{k+1} = [\mathrm{L_S}]\mathrm{u}_\Gamma^k + \mathrm{b_S}, \tag{3.12}$$

where the entries of the matrix $[\mathrm{L_S}]$ correspond to the local solutions of the harmonic local problems with elementary boundary conditions, while the entries of the vector $\mathrm{b_S}$ are made of the local solutions for $\mathrm{u}_{\Gamma^{(d)}} = 0_\Gamma$ (see Appendix A for more details). Moreover, it can be seen that

$$\mathrm{u}_\Gamma^k \longrightarrow \mathrm{u}_\Gamma \tag{3.13}$$

and the boundary solution vector u_Γ solves the Schwarz problem

$$[\mathrm{S}]\mathrm{u}_\Gamma = \mathrm{b_S}, \quad [\mathrm{S}] = [\mathrm{I}] - [\mathrm{L_S}]. \tag{3.14}$$

The matrix $[\mathrm{S}]$ of the Schwarz problem is invertible, reflecting the uniqueness of the subdomain boundary values yielding the global solution.

The constitutive iteration of the additive Schwarz method is summarized in Algorithm 1. We stress that the set of local problems corresponding to each iteration (line 4) can be solved in parallel.

Algorithm 1 Schwarz Method iteration.

1: **procedure** SM-ITERATION($\mathrm{u}_\Gamma^k, \kappa, f$) ▷ Do one Schwarz iteration
2: $\mathrm{u}_\Gamma^{k+1} \leftarrow \mathrm{u}_\Gamma^k$ ▷ Initialize u_Γ^{k+1}
3: **for** subdomain $d = 1, \ldots, D$ **do**
4: Solve local problems: $\mathcal{P}_k^{(d)}(\mathrm{u}_{\Gamma^{(d)}}^k, f, \kappa)$ ▷ get $w_k^{(d)}(x)$
5: **if** $d > 1$ **then**
6: $\mathrm{u}_\dashv^{(d-1),k+1} \leftarrow w_k^{(d)}\left(x_\dashv^{(d-1)}\right)$ ▷ Update the left internal boundary point
7: **end if**
8: **if** $d < D$ **then**
9: $\mathrm{u}_\vdash^{(d+1),k+1} \leftarrow w_k^{(d)}\left(x_\vdash^{(d+1)}\right)$ ▷ Update the right internal boundary point
10: **end if**
11: **end for**
12: **Return** u_Γ^{k+1} ▷ Return the updated vector u_Γ^{k+1}
13: **end procedure**

Let us denote g^k the gap between two consecutive estimates of the solution, and ϵ_Γ^k the error

of the k-th estimate, defined respectively as

$$\mathrm{g}^k := \mathrm{u}_\Gamma^{k+1} - \mathrm{u}_\Gamma^k = ([\mathrm{L_S}] - [\mathrm{I}])\mathrm{u}_\Gamma^k + \mathrm{b_S}, \quad \epsilon_\Gamma^k := \mathrm{u}_\Gamma - \mathrm{u}_\Gamma^k. \tag{3.15}$$

It is easy to show that the gap and the error at any iteration k are related by

$$[\mathrm{S}]\epsilon_\Gamma^k = \mathrm{g}^k. \tag{3.16}$$

Indeed, using (3.12) and (3.14) we obtain

$$[\mathrm{S}]\epsilon_\Gamma^k = [\mathrm{S}]\mathrm{u}_\Gamma - [\mathrm{S}]\mathrm{u}_\Gamma^k = \mathrm{b_S} - ([\mathrm{I}] - [\mathrm{L_S}])\mathrm{u}_\Gamma^k = \mathrm{g}^k. \tag{3.17}$$

Similarly, one can derive the following relation relating the successive gaps:

$$\mathrm{g}^{k+1} = [\mathrm{L_S}]\mathrm{g}^k, \tag{3.18}$$

Relation (3.18) shows that the SM iterations converge provided that $[\mathrm{L_S}]$ is contractive. Let $\{\lambda_1, \cdots, \lambda_{N_\Gamma}\}$ be the set of eigenvalues of $[\mathrm{L_S}]$; we denote ρ_S the spectral radius of $[\mathrm{L_S}]$, which is the modulus of the leading eigenvalue of $[\mathrm{L_S}]$ (i.e., the eigenvalue with the largest modulus). The convergence is ensured if it holds

$$\rho_\mathrm{S} < 1. \tag{3.19}$$

The proof of convergence of the Schwarz method is standard and is not fully detailed here for brevity. We mention that the contractive character of $[\mathrm{L_S}]$ is inherited from the maximum principle of the local problem (so $\|[\mathrm{L_S}]\|_\infty = 1$) and the global homogeneous boundary values. The convergence rate of the SM for a given sample is then given by ρ_S, and we have the relation

$$\|\mathrm{g}^{k+1}\|_2 \leq \rho_\mathrm{S}\|\mathrm{g}^k\|_2 \leq \rho_\mathrm{S}^k\|\mathrm{g}^0\|_2. \tag{3.20}$$

However, the spectral radius ρ_S rapidly approaches 1 as the number of subdomains increases.

The stochastic nature of problem (3.4) motivates the definition of convergence rate for a random event $\boldsymbol{\theta}$. We denote this stochastic quantity by $\boldsymbol{\rho}_\mathrm{S}$, and we are usually interested in

$\mathbb{E}\,[\boldsymbol{\rho}_{\mathrm{S}}]$.

3.2.5 Preconditioned Schwarz Method

The preconditioned version of the additive Schwarz method (PSM) is designed to improve the convergence rate of the SM iterative scheme. The central idea is to modify the SM iteration in (3.12) to

$$\mathrm{u}_\Gamma^{k+1} = [\mathrm{L_P}]\mathrm{u}_\Gamma^k + [\mathrm{b_P}], \tag{3.21}$$

where, as before, the iteration in (3.21) can be made by essentially solving the local problems in parallel. Also, we want to define the preconditioned iteration such that the corresponding pair $([\mathrm{L_P}], [\mathrm{b_P}])$ in (3.21) leads to the same fixed point solution u_Γ, but at a faster rate. In other words, $[\mathrm{L_P}]$ should have a lower spectral radius: $\rho_{\mathrm{P}} < \rho_{\mathrm{S}}$.

To derive the preconditioned iteration, we start from the solution of the Schwarz problem in (3.16). Using the gap and error definitions in (3.15), it comes

$$\epsilon_\Gamma^k := \mathrm{u}_\Gamma - \mathrm{u}_\Gamma^k = [\mathrm{S}]^{-1}\mathrm{g}^k \implies \mathrm{u}_\Gamma = \mathrm{u}_\Gamma^k + [\mathrm{S}]^{-1}\left([\mathrm{L_S}]\mathrm{u}_\Gamma^k + \mathrm{b_S} - \mathrm{u}_\Gamma^k\right).$$

Given this expression, we can propose the following preconditioned iteration, for some *invertible* preconditioner $[\mathrm{P}] \approx [\mathrm{S}]$:

$$\mathrm{u}_\Gamma^{k+1} = \mathrm{u}_\Gamma^k + [\mathrm{P}]^{-1}\left([\mathrm{L_S}]\mathrm{u}_\Gamma^k + \mathrm{b_S} - \mathrm{u}_\Gamma^k\right). \tag{3.22}$$

We see immediately that u_Γ is a fixed-point of the previous iterations. In addition, as for the Schwarz one, the preconditioned iteration can formally be recast in

$$\mathrm{u}_\Gamma^{k+1} = [\mathrm{L_P}]\mathrm{u}_\Gamma^k + [\mathrm{b_P}], \quad [\mathrm{L_P}] \doteq [\mathrm{I}] + [\mathrm{P}]^{-1}([\mathrm{L_S}] - [\mathrm{I}]), \quad [\mathrm{b_P}] \doteq [\mathrm{P}]^{-1}\mathrm{b_S}. \tag{3.23}$$

In practice, the matrix $[\mathrm{L_P}]$ will never be assembled. Instead, a preconditioned iteration will proceed along the lines shown in Algorithm 2. This algorithm is built on top of the parallel Schwarz iteration in Algorithm 1 to compute the gap. Then, the current estimate of the boundary conditions is corrected in the preconditioning step; this step is global because it involves the

whole vector of boundary values. Also, note that rather than $[\mathrm{P}]$, its inverse or more likely factorisation will be used in the preconditioned iteration.

Algorithm 2 Preconditioned Schwarz iteration.

1: **procedure** PSM-ITERATION($\mathrm{u}_\Gamma^k, \kappa, f, [\mathrm{P}]$) ▷ Do one Preconditioned Schwarz iteration
2: $\quad \mathrm{g} \leftarrow$ SM-ITERATION$(\mathrm{u}_\Gamma^k, \kappa, f) - \mathrm{u}_\Gamma^k$ ▷ Compute gap
3: $\quad \mathrm{u}_\Gamma^{k+1} \leftarrow \mathrm{u}_\Gamma^k + [\mathrm{P}]^{-1}\mathrm{g}$. ▷ Compute the preconditioned update
4: $\quad$ **Return** u_Γ^{k+1} ▷ Return the updated vector u_Γ^{k+1}
5: **end procedure**

The particular forms of the fixed point iteration (3.23) and its associated Algorithm 2 ensure that it has u_Γ as a fixed point. Therefore, the PSM will converge whenever the apparent spectral radius of ρ_P is less than one. This is not granted, a priori, except for the particular case of $[\mathrm{P}] = [\mathrm{S}]$, ensuring that the solution is obtained in just one iteration ($\rho([\mathrm{L}]_{[\mathrm{P}]=[\mathrm{S}]}) = 0$). It is hoped that using an approximation $[\mathrm{P}] \approx [\mathrm{S}]$ will result in $\rho_\mathrm{P} \ll \rho_\mathrm{S} < 1$.

3.3 Preconditioners for Schwarz Method

So far, the SM and PSM were derived in the deterministic context, consistently with the MC method. As discussed in the previous section, the preconditioner P is expected to yield a reduction of the spectral radius associated with the iteration operator $[\mathrm{L}_\mathrm{P}]$. The computational cost associated with determining and applying the preconditioner at each iteration is another aspect to consider. For instance, preconditioning with $[\mathrm{P}] = [\mathrm{S}]$ is ideal in terms of efficiency but is not practical for problem in higher dimension since the determination of $[\mathrm{P}]$ has a comparable or larger cost than solving the original problem. Consequently, the preconditioning of SM often considers geometric preconditioners based on mesh coarsening rather than defining $[\mathrm{P}] \approx [\mathrm{S}]$ (see [121], for example). However, the situation is different in the stochastic case, where dedicating significant efforts to determine the preconditioner $[\mathrm{P}]$ may subsequently save considerable computational resources if the preconditioner is employed multiple times to solve a large number of problem samples.

3.3.1 Deterministic Preconditioners

Let $\hat{\kappa} \approx \kappa$ be an approximation of the coefficient field. Selecting $\hat{\kappa}$ to minimize directly the averaged distance between $[\mathrm{S}]$ and $[\hat{\mathrm{S}}]$ is a difficult problem; as a proxy, one often simply selects $\hat{\kappa}$ to be representative of the future samples $\kappa^{(m)}$ using the expectation ($\mathbb{E}[\boldsymbol{\kappa}]$) or median ($\bar{\kappa}$) of the stochastic field $\boldsymbol{\kappa}$. In this approach, $[\mathrm{S}](\hat{\kappa})$ can be assembled combining elementary solutions of the local problems, namely $\mathcal{P}^{(d)}((1,0),0,\hat{\kappa})$ and $\mathcal{P}^{(d)}((0,1),0,\hat{\kappa})$ in our particular 1D case, contributing for an efficient parallel construction.

This idea is, in particular, supporting the construction of *deterministic* preconditioners. Here we call deterministic (sample-independent) preconditioner a preconditioner $[\mathrm{P}]$ which is not depending on the sample κ considered. The main advantage of such deterministic preconditioners is that they can be computed (and possibly factorized) once for all before initiating the MC sampling stage.

Although preconditioners based on fixed statistics can be very cheap to re-use at each sample, they can also be very ineffective if the sampled coefficient field is too different from the statistic used. For example, the particular case where $\hat{\boldsymbol{\kappa}} = \bar{\kappa}$ is a very poor approximation for a field with large variance or low correlation length. Therefore, one should not expect the resulting preconditioner $[\bar{\mathrm{S}}]$ to provide large acceleration rates. The convergence rate of the $[\bar{\mathrm{S}}]$-based method is denoted by

$$\boldsymbol{\rho}_0 \doteq \boldsymbol{\rho}\left([\mathrm{L}_{\bar{\kappa}}]\right). \tag{3.24}$$

3.3.2 Stochastic Preconditioners

Another type of preconditioners is the so-called *stochastic* preconditioners. This type of preconditioners is based on the random field $\hat{\boldsymbol{\kappa}}$, a finite representation of the field $\boldsymbol{\kappa}$. Here, we specifically focus on preconditioner $[\hat{\mathbf{S}}] \doteq \mathrm{S}(\hat{\boldsymbol{\kappa}})$, which consists of the Schwarz matrix for a particular parameter $\hat{\boldsymbol{\kappa}}$. This preconditioner is then ideal for solving the elliptic problem corresponding to $\boldsymbol{\kappa} = \hat{\boldsymbol{\kappa}}$, and it is expected to efficiently accelerate the SM provided that the subsequent MC samples $\kappa^{(m)}$ remain close to $\hat{\kappa}^{(m)}$ in the sense that $[\mathrm{S}](\kappa^{(m)}) \approx [\hat{\mathrm{S}}] \doteq [\mathrm{S}](\hat{\kappa}^{(m)})$ for all sampled $\kappa^{(m)}$. In order to construct this preconditioner, we introduce a finite representation $\hat{\boldsymbol{\kappa}} \approx \boldsymbol{\kappa}$ based on the Karhunen-Loève (KL) expansion of $\hat{\boldsymbol{\kappa}}$.

3.3.2.1 KL-expansion

The KL expansion of a stochastic process is a spectral decomposition according to its covariance matrix [65, 79, 70]. Let $\Xi^{N_{KL}}$ be a set of independent and jointly standard normal distributed events, i.e., $\boldsymbol{\xi} \doteq (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{N_{KL}}) \in \Xi^{N_{KL}}$, such that $\boldsymbol{\xi}_i \sim N(0,1)$ and i.i.d. Each sample from $\boldsymbol{\xi}$ is denoted as $\xi^{(m)} = (\xi_1^{(m)}, ..., \xi_{N_{KL}}^{(m)})$. The truncated KL expansion of $\boldsymbol{g}$ generated by $\boldsymbol{\xi}$ is denoted by $\hat{\boldsymbol{g}}(x, \boldsymbol{\xi}) \in L^2(\Omega, \Xi^{N_{KL}})$ and defined as,

$$\hat{\boldsymbol{g}}(x, \boldsymbol{\xi}) \doteq \mu + \sum_{i=1}^{N_{KL}} \sqrt{\lambda_i}\phi_i(x)\boldsymbol{\xi}_i, \tag{3.25}$$

where the set of eigenpairs $(\lambda_i, \phi_i(x))$ is the solution of the generalised eigenvalue problem

$$\int_\Omega C(x, x')\phi_i(x')\, dx' = \lambda_i \phi_i(x), \quad \text{with } \langle \phi_i, \phi_j \rangle = \delta_{i,j} \text{ and } \lambda_i \geq \lambda_{i+1}. \tag{3.26}$$

We denote the number of stochastic dimensions of discrete space that approximates $L^2(\Omega, \Xi^{N_{KL}})$ by $N_{KL} \in \mathbb{N}$, and define the KL-truncation of $\boldsymbol{\kappa}$ by N_{KL} terms as $\hat{\boldsymbol{\kappa}} \doteq \exp \hat{\boldsymbol{g}}$. Clearly, $\hat{\boldsymbol{\kappa}}$ converges to $\boldsymbol{\kappa}$ as N_{KL} increases.

3.3.2.2 KL-based Preconditioner

Using the above construction of the field $\hat{\boldsymbol{\kappa}}$, we can construct the KL-based preconditioner

$$[\hat{\mathbf{S}}] = [\mathbf{S}](\hat{\boldsymbol{\kappa}}^{(m)}), \tag{3.27}$$

where $\hat{\boldsymbol{\kappa}}$ is the KL expansion of $\boldsymbol{\kappa}$. Clearly, the following convergence result holds,

$$[\hat{\mathbf{S}}] \to [\mathbf{S}] \quad \text{as} \quad \hat{\boldsymbol{\kappa}} \to \boldsymbol{\kappa}. \tag{3.28}$$

The convergence rate of the $[\hat{\mathrm{S}}]$-based method is denoted by

$$\boldsymbol{\rho}_{N_{KL}} \doteq \boldsymbol{\rho}\left([\mathrm{L}_{\hat{\mathrm{S}}}]\right). \tag{3.29}$$

If we want to compute a large number of samples from the solution of problem (3.4), we need

to assemble the same amount of preconditioners. According to definition (3.14), constructing $[\hat{\mathrm{S}}]^{(m)}$ amounts to solve $2D$ local problems (3.10) per sample, which is a significant computational effort at a pre-processing stage, and in particular, it may be more demanding than solving the problem itself using a deterministic preconditioning approach. Therefore, using the stochastic preconditioner $[\hat{\mathbf{S}}]$ is not a feasible approach, and instead, we proceed by constructing a surrogate $[\widetilde{\mathbf{S}}] \approx [\hat{\mathbf{S}}]$, that is easily adaptive to each sample $\kappa^{(m)}$.

3.3.2.3 PC-based Preconditioner

The surrogate of the KL-based preconditioner given by the Polynomial Chaos (PC) expansion is defined as,

$$[\hat{\mathbf{S}}](\boldsymbol{\xi}) \approx [\widetilde{\mathbf{S}}](\boldsymbol{\xi}) \doteq \sum_{\alpha=0}^{J} [\mathrm{S}_\alpha] \boldsymbol{\Psi}_\alpha(\boldsymbol{\xi}),\ \boldsymbol{\xi} \in \Xi^{N_{KL}}, \tag{3.30}$$

where $\{\boldsymbol{\Psi}_\alpha(\boldsymbol{\xi})\}_{\alpha=0}^{J}$ are orthonormal polynomials that form a basis of $L^2(\Xi^{N_{KL}})$, and $[\widetilde{\mathbf{S}}]$ are preconditioner modes defined by the Non-Intrusive (NI) orthogonal projection

$$[\mathrm{S}_\alpha] \doteq \langle [\hat{\mathbf{S}}](\boldsymbol{\xi}), \boldsymbol{\Psi}_\alpha(\boldsymbol{\xi}) \rangle_{\Xi^{N_{KL}}} = \int_{\Xi^{N_{KL}}} [\hat{\mathbf{S}}](\boldsymbol{\xi}) \boldsymbol{\Psi}_\alpha(\boldsymbol{\xi}) \mathrm{p}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{3.31}$$

with $\mathrm{p}(\boldsymbol{\xi})$ a standard joint-normal probability density function. To approximate equation (3.31), we use a Sparse Grid Pseudo Spectral Projection (PSP) [133] quadrature rule of level l given by

$$[\mathrm{S}_\alpha] \approx Q_l([\hat{\mathbf{S}}], \boldsymbol{\Psi}) = \sum_{q=0}^{N_l} [\hat{\mathrm{S}}](\boldsymbol{\eta}_q^l) \Psi_\alpha(\boldsymbol{\eta}_q^l) w_q^l, \tag{3.32}$$

where $\boldsymbol{\eta}_q^l \in \mathbb{R}^{N_{KL}}$ and w_q^l are the quadrature nodes and weights, respectively. The deterministic polynomials Ψ_α are realisations of $\boldsymbol{\Psi}_\alpha$ at the quadrature nodes. The PSP level l is adapted to each polynomial Ψ_α. We write $[\widetilde{\mathbf{S}}]^l$ when we want to stress the PSP level of the surrogate.

Most of the weights w_q^l are zero for most polynomials, and the quadrature nodes for coarse levels are nested into the finer ones, which means the number of preconditioner modes computed equals the number of nodes at the finest level. The number of terms of the PC expan-

sion (3.30) is based on the total-order degree rule, and it is given by

$$J = \frac{(N_{KL} + N_o)!}{N_{KL}! N_o!}, \tag{3.33}$$

where N_o denotes the polynomial order. Each $[\hat{\mathrm{S}}]$ matrix has an associated cost of solving 2 sets of local problems (3.10) in parallel (because each subdomain has 2 internal boundaries). Therefore, assembling preconditioner $[\widetilde{\mathbf{S}}]$ entails fiding N_l coefficients $[\hat{\mathrm{S}}]$, with an associated cost of $2N_l$. For a large number of solutions M, the surrogate approach becomes much cheaper than the exact one. The number of quadrature nodes N_l is dependent on the number of stochastic dimensions N_{KL} and polynomial order N_o.

3.4 Numerical Results

We have presented three different types of preconditioners. The median-based $[\bar{\mathrm{S}}]$ is known for its cheap implementation. However, it may perform poorly if non-smooth fields are considered. Next, we presented the KL-based preconditioner $[\hat{\mathbf{S}}]$, a stochastic preconditioner that provides significantly better acceleration for fields with high variance or low correlation. However, it is not suited for practical implementation since its construction cost is equivalent to construct $[\mathbf{S}]$. Finally, we presented a $[\widetilde{\mathbf{S}}]$, a PC expansion of $[\hat{\mathbf{S}}]$ with a significantly reduced cost of construction w.r.t. $[\hat{\mathbf{S}}]$.

In this section we compare the above three preconditioners with a one-dimensional test case. The test case uses a uniform mesh of $N_e = 1005$ elements of the unit interval, partitioned in $D = 20$ subdomains with 5 overlapping elements. For the stochastic field $\boldsymbol{\kappa}(x, \theta)$, we consider a stationary log-normal distribution, defined from a Gaussian field $\boldsymbol{g}(x, \theta)$ with zero mean, variance σ^2, and squared-exponential covariance, i.e., $\lambda = 2$.

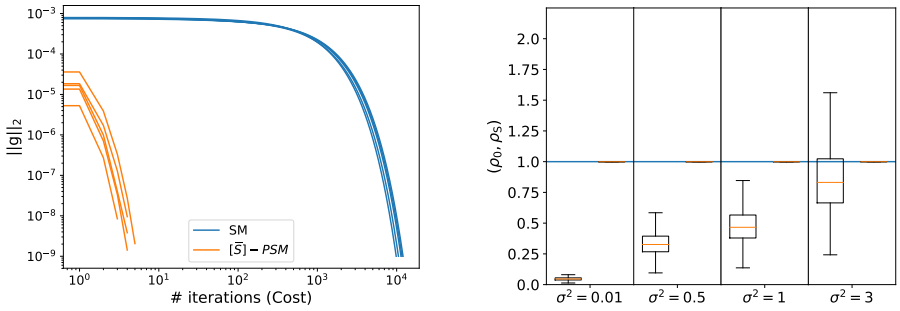
3.4.1 Median-based Preconditioner

We start this discussion by illustrating the performance of the $[\bar{\mathrm{S}}]$-PSM. The analysis is split into two parts. First, we illustrate the significant acceleration provided by $[\bar{\mathrm{S}}]$ using low variance.

Figure 3.2a shows the preconditioner's effect on reducing the gap norm $\|\mathrm{g}^k\|$ the number of iterations (cost). The figure illustrates 5 realizations of $\boldsymbol{\kappa}$ randomly generated for a field with $\sigma^2 = 0.01$ and $\ell_c = 0.1$. In this case, the SM needs about 10,000 iterations to get $\|\mathrm{g}^k\| < 10^{-9}$ when about 5 iterations on averaged (out of 10,000 samples) are needed to achieve the same reduction using PSM with $[\bar{\mathrm{S}}]$. Since the two methods have a comparable computational cost per iteration (assuming the cost of applying the preconditioner in line 3 of Algorithm 2 negligible compared to the call to SM-ITERATION for the resolution of the local problems), the preconditioner has, in this case, a speed-up of 100. A finer characterization of the two methods consists of comparing the spectral radius of their respective iteration. Owing to the low dimensionality of the problem reduced problem ($N_\Gamma = 40$), the matrices $[\mathrm{L_S}]$ and $[\mathrm{L}_{\bar{\kappa}}]$ can be assembled, and their respective spectral radius computed for each sample of $\boldsymbol{\kappa}$. Using 10,000 samples, we found $\mathbb{E}[\boldsymbol{\rho}_\mathrm{S}] \approx 0.999$ and $\mathbb{E}[\boldsymbol{\rho}_0] \approx 0.05$ ($\sigma^2 = 0.01$), denoting that the PSM reduces the gap at each iteration by a factor that is 20 times larger.

Now, we study the performance of the $[\bar{\mathrm{S}}]$-PSM for increasing variance. As discussed above, the more representative the sampled field $\hat{\kappa}$ is, the more efficient the preconditioner will be. Conversely, it is expected that the efficiency of the preconditioner degrades as $\hat{\kappa}$ becomes less and less representative. Figure 3.2b highlights this effect when using $\hat{\kappa} = \bar{\kappa}$. To this end, we consider an increasing variance such that the mean distance $||\boldsymbol{\kappa} - \bar{\kappa}||_\Omega$ augments. The figure shows the boxplot of the distributions of the spectral radius of $(\boldsymbol{\rho}_\mathrm{S}, \boldsymbol{\rho}_0)$ for variances $\sigma^2 = 0.01, 0.5, 1$ and 3. The box corresponds to the 50% probability range, with a line at the median value, and the whiskers encompass 99.825% probability range. The boxplots are estimated from 10,000 samples of $\boldsymbol{\kappa}$ (the remaining 0.175% outliers are not shown here) in these tests. We first observe that $\mathbb{E}[\boldsymbol{\rho}_0]$ is increasing with σ^2, denoting the degradation of the preconditioner performance. Second, although the median of the spectral radius remains less than one over the range of variance considered, we observe that the spread of its distribution also grows with σ^2. Eventually, for $\sigma^2 = 3$, the probability of any $\boldsymbol{\rho}_0$ to be larger than 1 is close to 0.25. This means that the preconditioned iterations will diverge for one out of four samples. In contrast, the boxplots of $\boldsymbol{\rho}_\mathrm{S}$ for the SM, also reported in the figure, are hardly distinguishable at the plot scale: $\mathbb{E}[\boldsymbol{\rho}_\mathrm{S}] \sim 1$ with a very low variability for all the values of σ^2.

To better understand the degradation with the variance of $[\bar{\mathrm{S}}]$-PSM, Figure 3.3 illustrates the



(a) Norm of the gap g^k with the iteration index for 5 samples with $\sigma^2 = 0.01$.

(b) Boxplots of the spectral radius ρ_0 and ρ_S considering different σ^2.

Figure 3.2: Comparison of the convergence between the classical SM and the $[\bar{S}]$-PSM.

compounded set of complex eigenvalues the $[\bar{S}]$-PSM operator for 3,000 samples $\kappa^{(m)}$. These eigenvalues are shown for the four values of σ^2 considered in Figure 3.2a. Black dots represent the eigenvalues associated with a particular sample $\kappa^{(m)}$for each variance value. The plots show that for a low variance, *i.e.*, when $\bar{\kappa}$ is representative of all $\kappa^{(m)}$, the eigenvalues are all clustered around the origin. When σ^2 increases to 0.5, $\bar{\kappa}$ becomes less representative, and the eigenvalues start to spread but remain inside the unit circle: the PSM remains convergent and more effective than the SM. For $\sigma^2 = 1$, all eigenvalues for the $3,000$ samples are still inside the unit circle, but some of them have modulus very close to one, denoting that the PSM barely improve over the SM for these samples (recall that $\mathbb{E}[\rho_S] \sim 1$). Finally, many eigenvalues fall outside the unit circle when $\sigma^2 = 3$, underlying a significant probability of samples with unstable PSM iterations.

3.4.2 KL-based Preconditioner

In the previous subsection, we have seen that the acceleration provided by the $[\bar{S}]$-PSM degrades significantly as the variance of the field increases. In addition, $[\bar{S}]$-PSM iterations do not convergence for a significant number of samples if σ^2 is moderately large. Similar behaviour is expected if ℓ_c or γ goes smaller, since then the distance $\|\bar{\kappa} - \boldsymbol{\kappa}\|$ would again grow larger. To rectify this behaviour, we suggested a preconditioner $[\hat{S}] = [S](\hat{\kappa}^{(m)})$, where $\hat{\kappa}^{(m)}$ replicates the fluctuations of each $\kappa^{(m)}$ better than $\bar{\kappa}$.

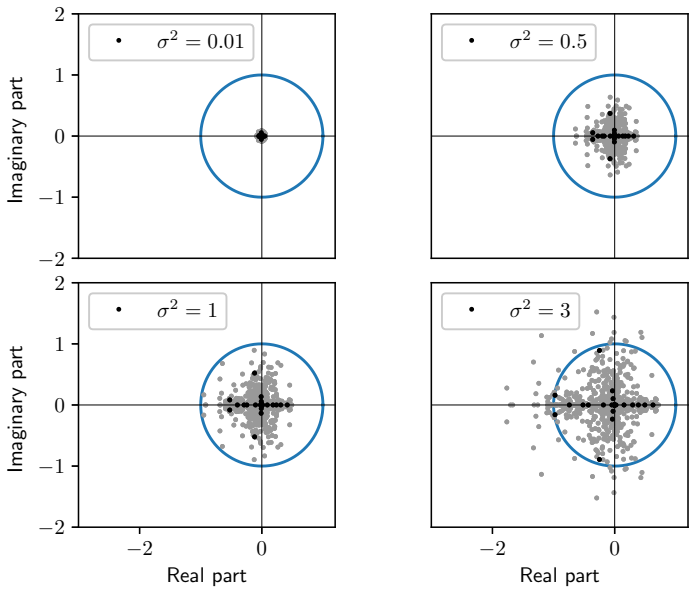
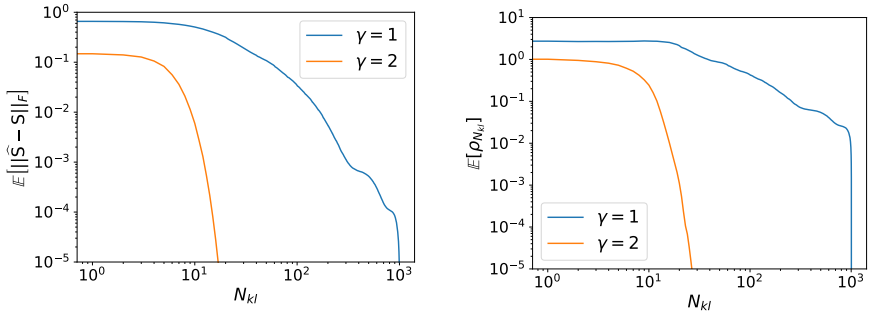


Figure 3.3: Complex eigenvalues $[\mathrm{L}_{\bar{\kappa}}]$ for 3,000 samples $\kappa^{(m)}$ and different values of σ^2. The spectrum of a particular sample (seed) is shown in black.

Figure 3.4 shows the convergence of $[\hat{\mathbf{S}}]$ to $[\mathbf{S}]$ and the corresponding acceleration, w.r.t. the number of KL terms used and for different field roughness, $\gamma = 1$ and $\gamma = 2$. The variation of the average error $\mathbb{E}\left[\|[\mathbf{S}] - [\hat{\mathbf{S}}]\|_F\right]$ is illustrated in Figure 3.4a. The decay of the curve corresponding to $\gamma = 1$ is significantly slower than the one corresponding to $\gamma = 2$. While for $\gamma = 2$ the error is of the order $\mathcal{O}(1^{-6})$ using $N_{KL} = 18$, for $\gamma = 1$ we need to use $N_{KL} = 996$ to achieve the same accuracy. The slower decay of the eigenvalues explains this behaviour in the case of $\gamma = 2$ w.r.t. $\gamma = 1$.

The convergence of the surrogate directly influences the rate of convergence given by the preconditioner, illustrated in Figure 3.4b. Once again, the curve corresponding to $\gamma = 1$ has a significant slower decay than the corresponding curve for $\gamma = 2$. An average gap reduction of $10e^{-5}$ per iteration is achieved using $N_{KL} = 1006$ and $N_{KL} = 28$ for $\gamma = 1$ and $\gamma = 2$, respectively. As results for $\gamma = 2$ demand significantly less computationally effort, we restrict the subsequent results in this section to this case without loss of generality.

Similarly to the analysis done for the $[\bar{\mathrm{S}}]$ preconditioner, in Figure 3.5a, we illustrate the



(a) Error $\mathbb{E}\left[\|[\mathbf{S}] - [\hat{\mathbf{S}}]\|_F\right]$. (b) Average spectral radius $\mathbb{E}\left[\boldsymbol{\rho}_{N_{KL}}\right]$.

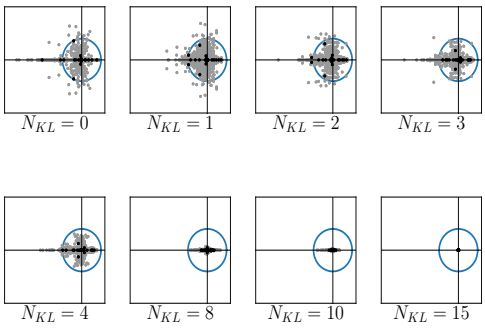
Figure 3.4: Variation of the error against $[\mathbf{S}]$ (left), and $\mathbb{E}\left[\boldsymbol{\rho}_{N_{KL}}\right]$ (right), with N_{KL}. The $\gamma = 1$ case yields curves with slower decay compared with the $\gamma = 2$ case.

spectrum of a set of $3,000$ samples of matrices $[\mathrm{L_P}]$. The variance used is $\sigma^2 = 4$ will certainly generate large unstable samples for $[\bar{\mathrm{S}}]$-PSM, which motivates the use of $[\hat{\mathrm{S}}]$-PSM. For $N_{KL} \leq 4$ we can identify a significant number of eigenvalues outside the unit circle. The distribution of the spectrum tends to squeeze towards the origin as N_{KL} increases. We can eventually include all eigenvalues from $3,000$ matrices inside the circle at $N_{KL} = 8$. This converging behaviour explains the decay of $\mathbb{E}\left[\boldsymbol{\rho}_{N_{KL}}\right]$ and the increasing number of stable samples with N_{KL}.

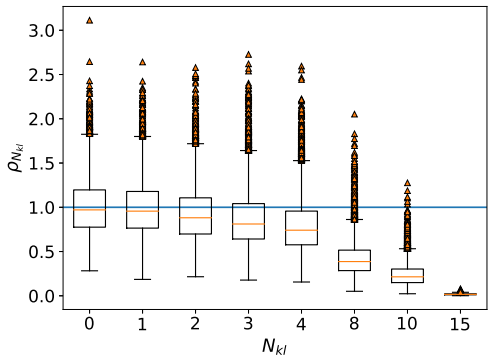
Figure 3.5b illustrates the relation between the distribution of $\boldsymbol{\rho}_{N_{KL}}$ and the average acceleration provided by a fixed number of N_{KL} modes. For each N_{KL} we find $10,000$ matrices $[\mathrm{L_P}]^{(m)}$, compute the corresponding $\rho^{(m)}_{N_{KL}}$ and plot this value. The case of $N_{KL} = 0$ corresponds to $[\bar{\mathrm{S}}]$-PSM, having about 50% of unstable samples. As N_{KL} decreases, the distribution of $\boldsymbol{\rho}_{\mathrm{P}}$ squeezes, and the mean value decreases, leading to improved stability and rate of convergence. Consistently with Figure 3.5a, for $N_{KL} \leq 8$ we ensure an almost sure stability, with less 0.175% of unstable samples.

3.4.3 PC-based Preconditioner

The median-based preconditioner presented two main issues: a) its acceleration degrades as the variance increases, and b) for moderate variance, the stability of the method is compromised. We presented a preconditioner based on the KL expansion of $\boldsymbol{\kappa}$ and overcomes these two issues using high enough N_{KL}. However, the construction of each matrix $[\hat{\mathrm{S}}]$ has a similar



(a) The spectrum of 3000 matrices. A particular sample is highlighted in dark.



(b) Distribution of $\boldsymbol{\rho}_{N_{KL}}$ for different N_{KL}.

Figure 3.5: Representation of 3000 spectra of $[\mathrm{L}_{\hat{\mathrm{S}}}]$ w.r.t. different N_{KL}. Each complex plane exhibits the spectrum of 3000 matrices (left). Each boxplot iillustrates the distribution of 3000 spectral radii (right).

cost of computing the matrix $[\mathrm{S}](\kappa^{(m)})$. To alleviate this computational burden, we introduce a surrogate of the KL-based preconditioner, $[\widetilde{\mathbf{S}}] \approx [\hat{\mathbf{S}}]$. Each sample $[\widetilde{\mathbf{S}}]$ will now be much easier to compute, giving rise to a fast and adaptive PSM.

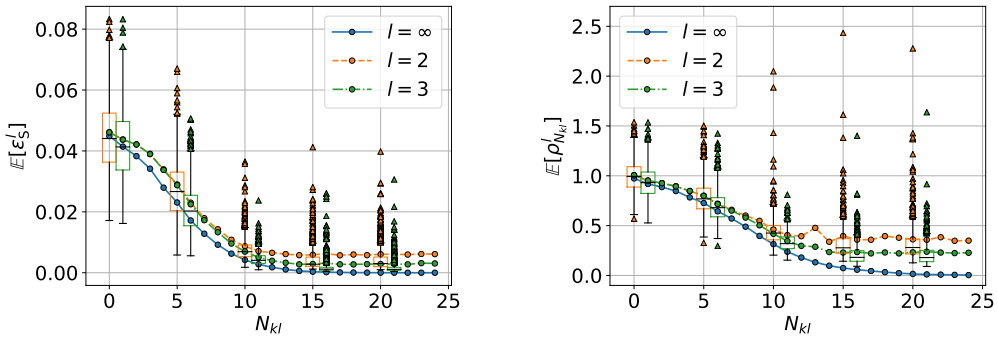
The accuracy and the cost of expansion (3.30) is most influenced by two parameters: the number of stochastic dimensions N_{KL} and the PSP level l. The larger the magnitude of these parameters, the more accurate and the more costly the preconditioners become.

Consider the average relative error defined by,

$$\epsilon_{\mathrm{S}}^{l} := \sqrt{\frac{\mathbb{E}\left[\|[\mathbf{S}] - [\widetilde{\mathbf{S}}]^{l}\|_{\Omega}\right]}{\mathbb{E}\left[\|[\mathbf{S}]\|_{\Omega}\right]}}, \tag{3.34}$$

where $\epsilon_{\mathrm{S}}^{\infty}$ corresponds to the PC truncation error, i.e., $[\widetilde{\mathbf{S}}]^{l} = [\hat{\mathbf{S}}]$.

Figure 3.6a represents the variation of the average error $\epsilon_{\mathrm{S}}^{l}$ for $l = 2, 3$ and ∞ over $3,000$ samples for different N_{KL}. The curve corresponding to $l = \infty$ has a steady decay until $N_{KL} = 15$, when it stabilizes near zero, i.e., when the KL truncation error is nearly zero at $N_{KL} = 15$ ($[\hat{\mathbf{S}}] \approx [\mathbf{S}]$). This behaviour goes according to the results in the previous subsection. The other two curves represent the cases of $l = 2, 3$. These two curves have a steady decay until $N_{KL} = 10$. From this point, the curve corresponding to $l = 2$ stabilizes while the case of $l = 3$



(a) Relative error.

(b) Convergence rate.

Figure 3.6: Relative error (left) and convergence rate (right) comparisons between KL-based and corresponding PC-based PSM.

continues its decreasing behaviour and eventually also stabilizes at $N_{KL} = 15$. The inflexion point occurs when the surrogate error is dominated by the PC truncation error and not by the KL truncation error. This justifies that the error corresponding to $l = 3$ is lower than $l = 2$. The PSP level also influences the range of the distribution of $\epsilon^l_{\tilde{\mathrm{S}}}$. For $l = 3$ this distribution is narrower, which is again explained by the higher accuracy of the PC expansion.

Now, we analyse the average reduction of g^k. The convergence rate of the $[\widetilde{\mathrm{S}}]$-based method is denoted by

$$\boldsymbol{\rho}^l_{N_{KL}} \doteq \boldsymbol{\rho}\left([\mathrm{L}_{\tilde{\mathrm{S}}}]\right), \tag{3.35}$$

where, for consistency, $\boldsymbol{\rho}^\infty_{N_{KL}} = \boldsymbol{\rho}_{N_{KL}}$. Figure 3.6b illustrates the variation of the average acceleration of $[\widetilde{\mathbf{S}}]^l$-PSM for the three previous cases $l = 2, 3, \infty$. The three curves behave similarly to the ones in Figure 3.6a. Using $N_{KL} = 15$ we can achieve a reduction of g^k of approximately 0.25 using $l = 3$. The acceleration provided for $l = 2$ is nearly doubled using $l = 3$. The distribution of the eigenvalues is also narrower for larger PSP level, contributing to improved stability of the method. Only a few outliers can be found above the unit, which means that the method is robust for a substantial percentage of samples.

We now turn into a closer look at the accuracy, cost, and performance of the PSM for fixed N_{KL}. To do so, we introduce a few notation that will help the analysis. Let ϵ_κ and $\epsilon^l_{\tilde{\mathrm{S}}}$ be defined

as

$$\epsilon_\kappa := \sqrt{\frac{\mathbb{E}\left[\|\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}}\|_\Omega\right]}{\mathbb{E}\left[\|\boldsymbol{\kappa}\|_\Omega\right]}}, \qquad \epsilon^l_{\hat{\mathrm{S}}} := \sqrt{\frac{\mathbb{E}\left[\|[\hat{\mathbf{S}}] - [\widetilde{\mathbf{S}}]^l\|_\Omega\right]}{\mathbb{E}\left[\|[\mathbf{S}]\|_\Omega\right]}}, \tag{3.36}$$

denote the average relative error of coefficient w.r.t. its KL representation, and the average relative error of the preconditioner $[\hat{\mathrm{S}}]$ and w.r.t. its surrogate, respectively. Using ϵ^l_{S} and $\epsilon^l_{\hat{\mathrm{S}}}$, we can get a bound of the error between the surrogate and the ideal preconditioner. These errors are described in Table 3.1, using $100,000$ samples. The accuracy of both the fields and the preconditioners decreases with N_{KL}. Moreover, as expected from Figure 3.6a, the errors for $l = 3$ are smaller than the ones with respect to $l = 2$. We can see a stagnation of the errors for $N_{KL} \geq 10$. The cost of building a surrogate is also depicted in Table 3.1. We give the number of PSP nodes used for each level and N_{KL}. As expected, the cost of the PC expansion increases with N_{KL}. For $N_{KL} = 5$, it is 5 times more costly to build $[\widetilde{\mathbf{S}}]^{l=3}$ than $[\widetilde{\mathbf{S}}]^{l=2}$, and as the number of stochastic dimensions increases to $N_{KL} = 15$, it becomes 10 times more expensive to do $[\widetilde{\mathbf{S}}]^{l=3}$. Whether to go for larger PSP level necessarily depends on the complexity of the problem and the number of samples needed.

N_{KL}		0	2	5	10	15
ϵ_κ		9.97×10^{-1}	9.49×10^{-1}	6.33×10^{-1}	1.3×10^{-1}	1.24×10^{-2}
ϵ^l_{S}		4.58×10^{-2}	4.44×10^{-2}	3.37×10^{-2}	9.17×10^{-3}	1.08×10^{-3}
$\epsilon^l_{\hat{\mathrm{S}}}$	$l=2$	-	1.36×10^{-3}	3.9×10^{-3}	5.82×10^{-3}	5.96×10^{-3}
	$l=3$	-	6.32×10^{-4}	1.82×10^{-3}	2.78×10^{-3}	2.85×10^{-3}
$J-(N_l)$	$l=2$	-	$8-(17)$	$26-(71)$	$76-(241)$	$151-(511)$
	$l=3$	-	$20-(49)$	$96-(351)$	$416-(2001)$	$1086-(5951)$

Table 3.1: Comparison of the accuracy of the different KL-based preconditioners and corresponding surrogates, against S. Average values over 100 000 samples.

Besides its slow convergence, the SM is stable for all samples. On the contrary, the previous results show that we cannot guarantee convergence of the PSM with probability 1. This motivates the introduction of the effective spectral radius, denoted by $\rho^l_{\text{eff}} := \min\{\rho_{\mathrm{S}}, \rho^l_{N_{KL}}\}$, and the average effective improvement of the convergence of PSM w.r.t. SM, denoted by,

$$\mathrm{EI}^l_{\rho_{\text{eff}}} \doteq \mathbb{E}\left[\frac{\boldsymbol{\rho}^l_{\text{eff}}}{\boldsymbol{\rho}_{\mathrm{S}}}\right]. \tag{3.37}$$

We also compare the improvement on the number of iterations preformed from PSM to SM. Let $\mathrm{g}_{\mathrm{S}}^{n_{\mathrm{S}}}, \mathrm{g}_{\mathrm{P}}^{n_{\mathrm{P}}}$ denote the gap obtained by performing n_{S} (resp. n_{P}) SM (resp. PSM) iterations. Also, assume $n_{\mathrm{S}}, n_{\mathrm{P}}$ are such that $\|\mathrm{g}_{\mathrm{S}}^{n_{\mathrm{S}}}\|_{\Omega} \approx \|\mathrm{g}_{\mathrm{P}}^{n_{\mathrm{P}}}\|_{\Omega}$. Then, the following holds,

$$\mathrm{g}_i^{n_i} \approx \mathrm{g}\,\mathrm{L}_i^{n_i}, \text{ for } i = [\mathrm{S}], [\mathrm{P}].$$

This means, in particular, that $n_{\mathrm{S}} \approx \mathrm{EI}_{\#\mathrm{iter}}^{l} n_{\mathrm{P}}$ with

$$\mathrm{EI}_{\#\mathrm{iter}}^{l} := \mathbb{E}\left[\frac{\log(\boldsymbol{\rho}_{\mathrm{eff}}^{l})}{\log(\boldsymbol{\rho}_{\mathrm{S}})}\right],$$

which is called the average effective improvement in the number of iterations preformed from SM to PSM. We should expect to perform $\mathrm{EI}_{\#\mathrm{iter}}^{l}$ SM iterations to reduce the gap as much as we would with a single PSM iteration.

Table 3.2 summarises the effective spectral radius and the effective improvement in the number of iterations. It also gives the number of unstable samples of each PSM. Due to the low dimensionality of the problem, we can compute the spectral radius of $[\mathrm{L}_{\mathrm{P}}]$ for each sample. If a sample is unstable, we perform SM. Results for $N_{KL} = 15$ are considerably better then for $N_{KL} = 0$. This improvement is partially justified by the better convergence, i.e., a smaller spectral radius. It is also supported by the significant reduction of the number of unstable samples. Indeed, the more unstable samples there are, the more SM iterations are performed, which significantly deteriorates the performance of the method.

One could argue whether, for this simple 1-D test case, we may actually need to perform $50,000$ SM iterations. In fact, the average number of SM iterations to compute $1,000$ solutions with a tolerate of $\|\mathrm{g}_{\mathrm{S}}\|_{\Omega} < 10^{-10}$ is over $110,000$ SM iterations. Moreover, there are a few samples that take over a million SM iterations. Therefore, it is very likely that we need to perform more than $50,000$ SM iterations for this test case. For smaller N_{KL}, the performance of the surrogates is similar to the one of the corresponding KL-based preconditioner. For larger N_{KL} the performance is still better, but the improvement is not as significant for the surrogate as it is of its corresponding KL-based preconditioner, especially for $l = 2$. This is explained by the stagnation of the errors, illustrated in Figure 3.6a and confirmed in Figure 3.6b. Most of the

improvement is due to the decrease on the number of unstable samples rather the average gap reduction.

N_{KL}		0	2	5	10	15
% of unstable samples	KL-based	46%	36%	15%	0.07%	<0.01%
	PSP level 2	-	36%	15%	0.5%	0.6%
	PSP level 3	-	36%	15%	0.1%	0.05%
$\mathrm{EI}_{\rho^l_{\mathrm{eff}}}$	KL-based	8.78×10^{-1}	8.33×10^{-1}	6.8×10^{-1}	2.39×10^{-1}	1.75×10^{-2}
	PSP level 2	-	8.33×10^{-1}	6.8×10^{-1}	2.71×10^{-1}	1.2×10^{-1}
	PSP level 3	-	8.33×10^{-1}	6.8×10^{-1}	2.48×10^{-1}	5.82×10^{-2}
$\mathrm{EI}_{\#\mathrm{iter}}$	KL-based	564	1145	4122	18919	49648
	PSP level 2	-	1129	4003	13357	18997
	PSP level 3	-	1141	4109	16189	27376

Table 3.2: Comparison of the performance and cost of PSM and $[\hat{\mathrm{S}}]$-PSM for different stochastic dimensions and PSP levels.Average values over 100 000 samples.

Finally, we conclude that the increase in performance largely pays off the cost of building a surrogate. On the one hand, to reduce $\|\mathrm{g}\|$ by the same amount as $[\bar{\mathrm{S}}]$-PSM, we should do on average ≈ 500 SM iterations. The $[\bar{\mathrm{S}}]$-PSM has the minimal cost of just computing a single preconditioner. On the other hand, we should do ≈ 27000 SM iterations to have the same gain as $[\widetilde{\mathrm{S}}]$-PSM with $N_{KL} = 15$ and $l = 3$. The associated cost is ≈ 6000 preconditioners. The choice between which method to choose will depend on the number of samples needed to compute. However, if this number is very high, and the field presents moderately high variance, high roughness or even low correlation, it is worth going for the $[\widetilde{\mathrm{S}}]$-PSM approach.

3.5 Conclusion and Discussion

In this work, we presented a stochastic preconditioner for the additive Schwarz method. Using this preconditioner, we managed to speed up the resolution of sampled elliptic SPDEs, which allowed the computation of significantly more accurate statistics estimates. This strategy is particularly suitable for non-smooth parameters.

The cost of solving a sampled problem is a major obstacle to determine accurate estimates using sampling methods. The acceleration of the resolution of each sampled problem is therefore of utmost interest in the UQ community. We focused on MC methods, but our strategy can be used in other methods based on model evaluations, such as non-intrusive spectral meth-

ods. In our case, generating an MC sample amounts to solve the boundary-to-boundary system (3.14) by PSM. A classical preconditioning strategy uses $[\mathrm{P}] = [\bar{\mathrm{S}}]$, which can lead to poor performances if the coefficient field is not smooth enough. We presented a stochastic preconditioner $[\mathrm{P}] = [\widetilde{\mathbf{S}}]$, that is easily adapted to each sample and is much more efficient than $[\bar{\mathrm{S}}]$, particularly for non-smooth fields.

The stochastic preconditioner is a PC expansion of a KL-based preconditioner. The stochastic preconditioner is constructed at an off-line stage before the MC sample. The construction of this PC expansion is very efficient because each coefficient is constructed through local solves (see Appendix A). The cost of constructing this PC expansion is equivalent to find N_l KL-based preconditioners.

The performance of both $[\bar{\mathrm{S}}]$ and $[\widetilde{\mathbf{S}}]$ preconditioners was tested on a one-dimensional test case. We have shown that the high acceleration rates that characterize the $[\bar{\mathrm{S}}]$-PSM for smooth fields significantly deteriorate as smoothness degrades. Opposite to that, the $[\widetilde{\mathbf{S}}]$ preconditioner remains effective provided that the number of random variables is sufficiently large. The tests conducted in this section suggest that an iteration of the $[\bar{\mathrm{S}}]$-PSM is equivalent to 500 iterations of the classical SM. However, using $10 \leq N_{KL} \leq 15$, an SM iteration is equivalent to between $16,000$ and $27,000$ $[\widetilde{\mathbf{S}}]$-PSM iterations. Although the cost of constructing $[\widetilde{\mathbf{S}}]$ is N_l times larger than constructing $[\bar{\mathrm{S}}]$, the acceleration provided can compensate for the higher cost at the off-line stage, particularly if the number of MC samples is large enough.

The low dimensionality of this problem allowed an extensive spectral analysis of the main operators that characterize the convergence of the PSM method. This analysis prepared the ground for an extension to a two-dimensional application. Results showed that even for $N_{KL} = 15$ we can still find unstable samples, suggesting that almost sure stability is never guaranteed. This number of variables is not particularly problematic for the current test problem. However, the number of random variables necessary to have a sufficiently stable method may become too large in higher-dimensional problems. Therefore, we may expect the construction of an effective stochastic preconditioner based on a global PC expansion to be unfeasible in two dimensions. To resolve this issue, we intend to exploit the construction of $[\widetilde{\mathbf{S}}]$ based on local PC expansions, as presented in [31, 32]. In these works, the authors use a non-overlapping DD setting. The corresponding boundary-to-boundary operator is the well known Schur complement

matrix [105]. The structure of the Schur complement matrix allows the construction of the corresponding surrogate through blocks. A local PC expansion represents each block, and each local PC expansion is defined on a subdomain, exploiting independent local parameterizations. The parallel construction is possible due to the independence between each local PC expansion. The locality of this construction has been proven to yield cheap and accurate surrogates, even for non-smooth fields. As a result, this local construction will be the building block to construct effective stochastic preconditioners for iterative DD methods in future works.

Chapter 4

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

The surrogate-based preconditioners presented in the previous chapter have two flaws: a) the number of random variables required is undesirably large, and b) the preconditioning approach is unstable for some samples. This chapter presents a surrogate-based preconditioning strategy for the resolution of many sampled Schur complement systems. The novel strategy uses a minimal number of local random variables, exploiting the DD-KL approach introduced in [31]. Moreover, the novel preconditioning approach is guaranteed to converge for all samples. This contribution performs numerical experiments on a two-dimensional example with a practical interest in many real case problems. *This chapter is adapted from Reis, João F. and Le Maître, Olivier P. and Congedo, Pietro M. and Mycek, Paul (2021). Stochastic Preconditioning of Domain Decomposition Methods for Elliptic Equations with Random Coefficients,* Computer Methods in Applied Mechanics and Engineering.

4.1 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, ...), making 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 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 [17, 77]. 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 [27] 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 [56, 70] which have been applied to numerous linear and non-linear PDEs with random coefficients [34, 71, 72, 51, 66, 4, 47, 88]. 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 func-

tional 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 [91, 93, 92, 28, 9, 10, 122, 24]). Some construction methods for functional approximations, often termed non-intrusive methods, rely on samples (observations) of the model solution (*e.g.*, regression methods [14] and spectral projection methods [99, 107, 29, 30]). 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.* [102, 112, 72] 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 [103, 120, 119, 121]), 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 [31, 32]. 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 [113]. The spatial discretization is a standard finite-element (FE) method, but the approach proposed in the chapter 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 resulting 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 [105, 127, 84]. The Schur problem's size is much smaller than the original problem and can be solved iteratively without explicitly forming the Schur system. 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 precondi-

tioner 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 favours 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 [103], 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 [32], 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 Polynomial Chaos (PC) expansions. The use of local parametrizations to reduce the stochastic dimension follows ideas similar to the works in [22, 104, 63, 126]. The numerical results of [32] 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 [32] 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.

4.2 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.

4.2.1 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

$$\forall f, g \in L^2(\Omega), \quad \langle f, g \rangle_\Omega \doteq \int_\Omega f(x) g(x)\, dx, \quad \|f\|_\Omega = \langle f, f \rangle_\Omega^{1/2} < +\infty. \tag{4.1}$$

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

$$H^1(\Omega) \doteq \left\{ f(x) \in L^2(\Omega) : \partial_{x_i} f(x) \in L^2(\Omega),\ i = 1, \dots, n \right\}. \tag{4.2}$$

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

$$\mathbb{E}[\boldsymbol{u}] \doteq \int_\Theta \boldsymbol{u}(\theta)\, d\mu_\Theta(\theta). \tag{4.3}$$

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

$$\forall \boldsymbol{u}, \boldsymbol{v} \in L^2(\Theta), \quad \langle \boldsymbol{u}, \boldsymbol{v} \rangle_\Theta \doteq \mathbb{E}[\boldsymbol{u}\boldsymbol{v}], \tag{4.4}$$

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

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\Omega \times \Theta} \doteq \mathbb{E}\left[\langle \boldsymbol{u}(x, \theta), \boldsymbol{v}(x, \theta) \rangle_\Omega \right]. \tag{4.5}$$

4.2.2 Stochastic Elliptic Equation

The stochastic elliptic equation we are interested in has the form

$$\begin{aligned} &\nabla \cdot [\boldsymbol{\kappa}(x, \theta) \nabla \boldsymbol{u}(x, \theta)] = -f(x) \quad x \in \Omega,\, \theta \in \Theta \\ &\boldsymbol{u}(x, \theta) = 0, \quad x \in \partial\Omega,\, \theta \in \Theta, \end{aligned} \tag{4.6}$$

where $f(x)$ is a deterministic source and $\boldsymbol{\kappa}$ is the stochastic coefficient field of the equation. The equalities in the equations of (4.6) 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 (4.6) with the modified right-hand-side $-f(x) - \nabla \cdot [\boldsymbol{\kappa}(x, \theta) \nabla \boldsymbol{u}_{BC}(x, \theta)]$.

Problem (4.6) 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 [21]. In this work, we restrict ourselves to the case of $\boldsymbol{\kappa}$ being a stationary log-normal stochastic process, whose $\log$ is a centred Gaussian process $\boldsymbol{g}$ with covariance function C:

$$\boldsymbol{g}(x, \theta) \doteq \log \boldsymbol{\kappa}(x, \theta) \sim \mathcal{N}(0, C). \tag{4.7}$$

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

$$C(x, x') \doteq \sigma^2 \exp\left(-\frac{\|x - x'\|_\Omega^\gamma}{\gamma {\ell_c}^\gamma} \right), \tag{4.8}$$

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

4.2.3 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 (4.6). 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

$$z(\boldsymbol{u}) = \int_{\Omega' \subset \Omega} \boldsymbol{u}(y, \theta) dy. \tag{4.9}$$

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

$$\mathbb{E}\left[z(\boldsymbol{u})\right] \approx \frac{1}{M} \sum_{m=1}^{M} z\left(u^{(m)}\right), \tag{4.10}$$

where $u^{(m)} \in H^1(\Omega)$ is a random sample of the solution $\boldsymbol{u}(x, \theta)$ of (4.6) for the sampled coefficient value $\kappa^{(m)}$:

$$\begin{aligned} &\nabla \cdot \left(\kappa^{(m)} \nabla u^{(m)}\right) = -f, \quad x \in \Omega, \\ &u^{(m)} = 0, \quad x \in \partial\Omega. \end{aligned} \tag{4.11}$$

The random estimate in (4.10) is unbiased, provided that the $\kappa^{(m)}$ are drawn randomly and has an error whose variance is $\mathbb{V}\left[z(\boldsymbol{u})\right]/M$. 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.

4.2.4 Domain Decomposition and the Schur Complement System

Let us now introduce a divide to parallelize strategy to compute the solution of each problem (4.11). 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 (4.11).

Consider a partition of Ω into D subdomains $\Omega^{(d)}$, each with boundary $\partial\Omega^{(d)}$, where $\overline{\cup_{d=1}^{D} \Omega^{(d)}} =$

Ω. 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 (4.11) can be reduced to determining u_Γ such that the solutions w_d of the local problems,

$$\begin{aligned} &\nabla \cdot [\kappa \nabla w_d] = -f \quad x \in \Omega^{(d)}, \\ &w_d = u_\Gamma, \quad x \in \Gamma^{(d)}, \\ &w_d = 0, \quad x \in \overline{\Omega^{(d)}} \cap \partial\Omega, \end{aligned} \tag{4.12}$$

satisfying some compatibility conditions at the internal boundaries.

Now we introduce the particular domain decomposition (DD) method used in this work. We start by introducing the discrete version of the deterministic problem (4.11). 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 N_{nod}. We denote by $\{\Phi_l\}_{l=1}^{N_{\text{nod}}}$ the finite element basis and approximate the solution u as

$$H^1(\Omega) \ni u(x) \approx \sum_{l=1}^{N_{\text{nod}}} \Phi_l(x) u_l, \tag{4.13}$$

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 $N_\Gamma^{(d)}$. 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 (not in $\partial\Omega$) that belong to the $\mathcal{N} \setminus \mathcal{N}_\Gamma$. Proceeding with a FE discretization and Galerkin approach [32], and dropping again the sample index, problem (4.11) can be recast in the finite-dimensional system

$$[\mathrm{A}]\mathrm{u} = \mathrm{b}, \tag{4.14}$$

where the solution is defined as vector of nodal values $\mathrm{u} = (u_1 \cdots u_{N_{\text{nod}}})^\top$. The FE matrix

$[\mathrm{A}] \in \mathbb{R}^{N_{\mathrm{nod}} \times N_{\mathrm{nod}}}$ is assumed to be symmetric positive definite (SPD) with entries

$$[\mathrm{A}]_{l,l'} = \int_{\Omega} \kappa(x) \nabla \Phi_l(x) \cdot \nabla \Phi_{l'}(x) dx, \tag{4.15}$$

while the right-hand side is

$$\mathrm{b}_l = \int_{\Omega} f(x) \Phi_l(x) dx. \tag{4.16}$$

System (4.14) can be reorganized in the form

$$\begin{bmatrix} [\mathrm{A}_{\Gamma,\Gamma}] & [\mathrm{A}_{\Gamma,\mathrm{in}}] \\ [\mathrm{A}_{\mathrm{in},\Gamma}] & [\mathrm{A}_{\mathrm{in},\mathrm{in}}] \end{bmatrix} \begin{bmatrix} \mathrm{u}_{\Gamma} \\ \mathrm{u}_{\mathrm{in}} \end{bmatrix} = \begin{bmatrix} \mathrm{b}_{\Gamma} \\ \mathrm{b}_{\mathrm{in}} \end{bmatrix}. \tag{4.17}$$

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

$$[\mathrm{S}] \doteq [\mathrm{A}_{\Gamma,\Gamma}] - [\mathrm{A}_{\Gamma,\mathrm{in}}][\mathrm{A}_{\mathrm{in},\mathrm{in}}]^{-1}[\mathrm{A}_{\mathrm{in},\Gamma}]. \tag{4.18}$$

This gives the Schur system

$$[\mathrm{S}]\mathrm{u}_{\Gamma} = \mathrm{b}_{\mathrm{S}}, \quad \mathrm{b}_{\mathrm{S}} \doteq \mathrm{b}_{\Gamma} - [\mathrm{A}_{\Gamma,\mathrm{in}}][\mathrm{A}_{\mathrm{in},\mathrm{in}}]^{-1}\mathrm{b}_{\mathrm{in}}. \tag{4.19}$$

Classically, system (4.19) is solved by a matrix-free iterative method, since applying $[\mathrm{S}]$ to a given iterate u_{Γ} amounts to solving local problems (expressed by the matrix operator $[\mathrm{A}_{\mathrm{in},\mathrm{in}}]^{-1}$). The matrix $[\mathrm{S}]$ is SPD such that classical Conjugate Gradient (CG) methods can be applied. In practice, the conditioning of $[\mathrm{S}]$ 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 (4.19) [105, 127, 84]. In the following, we introduce different kinds of preconditioning strategies suited for solving the deterministic problems (4.11) for many samples.

4.3 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 (4.6). In the DD approach, each solution sample is computed by solving the Schur complement associated with the sample's deterministic problem in (4.11). Therefore, we have to solve many Schur systems (4.19) corresponding to different samples of $\boldsymbol{\kappa}$. In practice, solving for each realization of $\boldsymbol{\kappa}$ the Schur system (4.19) by a direct approach is costly as it demands to solve many local problems to assemble $[\mathrm{S}]$. Instead, it is usually more effective to solve (4.19) iteratively without assembling $[\mathrm{S}]$. 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.

4.3.1 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 $[\bar{\mathrm{S}}]$ the deterministic preconditioner, we want to ensure that $[\mathbf{S}](\theta)^{-1}[\bar{\mathrm{S}}]$ is close to the identity for almost all events θ. One straightforward choice is to define $[\bar{\mathrm{S}}]$ 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 $[\bar{\mathrm{S}}]$ using the average or median value of the stochastic coefficient field. In the following, we shall consider the deterministic preconditioner $[\bar{\mathrm{S}}]$ constructed on the median $\bar{\kappa}$ of $\boldsymbol{\kappa}$, therefore the notation $[\bar{\mathrm{S}}] = [\mathrm{S}](\bar{\kappa})$. Specifically, $[\bar{\mathrm{S}}]$ corresponds to the reduction of the matrix $\overline{[\mathrm{A}]}$ in (4.18), with entries

$$\overline{[\mathrm{A}]}_{l,l'} \doteq \int_{\Omega} \bar{\kappa}(x) \nabla \Phi_l(x) \cdot \nabla \Phi_{l'}(x) dx \quad \forall\, l, l' \in \mathcal{N}. \tag{4.20}$$

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

As will be evidenced in Section 4.4, 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.

4.3.2 Stochastic Schur Complement System

The stochastic Schur complement system is the stochastic counterpart of system (4.19). The solution of problem (4.6) is again approximated in a FE space V^h using *random nodal values* $\boldsymbol{u}_l(\theta)$:

$$\boldsymbol{u}(x,\theta) \approx \sum_{l=1}^{N_{\mathrm{nod}}} \Phi_l(x)\boldsymbol{u}_l(\theta) \in V^h \times L^2(\Theta) \subset H^1(\Omega) \times L^2(\Theta). \tag{4.21}$$

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

$$[\mathbf{S}](\theta)\mathrm{u}_\Gamma(\theta) = \mathrm{b}_\mathrm{S}(\theta), \tag{4.22}$$

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

$$[\mathrm{A}]_{l,l'}(\theta) = \int_\Omega \boldsymbol{\kappa}(x,\theta)\nabla\Phi_l(x)\cdot\nabla\Phi_{l'}(x)dx \quad \forall\, l,l' \in \mathcal{N}. \tag{4.23}$$

Following [32], the Schur complement system can be written as the sum of local contributions from individual subdomains:

$$[\mathbf{S}](\theta) = \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top [\mathbf{S}^{(d)}](\theta)[\mathrm{R}^{(d)}], \tag{4.24}$$

where $[\mathbf{S}^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma^{(d)}}$ denotes the stochastic *influence matrix* of subdomain $\Omega^{(d)}$, and $[\mathrm{R}^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma}$ is the so-called restriction operator, which is a deterministic matrix that maps global internal boundary nodes to local boundary nodes of $\Omega^{(d)}$. The influence matrix $[\mathbf{S}^{(d)}]$ is the boundary-to-boundary operator of the local stochastic problem, see [32] for more details.

The interest in the representation of the stochastic Schur matrix in (4.24) 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.

4.3.3 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 $\Omega^{(d)}$:

$$\boldsymbol{g}^{(d)}(x,\theta) \approx \hat{g}^{(d)}(x,\theta) \doteq \sum_{i=1}^{N_{KL}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta), \quad x \in \Omega^{(d)}, \tag{4.25}$$

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

$$\boldsymbol{\kappa}^{(d)}(x,\theta) \approx \hat{\boldsymbol{\kappa}}^{(d)}(x,\theta) \doteq \exp\left[\sum_{i=1}^{N_{KL}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta)\right], \tag{4.26}$$

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 $N_{KL}^{(d)}$ dominant modes of the coefficient will affect the error in the approximation of the influence matrix: the larger $N_{KL}^{(d)}$, the closer $[\mathbf{S}^{(d)}]$ and $[\widehat{\mathbf{S}}^{(d)}]$. In fact, as illustrated in Section 4.4, $N_{KL}^{(d)}$ 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 $N_{KL}^{(d)} = 0$ for all subdomains results in the deterministic preconditioner $[\bar{\mathrm{S}}]$.

For $N_{KL}^{(d)} \geq 1$, we now have to approximate the dependencies of $[\widehat{\mathbf{S}}^{(d)}]$ on the vector of $N_{KL}^{(d)}$ 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 $L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{KL}})$ defined by

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

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

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

We remark that for any $\boldsymbol{f}, \boldsymbol{g} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{KL}})$, $\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\boldsymbol{\xi}} = \langle \boldsymbol{f}(\boldsymbol{\xi}), \boldsymbol{g}(\boldsymbol{\xi}) \rangle_{\Theta}$, so that $\boldsymbol{f} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{KL}}) \iff \boldsymbol{f}(\boldsymbol{\xi}) \in L^2(\Theta)$. Following [132, 18], we introduce the Polynomial Chaos basis of $L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{KL}})$. Since the random variables $\boldsymbol{\xi}_i$ are independent and follow the standard Gaussian distribution, the PC basis consists of the infinite set of orthonormal Hermite polynomials $\boldsymbol{\Psi}_{\alpha}^{N_{KL}}(\boldsymbol{\xi})$. The multi-variate Hermite polynomials are defined as products of univariate orthonormal Hermite polynomials, through

$$\boldsymbol{\Psi}_{\alpha}^{N_{KL}}(\boldsymbol{\xi}) = \prod_{j=1}^{N_{KL}} \boldsymbol{\varphi}_{\alpha_j}(\boldsymbol{\xi}_j), \tag{4.27}$$

where $\alpha = (\alpha_1, \cdots, \alpha_{N_{KL}}) \in \mathbb{N}^{N_{KL}}$ is a multi-index and $\boldsymbol{\varphi}_{\alpha_j}$ is the univariate Hermite polynomial of degree $\alpha_j \in \mathbb{N}$. Any function $\boldsymbol{f} \in L^2_{\boldsymbol{\xi}}(\mathbb{R}^{N_{KL}})$ has a PC expansion [132, 18] of the form

$$\boldsymbol{f}(\boldsymbol{\xi}) = \sum_{\alpha \in \mathbb{N}^{N_{KL}}} f_{\alpha} \boldsymbol{\Psi}_{\alpha}^{N_{KL}}(\boldsymbol{\xi}). \tag{4.28}$$

In practice, the PC expansion (4.28) must be finite, and a truncation of the series (4.28) 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:

$$\mathcal{B} \doteq \left\{ \alpha \in \mathbb{N}^{N_{KL}} : \max_{1 \leq j \leq N_{KL}} \{\alpha_j\} = \|\alpha\|_{\infty} \leq p \right\};$$

- total-degree:

$$\mathcal{B} \doteq \left\{ \alpha \in \mathbb{N}^{N_{KL}} : \sum_{j=1}^{N_{KL}} \alpha_j = \|\alpha\|_{\ell_1} \leq p \right\};$$

- hyperbolic-cross:

$$\mathcal{B} \doteq \left\{ \alpha : \prod_{j=1}^{N_{KL}} (\alpha_j + 1) \leq p + 1 \right\}.$$

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

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

has a finite number of terms $J = |\mathcal{B}|$. For a fixed p, the hyperbolic-cross truncation gives the smallest PC basis (see [9] for discussion on dimensionality), while the partial-degree truncation gives the largest one with $J = (p+1)^{N_{KL}}$.

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 [32]. 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 4.3.4, which are readily amenable to generic NI approaches. Thanks to the orthonormality of the PC basis, $\langle \boldsymbol{\Psi}_\alpha^{N_{KL}}, \boldsymbol{\Psi}_\beta^{N_{KL}} \rangle_{\boldsymbol{\xi}} = \delta_{\alpha,\beta}$, the coefficient f_α is given by

$$f_\alpha = \langle \boldsymbol{f}, \boldsymbol{\Psi}_\alpha^{N_{KL}} \rangle_{\boldsymbol{\xi}} = \int_{\mathbb{R}^{N_{KL}}} \boldsymbol{f}(\mathrm{y}) \boldsymbol{\Psi}_\alpha^{N_{KL}}(\mathrm{y}) p_{\boldsymbol{\xi}}(\mathrm{y}) d\mathrm{y}. \tag{4.29}$$

In our NI implementation, the integral in (4.29) is simply discretized using a N_{KL}-dimensional quadrature formula,

$$f_\alpha \approx \sum_{q=1}^{N_{\mathrm{Q}}} \boldsymbol{f}(\mathrm{y}_q) \boldsymbol{\Psi}_\alpha^{N_{KL}}(\mathrm{y}_q) w_q, \tag{4.30}$$

where $\mathrm{y}_q \in \mathbb{R}^{N_{KL}}$ 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:

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

This characteristic guarantees 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 $[\widehat{\mathbf{S}}^{(d)}]$, and reintroducing the subdomain index, we assume that all its entries are in $L^2(N_{KL}^{(d)})$; it comes

$$[\widetilde{\mathbf{S}}^{(d)}](\theta) = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{S}^{(d)}]_\alpha \mathbf{\Psi}_\alpha^{N_{KL}^{(d)}} \left(\boldsymbol{\xi}^{(d)}(\theta) \right), \tag{4.31}$$

with

$$[\mathrm{S}^{(d)}]_\alpha = \sum_{q=1}^{N_{\mathrm{Q}}^{(d)}} [\widehat{\mathrm{S}}^{(d)}]_q \mathbf{\Psi}_\alpha^{N_{KL}^{(d)}}(\mathrm{y}_q^{(d)}) w_q^{(d)}, \tag{4.32}$$

and where $[\widehat{\mathrm{S}}^{(d)}]_q$ is the realization of the influence matrix for the realization of $\hat{\boldsymbol{\kappa}}^{(d)}$ corresponding to $\boldsymbol{\xi}^{(d)} = \mathrm{y}_q$ in (4.26). In the following, we restrict ourselves to a uniform PC order p for all subdomains, while the number of local random variables $N_{KL}^{(d)}$ will be fixed or adapted for each subdomain depending on the numerical experiments. Finally, the formal expression of the stochastic preconditioner is

$$[\widetilde{\mathbf{S}}](\theta) \doteq \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top \, [\widetilde{\mathbf{S}}^{(d)}](\theta) \, [\mathrm{R}^{(d)}]. \tag{4.33}$$

For each realization $\kappa^{(m)}(x)$, the corresponding preconditioner is obtained using (4.33), where the constitutive influence matrices $[\widetilde{\mathbf{S}}^{(d)}](\theta)$ are evaluated using (4.31). The random variables $\boldsymbol{\xi}^{(d)}(\theta^{(m)})$ are computed by projecting $\log \kappa^{(m)}$ on the local KL modes (see Appendix B). 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

$$\boldsymbol{\xi}_i^{(d)}(\theta^{(m)}) = \frac{1}{\sqrt{\lambda_i^{(d)}}} \int_\Omega \log \left(\kappa^{(m)}(x) \right) \phi_i^{(d)}(x) \, dx, \quad \forall i = 1, \ldots, N_{KL}^{(d)} \text{ and } d = 1, \ldots, D. \tag{4.34}$$

In practice, the integrals in (4.34) are numerically approximated using the quadrature rule employed to discretize the (local) KL eigenvalue problem (B.3). In this chapter, we use element-wise constant quadrature to estimate (B.3).

In the rest of the chapter, we call the preconditioner defined by (4.33) and (4.31) 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 stability 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.

4.3.4 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.

4.3.4.1 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,

$$[\widehat{\mathbf{S}}^{(d)}] = [\mathbf{L}][\mathbf{D}][\mathbf{L}]^\top, \tag{4.35}$$

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

$$[\widehat{\mathbf{S}}^{(d)}] = [\mathbf{H}^{(d)}][\mathbf{H}^{(d)}]^{\top}, \quad [\mathbf{H}^{(d)}] = [\mathbf{L}][\Delta], \tag{4.36}$$

where $[\Delta] \doteq [\mathbf{D}]^{\frac{1}{2}}$ uses the non-negative square-roots 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 (4.36), we would approximate $[\widehat{\mathbf{S}}^{(d)}]$ by

$$[\widetilde{\mathbf{S}}^{(d)}] = [\widetilde{\mathbf{H}}^{(d)}][\widetilde{\mathbf{H}}^{(d)}]^{\top}, \tag{4.37}$$

which would be almost surely non-negative for all PC basis. Unfortunately, the convergence with p of the PC approximation in (4.37) 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 [114].

4.3.4.2 Orthogonal factorization

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

$$[\mathbf{S}^{(d)}] = [\mathbf{Q}][\mathbf{D}][\mathbf{Q}]^{\top} \tag{4.38}$$

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

$$[\mathbf{H}^{(d)}] = [\mathbf{Q}][\Delta], \tag{4.39}$$

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 [114]). However, we can easily overcome this issue by defining an alternative factorization as

$$[\widehat{\mathbf{S}}^{(d)}] = \left([\mathbf{H}^{(d)}]\right)^2, \quad [\mathbf{H}^{(d)}] \doteq [\mathbf{Q}][\Delta][\mathbf{Q}]^\top. \tag{4.40}$$

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

$$[\widetilde{\mathbf{H}}^{(d)}](\theta) = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_\alpha \boldsymbol{\Psi}_\alpha^{N_{KL}^{(d)}}(\boldsymbol{\xi}^{(d)}), \tag{4.41}$$

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

$$[\mathrm{H}^{(d)}]_\alpha = \sum_{q=1}^{N_{\mathrm{Q}}^{(d)}} [\mathrm{H}^{(d)}]_q \boldsymbol{\Psi}_\alpha^{(d)}(\mathrm{y}_q) w_q^{(d)}. \tag{4.42}$$

In (4.42), the factors $[\mathrm{H}^{(d)}]_q$ of the quadrature nodes are defined as

$$[\mathrm{H}^{(d)}]_q = [\mathrm{Q}]_q [\Delta]_q [\mathrm{Q}]_q^\top, \tag{4.43}$$

where $[\mathrm{Q}]_q$ and $[\Delta]_q$ are obtained from the decomposition of $[\widehat{\mathrm{S}}^{(d)}]_q$ 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 $[\widehat{\mathrm{S}}^{(d)}]_q$. Algorithm 3 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.

Algorithm 3 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 $[\mathrm{H}^{(d)}]_\alpha = [0]$; ▷ 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 (4.26)
    Compute $[\widehat{\mathrm{S}}^{(d)}]_q$; ▷ Set the influence matrix
    Solve $[\widehat{\mathrm{S}}^{(d)}]_q = [\mathrm{Q}][\mathrm{D}][\mathrm{Q}]^\top$; ▷ Decompose the influence matrix
    Set $[\mathrm{H}^{(d)}]_q = [\mathrm{Q}][\Delta][\mathrm{Q}]^\top$; ▷ Set the factor, see (4.43)
    for all PC mode α **do**
      $[\mathrm{H}^{(d)}]_\alpha \leftarrow [\mathrm{H}^{(d)}]_\alpha + [\mathrm{H}^{(d)}]_q \boldsymbol{\Psi}_\alpha^{N_{KL}^{(d)}}(\mathrm{y}_q) w_q$; ▷ Update PC modes, see (4.42)
    end for
  end for
  return $\{[\mathrm{H}^{(d)}]_\alpha\}$; ▷ Return the PC modes
end procedure

Finally, the PC approximation of the influence operator reads

$$[\widetilde{\mathbf{S}}^{(d)}](\theta) = \left([\widetilde{\mathbf{H}}^{(d)}](\theta)\right)^2 \doteq \left(\sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_\alpha \boldsymbol{\Psi}_\alpha^{N_{KL}^{(d)}}\left(\boldsymbol{\xi}^{(d)}(\theta)\right)\right)^2, \tag{4.44}$$

where $\boldsymbol{\xi}^{(d)}(\theta)$ is given by (4.34), while the corresponding preconditioner is expressed as

$$[\widetilde{\mathbf{S}}](\theta) = \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top \left(\sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_\alpha \boldsymbol{\Psi}_\alpha^{N_{KL}^{(d)}}\left(\boldsymbol{\xi}^{(d)}(\theta)\right)\right)^2 [\mathrm{R}^{(d)}]. \tag{4.45}$$

Hereafter, we call the preconditioner in (4.45) the Factorized PC (FPC) preconditioner and the corresponding CG method the Factorized PCG (FPCG) method.

4.3.5 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. For each sample $\kappa^{(m)}(x)$, the sampling procedure involves 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 (4.12), b) compute the projection on the local KL basis (4.34) 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 (4.44). 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 (4.45), and the resolution step is engaged. The Schur system (4.19) 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 4 summarizes the workflow for the resolution of one sample.

Algorithm 4 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 (4.12);
5: Set $\boldsymbol{\xi}^{(d)}$ by local projection; ▷ see (4.34)
6: Set $[\mathrm{H}^{(d)}] = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_\alpha \boldsymbol{\Psi}_\alpha^{N_{KL}^{(d)}} \left(\boldsymbol{\xi}^{(d)}(\theta)\right)$; ▷ Realization of factor (4.41)
7: Set $[\widetilde{\mathrm{S}}] \leftarrow [\widetilde{\mathrm{S}}] + [\mathrm{R}]^{(d)} \, [\mathrm{H}^{(d)}] \, [\mathrm{H}^{(d)}] \, \left([\mathrm{R}]^{(d)}\right)^\top$; ▷ 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**

Algorithm 4 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 [113]), until the convergence criterion is met, that is when $\|\mathrm{r}^k\|/\|\mathrm{b_S}\| < \mathrm{tol}$. Algorithm 4 does not show the computation of the system's right-hand-side $\mathrm{b_S}$; this computation involves local solves, following (4.19), 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. 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 4. 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 $[\bar{\mathrm{S}}]$ 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 4 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.

4.4 Numerical tests

This section numerically investigates the performance of the different preconditioning strategies: median-based, DPC, and FPC. As a test problem, we consider 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 (4.22) with size $N_\Gamma = 3{,}389$. The left plot of Fig. 4.1 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 $\ell_c = 0.05$.

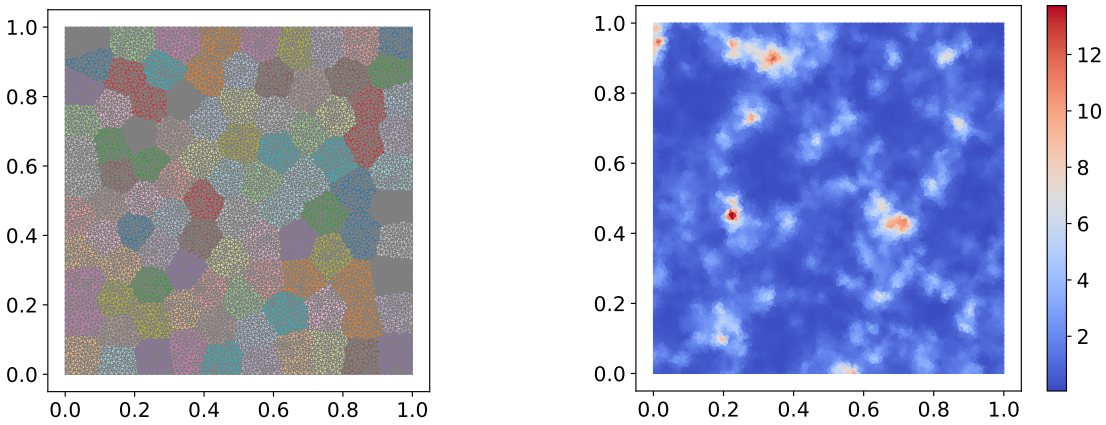


Figure 4.1: Finite Element mesh and partition of Ω in $D = 100$ subdomains (left), and a realization of $\boldsymbol{\kappa}$ for $\gamma = 1.2$, $\ell_c = 0.05$ and $\sigma^2 = 1$ (right).

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 ($\|\mathrm{r}_0\| > \mathrm{tol}\|\mathrm{b}_\mathrm{S}\|$). All numerical experiments presented in the chapter 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 initial guess $\mathrm{u}_\Gamma = 0$:

$$\boldsymbol{\rho} \doteq \frac{\text{\# MPCG iterations}}{\text{\# DPCG or FPCG iterations}}. \tag{4.46}$$

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.

4.4.1 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 4.2 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. 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 chapter. 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.

4.4.2 DPCG method

We now turn to the DPC preconditioner defined by equations (4.31)-(4.33). 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 behaviour 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 simplify the analysis, the number of local random variables in the approximation of $\boldsymbol{\kappa}$ is fixed to N_{KL} for all subdomains.

Figure 4.3 reports the evolution of the acceleration $\boldsymbol{\rho}$ of the DPCG method for different vari-

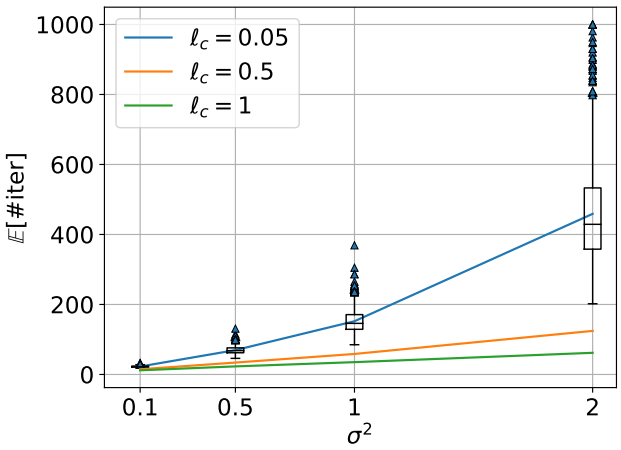
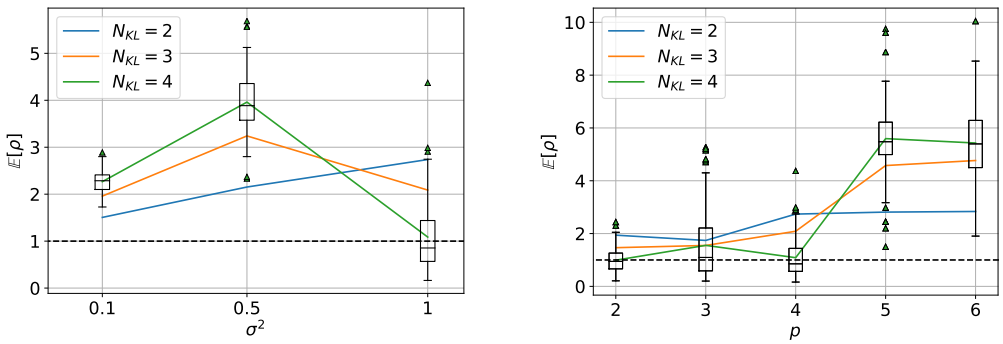


Figure 4.2: Average number of iterations to convergence (and corresponding boxplots for $\ell_c = 0.05$) in the MPCG method for different variances and correlation lengths. Case of $\gamma = 1.2$ with $D = 100$ subdomains.

ances 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. 4.3a, 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 $N_{KL} = 4$. The plot also shows that, for large σ^2, the acceleration deteriorates with N_{KL}. This behaviour 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 conclusions 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 precon-

ditioners 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 for 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 $N_{KL} = 4$). The PC truncation error becomes 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. 4.3b 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 $\sigma^2 > 1$.

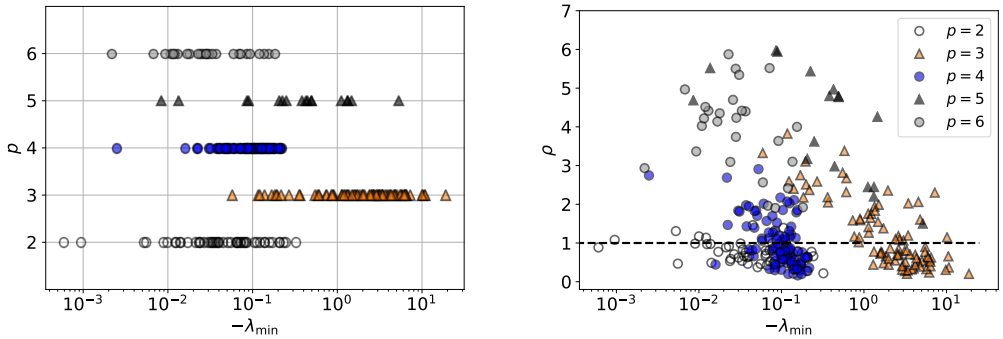


(a) Acceleration as a function of σ^2 ($p = 4$). (b) Acceleration as a function of p ($\sigma^2 = 1$).

Figure 4.3: Average acceleration of the DPCG method (and corresponding boxplots for $N_{KL} = 4$) 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 $D = 100$.

Figure 4.4 provides a more detailed spectral analysis of the DPC preconditioner. Figure 4.4a shows, for $N_{KL} = 4$, 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 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. 4.4b 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 eigenvalue is not far from 0.



(a) Smallest negative eigenvalues for different PC orders.

(b) Acceleration vs. smallest negative eigenvalue.

Figure 4.4: Spectral analysis of 100 DPC preconditioners for total-order truncation, with $N_{KL} = 4$.

To better understand the role of the PC truncation error, we compare in Fig. 4.5 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 $N_{KL} = 3$ 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 behaviour with odd/even degree effects, similar to the non-monotonous convergence reported in [32]. 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 incorporates much higher univariate degree polynomials at the expense of multivariate polynomials. For instance, in Fig. 4.5, the hyperbolic-cross truncation goes up to $p = 20$ while the partial degree truncation 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 behaviour, it is preferable to preserve the flexibility of arbitrary PC truncation strategies and to construct almost surely SPD stochastic preconditioners.

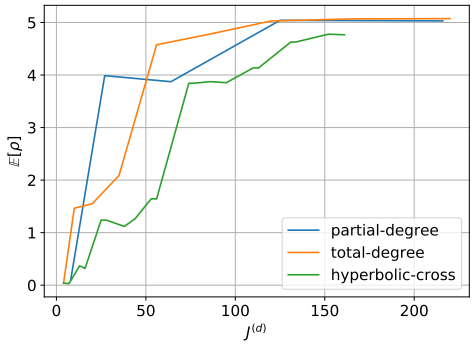


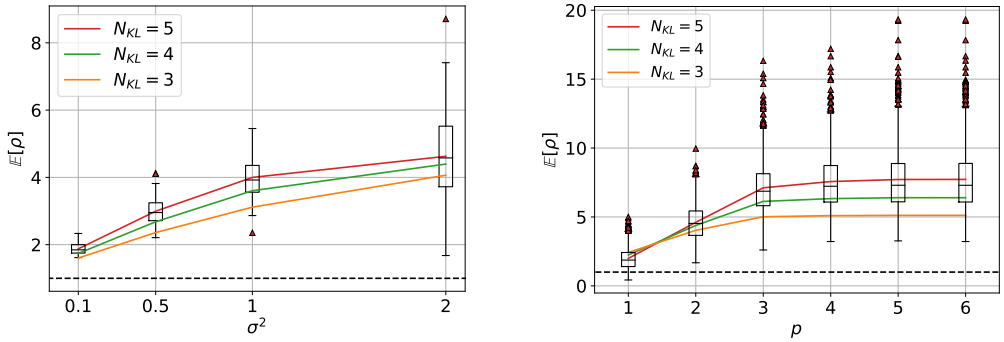
Figure 4.5: Average acceleration of the DPCG method as a function of the local PC bases dimension $J^{(d)}$ and for different PC truncations as indicated. Case of $N_{KL} = 3$ and $\sigma^2 = 1$, $\gamma = 2$ and $\ell_c = 0.05$.

4.4.3 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 4.6 reports the acceleration of the FPCG method for different variances σ^2, and local discretization parameters N_{KL} and p. Figure 4.6a 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. 4.3a. The average acceleration of the FPCG method remains greater than 3 for $\sigma^2 > 1$, and is much less significantly impacted compared to the DPCG case. Figure 4.6b 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 4.6b 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 deterministic preconditioning with the mean of the Schur system; it does not exactly coincide with the MPCG method that uses the Schur system associated with the median field, but the two methods are expected to achieve the same performance ($\boldsymbol{\rho} = 1$).



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

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

Figure 4.6: Average acceleration of the FPCG method, with corresponding boxplots for $N_{KL} = 5$. Case of $\gamma = 1.2$, $\ell_c = 0.05$, $D = 100$.

To complete the comparison with the DPCG method, Fig. 4.7 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 $N_{KL} = 3$. First, the FPCG acceleration is seen to achieve the asymptotic acceleration for much lower-dimensional bases (degree) compared to the case of DPCG method shown in Fig. 4.5. This much more satisfying behaviour 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 chapter 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.

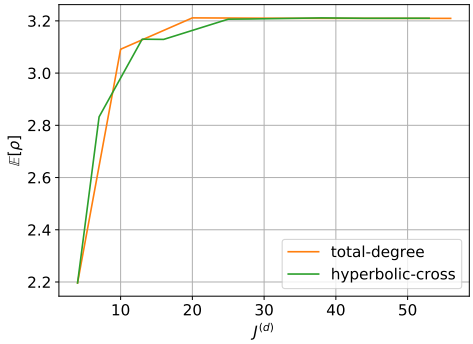
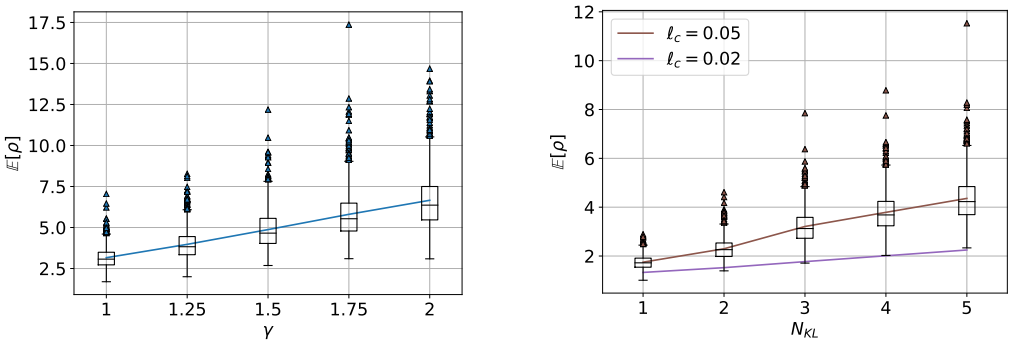


Figure 4.7: Average acceleration of the FPCG method as a function of the local basis dimension $J^{(d)}$ and hyperbolic-cross and total degree PC truncations. Case of $N_{KL} = 3$ and $\sigma^2 = 1$.

The analysis of the FPCG method continues with Fig. 4.8a, 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 $N_{KL} = 4$, 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 behaviour is expected since, for fixed N_{KL} and variance σ^2, the (local) KL truncation error reduces for increasing γ (see Appendix B). To further appreciate the effect of the KL truncation error, Fig. 4.8b shows the acceleration of increasing N_{KL} when $\gamma = 1.2$ and the variance as in Fig. 4.8a. Cases of $\ell_c = 0.02$ and $\ell_c = 0.05$ are reported. Consistently with the behaviour 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 B). 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 ($N_{KL} = 4, \ell_c = 0.05$). (b) Acceleration with N_{KL} ($\gamma = 1.2$).

Figure 4.8: 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 $p = 4$.

4.4.4 Influence of the number of subdomains

4.4.4.1 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 B, 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, $\ell_{\text{loc}} \doteq \ell_c / \operatorname{diam}(\Omega^{(d)})$, 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 subdomains with balanced sizes, their diameters will be $\operatorname{diam}(\Omega^{(d)}) \sim D^{1/n}$ such that $\ell_{\text{loc}} \sim \mathcal{O}(D^{-1/n})$ (recall that n is the number of spatial dimensions).

Figure 4.9 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 $N_{KL} = 4$ 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.

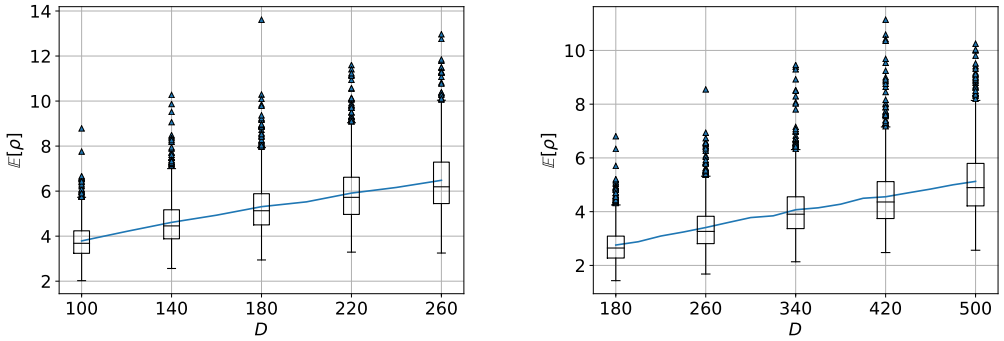


Figure 4.9: Average acceleration and corresponding boxplots of the FPCG method as a function of the number D of subdomains. Stochastic field $\boldsymbol{\kappa}$ with $\sigma^2 = 1$, $\gamma = 1.2$ and $\ell_c = 0.05$ (left) and $\ell_c = 0.02$ (right). Other parameters are $p = 4$ and $N_{KL} = 4$.

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 stagnation is not visible, for the range of values of D shown in Figure 4.9. 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. 4.12a), 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.

4.4.4.2 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. Fine control of the KL truncation error can be achieved by adapting the number of local KL modes, $N_{KL}^{(d)}$, in each subdomain. Specifically, in our settings, the fraction of energy R_{KL} of the Gaussian field accounted by the KL expansions is

$$R_{\mathrm{KL}} = \frac{\sum_{d=1}^{D} \sum_{i=1}^{N_{KL}^{(d)}} \lambda_i^{(d)}}{\sigma^2 |\Omega|}, \tag{4.47}$$

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 (5.18)). Tuning all the $N_{KL}^{(d)}$ to obtain a prescribed value for R_{KL} is not convenient. It is easier to rely on a local criterion and set $N_{KL}^{(d)}$ in each of the subdomains accordingly. Let us denote by $\tau \in (0, 1)$ the local tolerance on the KL error; we define $N_{KL}^{(d)}$ as the smallest positive integer such that

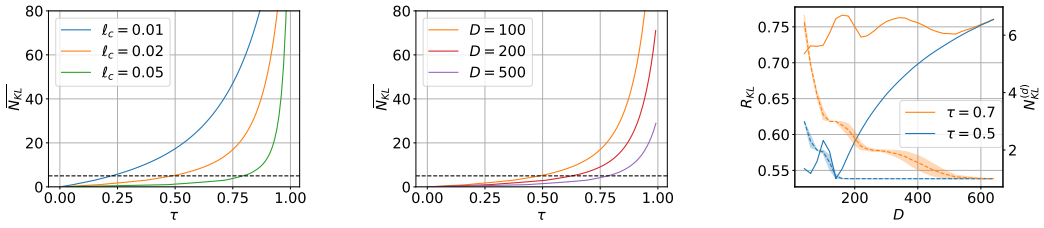
$$\sum_{i=1}^{N_{KL}^{(d)}} \lambda_i^{(d)} \geq \tau \sigma^2 |\Omega^{(d)}|. \tag{4.48}$$

One can easily check that (4.48) implies $R_{\mathrm{KL}} \geq \tau$. In other words, $1-\tau$ is an upper bound for the relative KL error. In the case of subdomains with roughly equal diameters, $N_{KL}^{(d)}$ satisfying (4.48) does not vary much from one subdomain to another and we introduce the average number of local modes

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

Figure 4.10a 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. 4.10b 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 (4.48) with $\tau = 0.6$ decreases from 7 to 3, owing to the reduction of the apparent correlation length ℓ_{loc}. Fig. 4.10c reports the resulting fraction of energy $R_{\mathrm{KL}}(\tau)$ and the number of local modes $N_{KL}^{(d)}$ 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 behaviour for the two τ are quite different. For $\tau = 0.5$ the average value of $N_{KL}^{(d)}$ quickly drops to one (left axis) and exhibits a low variability between subdomains (the extent of the shaded areas correspond to average value of $N_{KL}^{(d)} \pm$ the variance). After $N_{KL}^{(d)}$ 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 $\boldsymbol{\kappa}$.

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 4.11a reports the resulting average FPCG acceleration $\mathbb{E}[\boldsymbol{\rho}]$ (solid lines, left axis) and a 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 $N_{KL}^{(d)} = 1$ (see Fig. 4.10c) 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

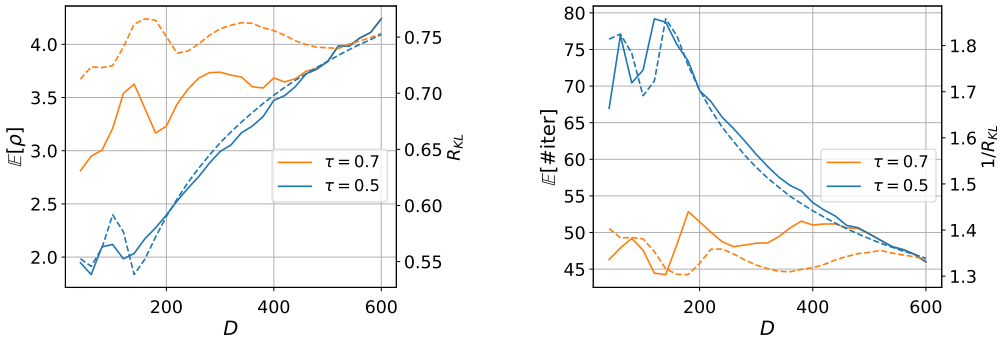


(a) Different ℓ_c ($D = 100$). (b) Different D ($\ell_c = 0.02$). (c) R_{KL} with D ($\ell_c = 0.05$).

Figure 4.10: 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 $N_{KL}^{(d)}$ (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 $N_{KL}^{(d)}$ around that mean. Case of $\gamma = 1.2$.

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. 4.11a 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 $N_{KL}^{(d)}$ 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. 4.11b, 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.

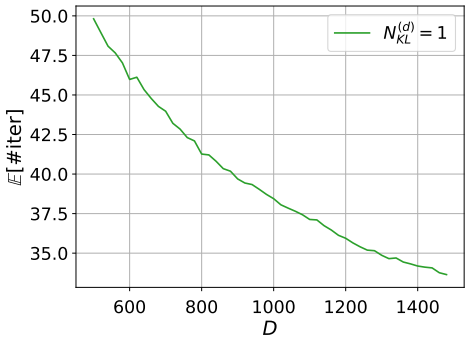
(b) # of FPCG iterations with D.

Figure 4.11: Performance of the FPCG method with the number of subdomains using local adaptation of $N_{KL}^{(d)}$. 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 $\ell_c = 0.05$.

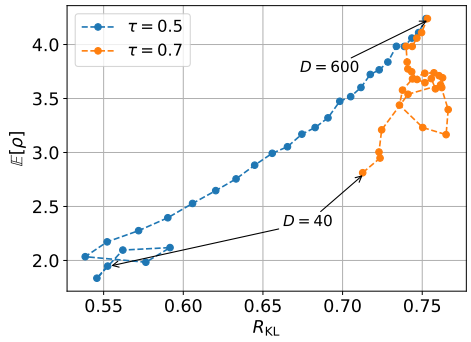
For $\tau = 0.7$, Fig. 4.12a shows that increasing the number of subdomains after $D = 600$, such that $N_{KL}^{(d)} = 1$ for all d, yields the same behaviour as for $\tau = 0.5$ and $D > 100$: a continuous decay of the number of iterations to convergence and therefore an improvement of the acceleration. The differences in the two regimes, before and after reaching $N_{KL}^{(d)} = 1$, are illustrated in Fig. 4.12b. 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 $N_{KL}^{(d)} = 1$. On the contrary, for $\tau = 0.5$ the relation between the two quantities is clear.

4.4.5 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



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



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

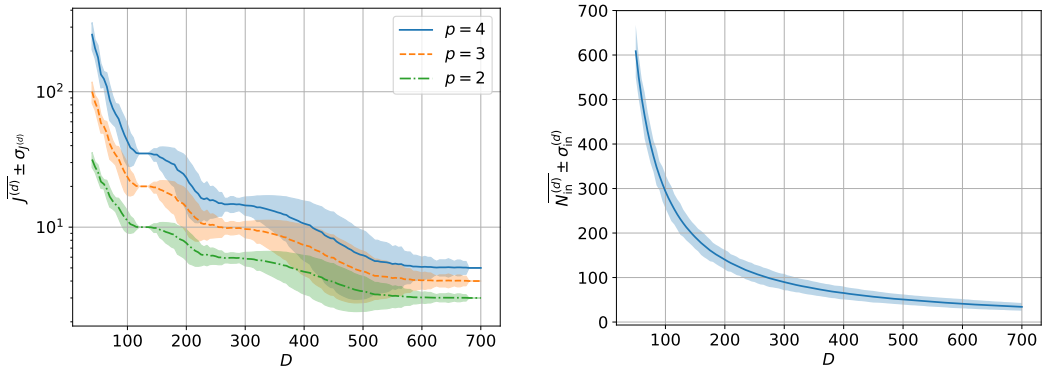
Figure 4.12: Performance of the FPCG method. Left: $\tau = 0.7$ and $D > 600$; Right: $\tau = 0.5$ and 0.7, $D \in [40, 600]$. Other parameters are $p = 4$, $\gamma = 1.2$, $\sigma^2 = 1$ and $\ell_c = 0.05$.

enough. As a starting point, one can assume that sufficiently many samples will be computed in the sampling stage to pay back 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 $N_{KL}^{(d)} > 1$, and the second regime where D and τ are such that $N_{KL}^{(d)} = 1$. We have seen that the average number of FPCG iterations is mostly controlled by $R_{\mathrm{KL}}(\tau)$, 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. 4.13 the evolution with D of the FPC preconditioner's complexity.

Figure 4.13a 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 $N_{KL}^{(d)}$, specifically $J^{(d)} = (p + N_{KL}^{(d)})!/p!N_{KL}^{(d)}!$. Increasing ℓ_{loc} through smaller subdomains allows for a reduction of $N_{KL}^{(d)}$ (see Fig. 4.10c) 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, $(p+1)^{N_{KL}^{(d)}}$, achieved through the reduction of $N_{KL}^{(d)}$ is impressive; even for 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. 4.13b 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.



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

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

Figure 4.13: Evolutions of the size $J^{(d)}$ of the local PC bases (for degrees 2 to 4) (left) and of the size $\mathrm{N}_{\text{in}}^{(d)}$ 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 $\gamma = 1.2$.

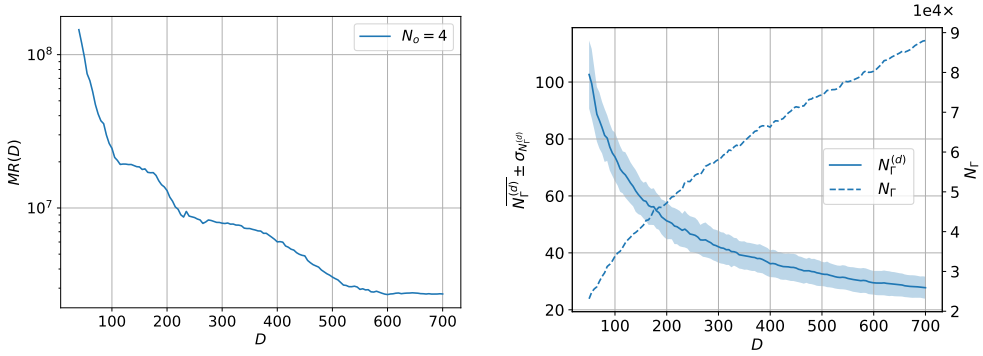
Then, as D increases, the computational complexity of the PC expansion of the factorized influence operator drops. However, as D increases, there are more and more PC expansions to store in order to subsequently assemble the realizations of the FPC preconditioner. The memory requirement is, therefore a concern. Since the PC coefficients of the stochastic influence operator $[\widetilde{\mathbf{H}}^{(d)}](\theta)$ are matrices with size $N_\Gamma^{(d)} \times N_\Gamma^{(d)}$ (see (4.41)), the total memory requirement MR to store the PC expansions of all the local influence operators is

$$\mathrm{MR}(D) = \sum_{d=1}^{D} \left(N_\Gamma^{(d)}\right)^2 J^{(d)}. \tag{4.49}$$

The evolution of $\mathrm{MR}(D)$ is reported in Figure 4.14a 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. 4.13a, which also levels off after $D \approx 500$, and by the joint decrease of $N_\Gamma^{(d)}$ on D. Even once the size of the PC bases has levelled off, the memory requirement does not increase but remains constant. The dependence of $N_\Gamma^{(d)}$ on D can be appreciated from Fig. 4.14b 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. 4.1); for more complex discretizations, *e.g.* adapted ones, it could be necessary to rely on more advanced partitioning procedures.

4.4.6 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 the 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



(a) Memory requirements $\mathrm{MR}(D)$.

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

Figure 4.14: Evolutions with the number of subdomains of the memory requirement (see (4.49)) for the FPC preconditioner (left) and local number of boundary nodes $N_\Gamma^{(d)}$ and total size of the Schur complement (right). The shaded area represents the RMS bounds around the average value.

FPCG method can be achieved. However, this rationale does not consider the inherent cost of applying the preconditioner during the sampling stage. Figure 4.14b 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 4). 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 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 $N_{KL}^{(d)}$) with the performance (related to $R_{\mathrm{KL}}(\tau)$).

Besides the complexity, one may also be concerned with the computational cost of the FPC preconditioner. The interested reader should refer to [32], where parallel experiments demon-

strated the scalability of the PC surrogates construction, thanks to trivial parallelism, as well as the low cost associated with the assembly of approximated $[\mathrm{S}]$ for each sample.

4.5 Conclusions

In this chapter, 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 (4.19) 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 [16, 117].

Chapter 5

A Surrogate-Based Balancing Domain Decomposition Method

Preconditioning with a Schur system's surrogates requires inverting at each iteration a problem with a size equal to the number of interface nodes, for each sample. Storing the decomposition (inverse) of this system may be a limitation for large-scale problems in higher dimensions. The current chapter presents an alternative preconditioning approach that is suitable for large-scale systems, by solving local (one per subdomain) inverses. The Balancing Neumann-Neumann method is adapted to the stochastic context by introducing PC-surrogates of the Neumann-Neumann maps. Numerical experiments show that the performance of the new surrogate-based method is close to the original approach.

5.1 Introduction

This chapter continues the development of efficient techniques to generate solution samples from elliptic equations with random coefficients. Following the same framework that has been used throughout this thesis, the focus will remain on coefficient fields associated with large variability and short correlation length.

5.1.1 The Schur Complement System

The discretization of the problem is performed with the same Schur-based methodology of the previous chapter. Each sampled problem is discretized according to a standard Finite Element (FE) method, and the resulting mesh is partitioned into non-overlapping subdomains. Recall the reduced Schur complement system for the interface values of the solution sample denoted by

$$[\mathrm{S}]\mathrm{u}_\Gamma = \mathrm{b}_\mathrm{S}. \tag{5.1}$$

The multiplication by $[\mathrm{S}]$ can be performed in a matrix-free fashion, exploiting the subdomain block structure of the Schur complement matrix:

$$[\mathrm{S}] = \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top [\mathrm{S}^{(d)}][\mathrm{R}^{(d)}], \tag{5.2}$$

where $[\mathrm{R}^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma}$ is a restriction matrix and $[\mathrm{S}^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma^{(d)}}$ is the influence matrix of $\Omega^{(d)}$ (see (4.24)). This matrix-free capability enables efficient iterative resolution of (5.1), for instance, using the Conjugate Gradient (CG) method. The application of each $[\mathrm{S}^{(d)}]$ is equivalent to solving a local problem. The smaller the subdomains are, the cheaper and faster the resolution of these local problems become. As a result, it is of best interest to reduce the size of the subdomains. Since the condition number of $[\mathrm{S}]$ increases with the number of subdomains, the convergence rate of the CG method deteriorates [113]. As a result, the CG method may require many iterations to converge, and calls for preconditioning techniques.

In the previous chapter, a Schur-based preconditioner was introduced to accelerate the CG scheme. The preconditioning step amounts to solve a system with size equivalent to (5.1). The

advantage of the method is that this system is known explicitly. However, preconditioning the residual requires the Cholesky decomposition of the system. The global character of this operation makes the preconditioner very effective, since the information depending on multiple subdomains is effectively treated. However, the cost associated with the Cholesky factorization grows extremely fast with the number of interface nodes [62]. In addition, leveraging the Cholesky factorization through a parallel construction is challenging, since typical parallel solvers are usually not specifically adapted to the DD setting. Nevertheless, even with parallel computation, storing the Cholesky factorization may still be an issue, and its application can involve costly communications, if the Schur system is large. As a result, the Cholesky factorization is not an option for large problems, and alternative approaches are in order. Bypassing the Cholesky factorization with an iterative method would have an equivalent cost of solving the original system (5.1). Therefore, a local preconditioning approach is needed.

This chapter proposes a preconditioner specifically designed to satisfy two major properties that resolve the limitations of the preconditioner proposed in the previous chapter. First, the application of the preconditioner will be based on independent local operations adapted to the DD setting. The locality of the preconditioner is also relevant for parallelization, as well as faster and cheaper resolution of system (5.1). Secondly, the number of PCG iterations produced using the preconditioner will be independent of the number of subdomains, i.e., the preconditioner will be scalable with the number of subdomains. This scalability is relevant to leverage the preconditioner's application using smaller subdomains without deteriorating the convergence.

5.1.2 Local Preconditioning

The Neumann-Neumann (NN) method [15] is a DD method that solves the Schur system by a PCG method. The key aspect of the NN method is that it is characterized by the *Neumann-Neumann preconditioner*, whose application is totally based on independent local operations amenable for parallelization.

Although the NN method has convenient locality properties, the number of PCG iterations it produces depends on the number of subdomains. The lack of scalability of the NN method is well known in the DD community, and several extensions have been proposed to achieve

scalability. This chapter uses a particular variant called the Balancing Domain Decomposition (BDD) method [81]. The BDD method has been successfully applied to a number of second-order elliptic problems [85, 19, 5] as well as in many other applications such as for CFD [33, 101], hydrogen dispersion [136], medical sciences [139], acoustic waves [138], elasticity [58], heat transfer [87], solid mechanics [6]. Similarly to the NN method, the BDD method amounts to solve the Schur system (5.1). However, instead of solving this system using only the NN preconditioner, the BDD method amounts to a hybrid approach: a combination of a direct and an iterative scheme. A summary of this hybrid approach follows.

The application of the NN preconditioner amounts to solve a set of local problems. Each local problem is associated with an influence matrix, mapping the subdomain portion of the original residual to the corresponding subdomain portion of the preconditioned residual. This map is often called the *Neumann-Neumann map*. The scalability issues associated with the NN preconditioner are related to the fact that some of these NN maps have non-trivial (non-zero) kernels. As a result, these NN maps are not invertible and the associated local problems have infinitely many solutions. Therefore, the preconditioner cannot correctly handle the components of the residual that belong to the subspace formed by these singularities, which are not preconditioned, and the convergence of the PCG scheme slows down.

The subspace spanned by the kernels of the NN maps associated with the elliptic problem (2.1) is often called the *Nicolaides* space [90]. The Nicolaides space is easy to construct, since the kernels of the NN maps are spanned by constant functions. The starting point of the BDD method is to identify a space containing the Nicolaides space. This space is often called the *coarse space*. The BDD method amounts to compute the part of the solution that belongs to the coarse space by a direct method, and the remaining solution components using an adaptation of the PCG iterative scheme. The usual practice is to remove, at each iteration, the residual components belonging to the coarse space using a projection-based strategy. The new iterative scheme is often called Projected PCG method [16] (PPCG).

Ideally, the coarse space is low dimensional. The smallest dimensional coarse space is the Nicolaides space itself, in which case it is referred to as the Nicolaides coarse space. In practice, the Nicolaides coarse space produces a scalable PPCG method for the type of elliptic problems (2.1) with continuous fields across the entire spatial domain. However, for other type

of problems, the Nicolaides coarse space may not be enough to achieve scalability, and larger coarse spaces are required. Moreover, although scalability may be achieved using the Nicolaides coarse space, larger coarse spaces can also be used to improve the rate of convergence of the PPCG method, as explored later in this chapter.

5.1.3 Motivation and Organization of the Chapter

The straightforward application of the BDD method to each sampled problem amounts to set up several sample-dependent operators/solvers. In solving many sampled problems, the process of setting all these operations can be costly, motivating the development of alternative cheaper procedures that can hopefully have the same performance as the original ones.

This chapter starts by presenting a new preconditioning approach for the BDD method. The idea is to bypass the NN maps' inversion associated with the NN preconditioner by surrogate-based operations. The surrogate-based preconditioner is combined with the Nicolaides coarse space to achieve a scalable surrogate-based BDD method. This chapter closes with a discussion on alternative approaches to the GenEO coarse space [89]. The GenEO coarse space is a coarse space that is usually used in the contexts where the Nicolaides coarse space cannot offer scalability. Contrary to the Nicolaides coarse space, the construction of the GenEO coarse space involves a number of sample-dependent operations with a significant computational burden. This chapter presents a cheaper GenEO-type coarse space construction that makes the PPCG scheme even more efficient to generate multiple samples.

The organization of the chapter is as follows. Section 5.2 describes the BDD method. Particular focus is put on the operations representing a significant computational load to the sampling stage and can hopefully be bypassed by cheaper ones with similar effect. Section 5.3 focuses on the preconditioning step of the BDD method and presents cheaper alternative strategies. These alternative strategies include a sample-independent strategy and a surrogate-based one. The preconditioners presented in this section are combined with the Nicolaides coarse space to obtain scalable variants of the BDD method. Section 5.4 presents several numerical experiments that compare the different preconditioning strategies previously discussed. Results show that the surrogate-based BDD method solves each sampled problem in a comparable number of

iterations as the usual BDD method and can therefore be an alternative in the context of solving multiple sampled problems. Results also show that the surrogate-based preconditioning strategy outperforms the sample-independent one. In Section 5.5, the Nicolaides coarse space is extended to the GenEO coarse space, sparkling a discussion on alternative sample-dependent coarse spaces adapted to the resolution of multiple sampled problems. Section 5.6 closes the chapter with some final remarks and future work.

5.2 The BDD Method

The starting point of this chapter is to generate solution samples by straightforward application of the BDD method. This section describes the BDD method and the process to obtain a solution sample.

5.2.1 The Neumann-Neumann Preconditioner

The strategy for accelerating convergence of the CG method followed in this chapter amounts to preconditioner the residual at each iteration according to the matrix $[\mathrm{P}_{\mathrm{inv}}] \approx [\mathrm{S}]^{-1}$. The preconditioned residual z is defined from the original residual r as[1]

$$\mathrm{z} := [\mathrm{P}_{\mathrm{inv}}]\mathrm{r}. \tag{5.3}$$

The multiplication by $[\mathrm{P}_{\mathrm{inv}}]$ should involve local operations. In addition, the process of applying the preconditioner should correctly map global residual vectors to local ones and vice-versa. The following paragraphs describe the approach used to achieve these two properties, starting with the latter.

The decomposition of global residual vectors into local contributions from each subdomain requires some care so that residual values are not counted twice. For example, consider a simple non-overlapping partition with $D = 2$ and a single interface $\Gamma = \Gamma^{(1)} = \Gamma^{(2)}$. A possible decomposition of some residual vector $\mathrm{r} \in \mathbb{R}^{N_\Gamma}$ is $\mathrm{r} = \mathrm{r}^{(1)} + \mathrm{r}^{(2)}$ with $\mathrm{r}^{(d)} = \frac{1}{2}\mathrm{r}$. In this case, the

[1] In this chapter, the preconditioner is the operator that maps the original residual to the preconditioned residual. Note that some authors consider the preconditioner to be the inverse of this matrix. Thus the $[\mathrm{P}_{\mathrm{inv}}]$ notation.

contribution of the nodes on each local interface $\Gamma^{(d)}$ was equally split. For a general subdomain partition, these types of decompositions are performed according to a *partition of the unity*. This partition of the unity is a diagonal matrix with non-negative entries $[\chi^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times N_\Gamma^{(d)}}$ such that

$$\sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top [\chi^{(d)}][\mathrm{R}^{(d)}] = [\mathrm{I}] \tag{5.4}$$

where $[\mathrm{I}] \in \mathbb{R}^{N_\Gamma \times N_\Gamma}$ is the identity matrix. Each diagonal entry of $[\chi^{(d)}]$ represents the contribution of an interface node to the local interface $\Gamma^{(d)}$. Then, the decomposition of the global vector becomes

$$\mathrm{r} = \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top \mathrm{r}^{(d)} \quad \text{with} \quad \mathrm{r}^{(d)} := [\chi^{(d)}][\mathrm{R}^{(d)}]\mathrm{r}. \tag{5.5}$$

The restriction matrix combined with a partition of unity is denoted as $[\check{\mathrm{R}}^{(d)}] := [\chi^{(d)}][\mathrm{R}^{(d)}]$.

Without loss of generality, and for the rest of the chapter, we assume that each node contributes *equally* to all subdomains it belongs to. Let $\mathcal{D}_i^{(d)}$ denote the set of subdomains that the interface node $x_i \in \Gamma^{(d)}$ belongs to. The entries of the partition of unity used throughout this thesis are defined such that

$$([\chi^{(d)}])_{i,i} := \frac{1}{|\mathcal{D}_i^{(d)}|} \tag{5.6}$$

Now that we have properly defined the operators that map global/local to local/global vectors, let us go back to the definition of the preconditioner $[\mathrm{P}_{\mathrm{inv}}]$. A natural candidate for $[\mathrm{P}_{\mathrm{inv}}]$ could be defined using $[\mathrm{S}^{(d)}]^{-1}$ in (5.2). However, for *floating subdomains* (subdomains that do not touch the physical boundary of Ω), the matrices $[\mathrm{S}^{(d)}]$ are singular [105]. The alternative is to use pseudo-inverses, denoted as $[\mathrm{S}^{(d)}]^\dagger$. Finally, the preconditioner is defined as

$$[\mathrm{P}_{\mathrm{inv}}] := \sum_{d=1}^{D} [\check{\mathrm{R}}^{(d)}]^\top [\mathrm{S}^{(d)}]^\dagger [\check{\mathrm{R}}^{(d)}], \tag{5.7}$$

using the pseudo-inverses and the partition of the unity defined above. The interest of the block structure (5.7) is that the preconditioned residual defined by (5.3) can be constructed through local operations involving $[\mathrm{S}^{(d)}]^\dagger$. The resolution of system (5.1) by a PCG method with preconditioner (5.7) is the Neumann-Neumann method, which is described in Algorithm 5. For this reason, preconditioner (5.7) will be called the *Neumann-Neumann (NN) preconditioner*.

Algorithm 5 The PCG method

1: **procedure** PCG-METHOD($[\mathrm{S}]$, $\mathrm{b_S}$, $[\mathrm{P_{inv}}]$, u_Γ^0, tol)
2: $\quad \mathrm{r}_0 = \mathrm{b_S} - [\mathrm{S}]\mathrm{u}_\Gamma^0$;
3: $\quad \mathrm{z}_0 = [\mathrm{P_{inv}}]\mathrm{r}_0$;
4: $\quad \mathrm{p}_0 = \mathrm{z}_0$;
5: $\quad r_2 = \|\mathrm{r}_0\|, k = 0$;
6: $\quad$ **while** $r_2 > \mathrm{tol}\|\mathrm{b_S}\|$ **do**
7: $\quad\quad \mathrm{q}_k = [\mathrm{S}]\mathrm{p}_k$;
8: $\quad\quad \alpha_k = \frac{\langle \mathrm{r}_k, \mathrm{z}_k \rangle}{\langle \mathrm{p}_k, \mathrm{q}_k \rangle}$;
9: $\quad\quad \mathrm{r}_{k+1} = \mathrm{r}_k - \alpha_k \mathrm{q}_k$;
10: $\quad\quad \mathrm{u}_\Gamma^{k+1} = \mathrm{u}_\Gamma^k + \alpha_k \mathrm{p}_k$;
11: $\quad\quad \mathrm{z}_{k+1} = [\mathrm{P_{inv}}]\mathrm{r}_{k+1}$;
12: $\quad\quad \beta_k = \frac{\langle \mathrm{r}_{k+1}, \mathrm{z}_{k+1} \rangle}{\langle \mathrm{r}_k, \mathrm{z}_k \rangle}$;
13: $\quad\quad \mathrm{p}_{k+1} = \mathrm{z}_{k+1} + \beta_k \mathrm{p}_k$;
14: $\quad\quad r_2 = \|\mathrm{r}_k\|, k = k + 1$;
15: $\quad$ **end while**
16: $\quad$ **return** u_Γ^k.
17: **end procedure**

The multiplication by the NN preconditioner (line 11) is performed through multiplication between the matrices $[\mathrm{S}^{(d)}]^\dagger$ and the corresponding subdomain portions of the residual. The cost of multiplying $[\mathrm{S}^{(d)}]^\dagger$ is directly related to the number of local interface nodes associated with each subdomain. The smaller these matrices are, the cheaper the local products become. Therefore, it is beneficial to reduce the size of the subdomains to reduce the cost of the preconditioning step, specially in the context of parallel implementations. However, as the number of subdomains increases, the convergence of the PCG scheme degrades, as explained in the following paragraph.

The components of the residual that belong to the Nicolaides coarse space are not preconditioned. The inappropriate treatment of these components becomes a source of slow convergence. As the number of subdomains grows, more floating subdomains appear, leading to an increasing number of singular matrices $[\mathrm{S}^{(d)}]$. As a result, the number of components of the residual that are not properly preconditioned increases and the convergence degrades. The lack of scalability of the NN method is a well-known problem in the community. The fix considered in this chapter is a projection-based strategy [16, 123, 89, 117] that amounts to apply a specific treatment to the solution components that belong to the Nicolaides coarse space.

5.2.2 The Nicolaides Coarse Space

The starting point of the projection-based strategy followed in this chapter is to define a coarse space. The particular coarse space considered here is the Nicolaides coarse space. Later in this chapter, the coarse space will be extended to a larger subspace to improve the convergence of the PPCG scheme.

Let D_{in} be the number of *floating* subdomains. Set $\mathcal{D}_{\mathrm{in}} := \{1, 2, \ldots, D_{\mathrm{in}}\}$ the set of indices of floating subdomains. Finally, set $s_f^{(d)} \in \mathcal{D}_{\mathrm{in}} \cup \{0\}$ the floating subdomain index of $\Omega^{(d)}$, equal to 0 if $\Omega^{(d)}$ is an outer subdomain ($\partial\Omega^{(d)} \cap \partial\Omega \neq \emptyset$).

Recall that the kernels of the NN maps associated to floating subdomains are spanned by local constant vectors [40], i.e., $[\mathrm{S}^{(d)}]\mathrm{c} = \underline{0}$, for any constant vector $\mathrm{c} \in \mathbb{R}^{N_\Gamma^{(d)}}$. As a result, the subspace formed by the non-zero kernels of each singular $[\mathrm{S}^{(d)}]$ are spanned by the vector $\underline{1} \in \mathbb{R}^{N_\Gamma^{(d)}}$, where all entries are one. The Nicolaides coarse space is the subspace spanned by the columns of the matrix $[\mathrm{W}] \in \mathbb{R}^{N_\Gamma \times D_{\mathrm{in}}}$ defined as

$$[\mathrm{W}]|_{s_f^{(d)}} := [\mathrm{R}^{(d)}]^\top \underline{1}, \quad s_f^{(d)} = 1, 2, \ldots, D_{\mathrm{in}}. \tag{5.8}$$

Note that the number of columns of $[\mathrm{W}]$ corresponds exactly to the number of floating subdomains. Indeed, according to definition (5.8), if $\Omega^{(d')}$ is an outer subdomain, it will not count as a column of $[\mathrm{W}^{(d)}]$ because $s_f^{(d')} = 0$.

Throughout this thesis, it will be useful to write the matrix $[\mathrm{W}]$ in terms of subdomain contributions. The block structure of $[\mathrm{W}]$ is given by

$$[\mathrm{W}] = \sum_{d=1}^{D} [\check{\mathrm{R}}^{(d)}]^\top [\mathrm{W}^{(d)}], \tag{5.9}$$

where $[\mathrm{W}^{(d)}] \in \mathbb{R}^{N_\Gamma^{(d)} \times D_{\mathrm{in}}}$ is a contribution of subdomain $\Omega^{(d)}$ to the matrix $[\mathrm{W}]$, defined as

$$([\mathrm{W}^{(d)}])_{i, s_f^{(d')}} := \begin{cases} 1, \text{ if } \quad x_i \in \Gamma^{(d)} \cap \Gamma^{(d')}, \\ 0, \text{ otherwise.} \end{cases}, \text{ for } s_f^{(d')} = 1, \ldots, D_{\mathrm{in}}, \tag{5.10}$$

Note that each matrix $[\mathrm{W}^{(d)}]$ is sparse, and the only non-zero columns correspond to the floating

subdomains $\Omega^{(d')}$ that are neighbours of $\Omega^{(d)}$. A neighbour of $\Omega^{(d)}$ is any subdomain $\Omega^{(d')}$ such that $\partial\Omega^{(d)} \cap \partial\Omega^{(d')} \neq \emptyset$, which in particular means that $\Omega^{(d)}$ is a neighbour of itself. We will denote $N_D^{(d)}$ the number of neighbours of $\Omega^{(d)}$.

5.2.3 The Projected PCG Method

Once the coarse space is defined, the idea is to treat the part of the solution within the Nicolaides coarse space separately from the iterative scheme. To this end, define the $[\mathrm{S}]$-orthogonal projection from $\mathbb{R}^{N_\Gamma}$ onto the Nicolaides coarse space by the matrix $[\Pi]$ defined as

$$[\Pi] := [\mathrm{W}][\mathrm{M}]^{-1}[\mathrm{W}]^\top[\mathrm{S}], \quad \text{with} \quad [\mathrm{M}] := [\mathrm{W}]^\top[\mathrm{S}][\mathrm{W}]. \tag{5.11}$$

The *projected* residual, denoted by $\bar{\mathrm{z}}$, is obtained by projecting the preconditioned residual defined in (5.3) to the subspace $[\mathrm{S}]$-orthogonal to the coarse space:

$$\bar{\mathrm{z}} := ([\mathrm{I}] - [\Pi])\mathrm{z}. \tag{5.12}$$

The matrix $[\Pi]$ is never explicitly formed. Instead, the application of $[\Pi]$ involves mostly local operations, and it assumes that matrix $[\mathrm{M}]$ is constructed and factorized prior to the iterative scheme.

The most expensive operation in the construction of $[\mathrm{M}]$ is the product $[\mathrm{S}][\mathrm{W}]$. Exploiting the sparsity and local structure of both $[\mathrm{S}]$ and $[\mathrm{W}]$, this product is decomposed into local products in the following form

$$[\mathrm{S}][\mathrm{W}] = \sum_{d=1}^{D} [\mathrm{R}^{(d)}]^\top [\mathrm{S}^{(d)}][\mathrm{W}^{(d)}]. \tag{5.13}$$

Each product $[\mathrm{S}^{(d)}][\mathrm{W}^{(d)}]$ amounts to solve $N_D^{(d)}$ non-trivial local problems per subdomain, corresponding to the non-zero columns of $[\mathrm{W}^{(d)}]$ (floating neighbours of $\Omega^{(d)}$). The construction of $[\mathrm{M}]$ is completed by applying $[\mathrm{W}]^\top$ to the product (5.13), which can also be performed through independent local products exploiting the block structure of $[\mathrm{W}]$. Hence, the total cost of assembling $[\mathrm{M}]$ is asymptotically equivalent to the resolution of $\sum_d^D N_D^{(d)}$ local problems. Note that these local problems can be computed independently from the others and in parallel.

It now comes down to find $[\mathrm{M}]^{-1}$. The matrix $[\mathrm{M}]$ is symmetric and SPD. Therefore, $[\mathrm{M}]^{-1}$ will represent a Cholesky factorization of $[\mathrm{M}]$. The cost of this Cholesky decomposition is $\mathcal{O}(D_{\mathrm{in}}^3)$ [62]. Thanks to the low dimensionality of $[\mathrm{M}]$, storing its Cholesky factorization requires low memory.

At each iteration, the computation of the projected residual (5.12) involves the multiplication of the original residual r by $[\mathrm{P}_{\mathrm{inv}}]$, as in (5.3), followed by the application of the matrix $[\Pi]$, as in (5.12). In this process, the most expensive operation is the application of $[\mathrm{S}]$, which involves the resolution of a set of D local problems. The additional operations involved in the computation of the projected residual are multiplications by the matrices $[\mathrm{W}]$ and $[\mathrm{W}]^\top$. Thanks to the block structure (5.9), these multiplications are performed through independent local operations associated to each $[\mathrm{W}^{(d)}]$. In this thesis, the subdomains are sufficiently small so that the (sparse) matrices $[\mathrm{W}^{(d)}]$ are explicitly formed and stored. As a result, the application of $[\mathrm{W}]$ involves cheap local matrix-vector products $[\mathrm{W}^{(d)}]$.

The projected residual obtained at each iteration has no components in the Nicolaides coarse space. As a result, the PCG method will not converge unless the iterations start at an initial guess within the coarse space. To this end, the initial iterate is set to be the part of the solution on the coarse space:

$$\mathrm{u}_\Gamma^0 := [\Pi]\mathrm{u}_\Gamma = [\mathrm{W}][\mathrm{M}]^{-1}[\mathrm{W}]^\top \mathrm{b}_{\mathrm{S}}. \tag{5.14}$$

Contrary to the computation of the projected residuals, the computation of the initial iterate is straightforward, as it does not involve the resolution of any local problem. Note that, if all subdomains touch the physical boundary of Ω, then the Nicolaides subspace is empty and therefore, no projection is needed.

The PPCG method is described in Algorithm 6. The part of the residual within the Nicolaides coarse space is computed according to the initial condition at line 2. At each iteration, the preconditioned residual is projected in the $[\mathrm{S}]$-orthogonal complement of the coarse space at step 14. Therefore, the resulting conjugate directions constructed at step 15 are restricted to the space $[\mathrm{S}]$-orthogonal to the Nicolaides coarse space. The process of preconditioning the residual and then projecting it back to the subspace $[\mathrm{S}]$-orthogonal to the Nicolaides coarse

Algorithm 6 The PPCG method

1: **procedure** PPCG-METHOD($[\mathrm{S}]$, $\mathrm{b_S}$, $[\mathrm{P_{inv}}]$, u_Γ^0, $[\Pi]$, tol)
2: $\quad \mathrm{r}_0 = \mathrm{b_S} - [\mathrm{S}]\mathrm{u}_\Gamma^0$;
3: $\quad \mathrm{z}_0 = [\mathrm{P_{inv}}]\mathrm{r}_0$;
4: $\quad \bar{\mathrm{z}}_0 = ([\mathrm{I}] - [\Pi])\mathrm{z}_0$;
5: $\quad \bar{\mathrm{p}}_0 = \bar{\mathrm{z}}_0$;
6: $\quad r_2 = \|\mathrm{r}_0\|, k = 0$;
7: $\quad$ **while** $r_2 > \mathrm{tol}\|\mathrm{b_S}\|$ **do**
8: $\quad\quad \mathrm{q}_k = [\mathrm{S}]\bar{\mathrm{p}}_k$;
9: $\quad\quad \alpha_k = \frac{\langle \mathrm{r}_k, \mathrm{z}_k \rangle}{\langle \bar{\mathrm{p}}_k, \mathrm{q}_k \rangle}$;
10: $\quad\quad \mathrm{r}_{k+1} = \mathrm{r}_k - \alpha_k \mathrm{q}_k$;
11: $\quad\quad \mathrm{u}_\Gamma^{k+1} = \mathrm{u}_\Gamma^k + \alpha_k \bar{\mathrm{p}}_k$;
12: $\quad\quad \mathrm{z}_{k+1} = [\mathrm{P_{inv}}]\mathrm{r}_{k+1}$;
13: $\quad\quad \beta_k = \frac{\langle \mathrm{r}_{k+1}, \mathrm{z}_{k+1} \rangle}{\langle \mathrm{r}_k, \mathrm{z}_k \rangle}$;
14: $\quad\quad \bar{\mathrm{z}}_{k+1} = ([\mathrm{I}] - [\Pi])\mathrm{z}_{k+1}$;
15: $\quad\quad \bar{\mathrm{p}}_{k+1} = \bar{\mathrm{z}}_{k+1} + \beta_k \bar{\mathrm{p}}_k$;
16: $\quad\quad r_2 = \|\mathrm{r}_k\|, k = k + 1$;
17: $\quad$ **end while**
18: $\quad$ **return** u_Γ^k.
19: **end procedure**

space is performed for each iteration. The iterations stops when the criteria $\|\mathrm{r}_k\|/\|\mathrm{b_S}\| < \mathrm{tol}$ is met (step 7).

The PPCG method is mostly based on local operations. Each iteration is recast through steps 7 - 17. Step 8 and step 14 involve the multiplication by $[\mathrm{S}]$, which amounts to solve D local problems. The resolution of these local problems is the most expensive operation of each iteration. In addition to the multiplication by $[\mathrm{S}]$, step 14 also involves the application of the matrices $[\mathrm{W}]$ and $[\mathrm{W}]^\top$, which amount to local matrix-vector multiplications exploiting the block structure (5.9). Finally, the preconditioning step 3 amounts to multiply the set of pseudo-inverse matrices. Eventually, the only global operation is the application of $[\mathrm{M}]^{-1}$, which, as discussed above, is cheap thanks to the low dimension of $[\mathrm{M}]$. If D is not too large, the Cholesky factorization of $[\mathrm{M}]$ can be copied and stored on each processing unit (subdomain), which decreases the communication latency at each iteration and facilitates the parallel execution.

5.3 The BDD method for Sampling

The BDD method is designed to solve a single deterministic problem. Naturally, this method can be used to generate individual solution samples. However, the BDD method amounts to several sample-dependent operations with a significant cost that may limit the number of samples produced. This section focuses on the set of operations involved in the preconditioning step and introduces alternative approaches suitable for sampling.

5.3.1 The Elliptic Equation Equation with Random Coefficients

Let us start with a summary of the stochastic problem described in the previous chapters. For some random event θ, the stochastic elliptic equation we are interested in has the form

$$\begin{aligned} &\nabla \cdot [\boldsymbol{\kappa}(x,\theta)\nabla \boldsymbol{u}(x,\theta)] = -f(x) \quad x \in \Omega,\, \theta \in \Theta \\ &\boldsymbol{u}(x,\theta) = 0, \quad x \in \partial\Omega,\, \theta \in \Theta, \end{aligned} \tag{5.15}$$

where $f(x)$ is a deterministic source term. The equalities in the equations of (5.15) stand in the $\mathcal{P}$-almost surely sense and for almost every x. In this work, we restrict ourselves to the case of $\boldsymbol{\kappa}$ being a stationary log-normal stochastic process, whose $\log$ is a Gaussian process $\boldsymbol{g}$ with mean $\mu_{\boldsymbol{g}}$ and covariance function C:

$$\boldsymbol{g}(x,\theta) := \log \boldsymbol{\kappa}(x,\theta) \sim \mathcal{N}(\mu_{\boldsymbol{g}}, C). \tag{5.16}$$

Without loss of generality, the covariance $C : (x, x') \in \Omega \times \Omega \mapsto \mathbb{R}$ is defined as

$$C(x,x') := \sigma_{\boldsymbol{g}}^2 \exp\left(-\frac{\|x-x'\|_\Omega^\gamma}{\gamma \ell_c}\right), \tag{5.17}$$

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

We are interested in approximating $\mathbb{E}[\boldsymbol{z}(\boldsymbol{u})]$ for some statistics $\boldsymbol{z}$ of the solution of (5.15) using a sampling method. This estimation amounts to generate solution samples $u^{(m)}$. Each solution sample is the solution of a deterministic elliptic problem for the sampled coefficient value $\kappa^{(m)}$. A large set of solution samples must be computed to ensure that the sampling error

is small enough and have an accurate approximation of $\mathbb{E}[\boldsymbol{z}(\boldsymbol{u})]$. The size of the sample set entails a significant computational effort.

5.3.2 Major Drawbacks of the Direct BDD Method for Sampling

The straightforward application of the BDD method to each sample problem involves performing several operations that, depending on the problem size, may represent a significant computational load and limit the number of solution samples. This chapter focuses on two particular operations: a) the set up of the preconditioner and b) the set up of the operators involved in the projection of the residuals. In order to make the BDD method more attractive for sampling, these (expensive) procedures are replaced with cheaper alternative ones. These alternative approaches are the topic of the rest of this chapter. The discussion starts with alternative preconditioning strategies, which is covered in the current section and in the following one. Alternative projection strategies will be discussed later in the chapter.

Throughout this thesis, we assume that the subdomains are sufficiently small so that the pseudo-inverses $[\mathrm{S}^{(d)}]^{\dagger}$ can be explicitly formed and stored. Without loss of generality, the approach used to compute each matrix $[\mathrm{S}^{(d)}]^{\dagger}$ amounts to perform an eigenvalue decomposition [62] of $[\mathrm{S}^{(d)}]$, which is constructed explicitly. The construction of the set of influence matrices $[\mathrm{S}^{(d)}]$ involves the resolution of the same local problems as in the application of $[\mathrm{S}]$ during the original CG scheme. Therefore, the additional cost of setting the NN preconditioner with respect to the original CG method is dominated by the eigenvalue decomposition of each $[\mathrm{S}^{(d)}]$. Note that, since the BDD method is scalable with the number of subdomains [5], the size of the subdomains can be made smaller without deteriorating convergence. Therefore, in case the subdomains are excessively large, increasing the number of subdomains can be a way to proceed by the direct approach just described.

In practice, the generation of each solution sample amounts to solve a Schur system (5.1) associated with some $\kappa^{(m)}$. The resolution of the Schur system can be performed using the NN preconditioner in Algorithm 6. However, the construction of the matrices $[\mathrm{S}^{(d)}]^{\dagger}$ for each sample represents significant computational load to the overall sampling stage, which may impact the number of sample problems that can be solved. Alternative preconditioning strategies suitable

for resolving multiple sampled problems include approaches such as sampling-independent preconditioning and surrogate-based preconditioning that have been extensively used throughout this thesis. This section presents variants of these alternative preconditioning approaches that bypass the construction of the pseudo-inverse matrices associated with the NN preconditioner and are explicitly designed for solving many sampled problems. Ideally, setting the approximated NN-based preconditioners is significantly cheaper than the original NN preconditioner, and their scalability properties are comparable. Under these assumptions, the preconditioners can represent an alternative to the original NN preconditioner in the context of multiple sampled problems.

5.3.3 The Median-NN Preconditioner

An alternative to avoid setting a new NN preconditioner for each new sample is to use the same preconditioner for all samples. This sample-independent approach amounts to set the local solvers associated with the preconditioning before the sampling stage. Once constructed, these solvers are repeatedly applied for each sample, meaning the same pseudo-inverses. The question comes down to select the preconditioner that adequately represents the different NN preconditioners associated with every coefficient sample.

The sample-independent preconditioner used in this chapter is associated with pseudo-inverse matrices defined according to the median of the coefficient field. Consider the set of pseudo-inverses of the median-based influence matrices, i.e., influence matrices of the Schur complement matrix corresponding to the median of κ. The median-based pseudo-inverses are computed before the sampling stage and then re-applied for each sample. This specific preconditioning approach will be called the *median-based Neumann-Neumann (M-NN) preconditioner*.

5.3.4 The PC-NN Preconditioner

In light of the deterministic Schur-based preconditioners presented in the previous chapter, it is expected (and later confirmed) that the M-NN preconditioner's performance degrades as the field's variance increases. Therefore, the NN preconditioner demands an alternative preconditioning approach that is better to each coefficient sample. To this end, the following surrogate-

based approach is introduced.

5.3.4.1 Factorization of the pseudo-inverse matrices

The idea of the surrogate-based preconditioning approach presented in this chapter is to bypass the expensive construction of $[\mathrm{S}^{(d)}]^\dagger$. To this end, a surrogate of each pseudo-inverse matrix is constructed at a pre-processing stage and subsequently evaluated for each sample.

The starting point of the surrogate construction is to introduce the finite representation of the field over $\Omega^{(d)}$. Let $\boldsymbol{\kappa}^{(d)}(x,\theta)$ denote the restriction of $\boldsymbol{\kappa}$ to $x \in \Omega^{(d)}$. The local truncated KL expansion of the Gaussian process $\boldsymbol{g}^{(d)} = \log \boldsymbol{\kappa}^{(d)}$ over $\Omega^{(d)}$ is given by

$$\boldsymbol{g}^{(d)}(x,\theta) \approx \hat{g}^{(d)}(x,\theta) := \mu_{\boldsymbol{g}} + \sum_{i=1}^{N_{KL}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta), \quad x \in \Omega^{(d)}, \tag{5.18}$$

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

$$\boldsymbol{\kappa}^{(d)}(x,\theta) \approx \hat{\boldsymbol{\kappa}}^{(d)}(x,\theta) := \exp\left[\mu_{\boldsymbol{g}} + \sum_{i=1}^{N_{KL}^{(d)}} \sqrt{\lambda_i^{(d)}}\hat{\phi}_i^{(d)}(x)\boldsymbol{\xi}_i^{(d)}(\theta)\right]. \tag{5.19}$$

In the following, we call $\|\boldsymbol{\kappa}^{(d)} - \hat{\boldsymbol{\kappa}}^{(d)}\|_{\Omega^{(d)}}$ the KL truncation error (see (4.1) for the definition of the norm $\|\cdot\|_{\Omega^{(d)}}$). Denote by $[\widehat{\mathbf{S}}^{(d)}](\theta)$ the stochastic influence matrix of the subdomain based on $\hat{\boldsymbol{\kappa}}^{(d)}$. The randomness of $[\widehat{\mathbf{S}}^{(d)}](\theta)$ is inherited by the randomness of $\hat{\boldsymbol{\kappa}}^{(d)}$, such that $[\widehat{\mathbf{S}}^{(d)}]$ is a function of $\boldsymbol{\xi}^{(d)}(\theta)$. If $N_{KL} = 0$, the resulting $\hat{\boldsymbol{\kappa}}^{(d)}$ corresponds to the median of the coefficient field, and pseudo-influence matrices obtained are exactly the median-based ones used to define the M-NN preconditioner.

One way to construct a surrogate for each pseudo-inverse matrix is to directly construct a PC-expansion of it. However, the resulting PC-based matrices $[\widetilde{\mathrm{S}}^{(d)}]^\dagger$ retrieved from each sample could potentially have negative eigenvalues for some samples. As a result, the surrogate-based preconditioner defined according to these matrices would not be symmetric and positive-definite (SPD). Non-SPD preconditioners can significantly degrade the convergence of the CG

scheme, as extensively illustrated in the previous chapter. Therefore, the PC-NN preconditioner must be based on a.s. non-negative $[\widetilde{\mathrm{S}}^{(d)}]^{\dagger}$ so that $[\mathrm{P}_{\mathrm{inv}}]$ is SPD and the convergence is also guaranteed. To this end, the following surrogate's construction follows the same strategy used in Section 4.3.4.

Let $[\widehat{\mathrm{S}}^{(d)}]$ denote an influence matrix of some sample of $\hat{\kappa}$. Recall the orthogonal decomposition of the influence matrix introduced in the previous chapter in (4.38):

$$[\widehat{\mathrm{S}}^{(d)}] = [\mathrm{Q}][\mathrm{D}][\mathrm{Q}]^{\top}. \tag{5.20}$$

From this decomposition, define the factor

$$[\mathrm{H}^{(d)}] := [\mathrm{Q}][\Delta]^{\dagger}[\mathrm{Q}]^{\top}, \tag{5.21}$$

where $[\Delta]^{\dagger}$ is a diagonal matrix whose entries are

$$([\Delta]^{\dagger})_{i,i} := \begin{cases} 0 \text{ if } ([\mathrm{D}])_{i,i} = 0, \\ \frac{1}{\sqrt{([\mathrm{D}])_{i,i}}} \text{ if } ([\mathrm{D}])_{i,i} > 0. \end{cases} \tag{5.22}$$

In practice, definition (5.22) is used with condition $([\mathrm{D}])_{i,i} < \epsilon$, for some $\epsilon \ll 1$.

The idea now is to build a PC expansion for $[\mathrm{H}^{(d)}]$. We refer to Section 4.3.3 for details on the Hermite polynomials, and set $\{\boldsymbol{\Psi}_{\alpha}, \alpha \in \mathcal{B}\}$ the finite dimensional PC basis. The PC expansion is written as

$$[\widetilde{\mathbf{H}}^{(d)}](\boldsymbol{\xi}^{(d)}) := \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_{\alpha} \boldsymbol{\Psi}_{\alpha}^{(d)}(\boldsymbol{\xi}^{(d)}), \tag{5.23}$$

where $\mathcal{B}^{(d)}$ is a multi-index. The coefficients $[\mathrm{H}^{(d)}]_{\alpha}$ of PC expansion (5.23) are given by a Sparse Grid Pseudo Spectral Projection (PSP) [133, 29, 30]. The PSP method uses Smolyak quadrature formulas adapted to each index α. For the PSP formula with the sparse grid level l it writes

$$[\mathrm{H}^{(d)}]_{\alpha} \approx \sum_{q=1}^{N_l^{(d)}} [\mathrm{H}^{(d)}] \left(\mathrm{y}_{q,l}^{(d)}\right) \boldsymbol{\Psi}_{\alpha}^{(d)} \left(\mathrm{y}_{q,l}^{(d)}\right) w_{q,\alpha,l}^{(d)}. \tag{5.24}$$

The projection with level l requires the evaluation of $N_l^{(d)}$ $\hat{\boldsymbol{\kappa}}$-based influence matrices, and $N_l^{(d)}$

increases quickly with l and $N_{KL}^{(d)}$. Also, the level fixes the PC basis (multi-index set $\mathcal{B}^{(d)}$), which increases with l. For simplicity, all subdomains have the same level and number of random variables $N_{KL}^{(d)}$, so that $N_l^{(d)}$ and $\mathcal{B}^{(d)}$ are all the same. Each evaluation $[\mathrm{H}^{(d)}]\left(\mathrm{y}_{q,l}^{(d)}\right)$ is defined by (5.21) for the orthogonal decomposition (5.20) based on $\hat{\kappa}^{(d)}\left(\mathrm{y}_{q,l}^{(d)}\right)$. Algorithm 7 summarizes the procedure to compute the coefficients for the PC expansion (5.23). This algorithm follows the same structure of Algorithm 3. In particular, this algorithm uses exclusively local information, and requires no communication between subdomains. Therefore, the set of PC expansions can be constructed in parallel.

Algorithm 7 Set PC expansion $[\widetilde{\mathbf{H}}^{(d)}]$

procedure COMPUTE-$[\widetilde{\mathbf{H}}^{(d)}]$(KL expansions $\hat{\boldsymbol{\kappa}}^{(d)}$, PC basis)
  Set quadrature nodes and weights;
  for all PC modes α **do**
    set $[\mathrm{H}^{(d)}]_\alpha = [0]$; ▷ Initialization of the PC modes
  end for
  for $q = 1, \ldots, N_l^{(d)}$ **do** ▷ Loop over quadrature nodes
    Evaluate $\hat{\kappa}^{(d)}$ for $\mathrm{y}_{q,l}^{(d)}$;
    Compute $[\mathrm{S}^{(d)}]$ for $\hat{\kappa}^{(d)}(\mathrm{y}_{q,l}^{(d)})$; ▷ Set the influence matrix
    Solve $[\mathrm{S}^{(d)}] = [\mathrm{Q}][\mathrm{D}][\mathrm{Q}]^\top$; ▷ Decompose the influence matrix
    Set $[\mathrm{H}^{(d)}] = [\mathrm{Q}][\Delta]^\dagger[\mathrm{Q}]^\top$; ▷ Set the factor, see (5.21)
    for all PC mode α **do**
      $[\mathrm{H}^{(d)}]_\alpha \leftarrow [\mathrm{H}^{(d)}]_\alpha + [\mathrm{H}^{(d)}]\Psi_\alpha^{N_{KL}^{(d)}}(\mathrm{y}_{q,l}^{(d)})w_{q,l}^{(d)}$; ▷ Update PC modes, see (5.23)
    end for
  end for
  return $\{[\mathrm{H}^{(d)}]_\alpha\}$; ▷ Return the PC modes
end procedure

The pre-processing stage of the proposed surrogate-based method amounts to construct local truncated PC expansions for each subdomain according to Algorithm 7. For sufficiently small subdomains, the local truncated KL expansions are very accurate using only a few of random variables. As a result, the number of PC terms ($|\mathcal{B}^{(d)}|$) is small, which, in particular, implies minimal storing requirements. Moreover, the small number of local random variables also induces a small number of sparse grid points $N_l^{(d)}$, making the estimation of the coefficients cheap. In any case, the cost of the setting these PC expansions is factorized over the number of solution samples. As a result, for a sufficiently large number of samples, the impact of the pre-processing stage onto the overall cost of the method is negligible, which is dominated by

the sampling stage.

5.3.4.2 Retrieving samples of local preconditioners

The stochastic surrogate-based pseudo-inverse matrices, denoted by $[\widetilde{\mathbf{S}}^{(d)}]^{\dagger}$ are defined taking the square of the PC expansions (5.23):

$$[\widetilde{\mathbf{S}}^{(d)}]^{\dagger}(\boldsymbol{\xi}^{(d)}) := \left([\widetilde{\mathbf{H}}^{(d)}](\boldsymbol{\xi}^{(d)})\right)^2. \tag{5.25}$$

Clearly, $[\widetilde{\mathbf{S}}^{(d)}]^{\dagger}$ is non-negative a.s. The stochastic *PC-based Neumann Neumann* (PC-NN) preconditioner is defined by:

$$[\mathbf{P}_{\mathrm{inv}}](\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(D)}) := \sum_{d=1}^{D} [\check{\mathrm{R}}^{(d)}]^{\top} [\widetilde{\mathbf{S}}^{(d)}]^{\dagger}(\boldsymbol{\xi}^{(d)}) [\check{\mathrm{R}}^{(d)}]. \tag{5.26}$$

The surrogate-based BDD method is the BDD method where the PC-NN preconditioner substitutes the original NN preconditioner. For each sample $\kappa^{(m)}$, one can associated a sample of the stochastic PC-NN preconditioned. However this sample is not actually formed. Instead, the local surrogate-based pseudo-inverses are computed according to

$$[\widetilde{\mathrm{S}}^{(d)}]^{\dagger}(\eta^{(d)}) := \left([\widetilde{\mathbf{H}}^{(d)}](\eta^{(d)})\right)^2, \tag{5.27}$$

where the $\eta^{(d)}$ are defined from $\kappa^{(m)}$ through

$$\eta_i^{(d)} := \frac{1}{\sqrt{\lambda_i^{(d)}}} \int_{\Omega^{(d)}} \left(\log(\kappa^{(m)}) - \mu_{\boldsymbol{g}}\right) \hat{\phi}_i^{(d)}, \quad \forall i = 1, \ldots, N_{KL}^{(d)} \text{ and } d = 1, \ldots, D. \tag{5.28}$$

Algorithm 8 describes the construction of the PC-based pseudo-inverses associated with each sample. The expensive computation of the pseudo-inverses that characterizes the set-up of the NN preconditioner is bypassed in Algorithm 8. If $N_{KL}^{(d)}$ is not too large, the number of polynomial evaluations is small, making the evaluations of the PC-expansions (step 4) very efficient. The resulting PC-based pseudo-inverse matrices are generated according to (5.27), and allocated to each subdomain independently of the remaining subdomains, in parallel.

Algorithm 8 Procedure to get a sample of the set of PC-based pseudo-inverses

procedure PC-NN(Sample $\kappa^{(m)}$, $\{[\mathrm{H}^{(d)}]_\alpha\}$, PC basis)
  for $d = 1, \ldots, D$ **do** ▷ Loop over subdomains
    Set $\eta^{(d)}$ by local projection; ▷ see (5.28)
    Set $[\widetilde{\mathbf{H}}^{(d)}](\eta^{(d)}) = \sum_{\alpha \in \mathcal{B}^{(d)}} [\mathrm{H}^{(d)}]_\alpha \mathbf{\Psi}_\alpha^{N_{KL}^{(d)}}\left(\eta^{(d)}\right)$; ▷ Realization of factor
    Set $[\widetilde{\mathrm{S}}^{(d)}]^\dagger$ according to (5.27); ▷ Surrogate-based pseudo-inverse
  end for
  Return $\{[\widetilde{\mathrm{S}}^{(d)}]^\dagger\}$; ▷ Return PC-based Pseudo-Inverses
end procedure

5.4 Numerical Comparison of NN Preconditioning Strategies

In general, the set-up cost associated with either the M-NN or the PC-NN preconditioners is significantly lower than the cost associated with the NN preconditioner. However, it expected that the NN preconditioner yields the smallest number of iterations. Indeed, the corrections of the residuals made by the NN preconditioner are based on exact pseudo-inverse matrices of each sample, while the two other alternatives use an approximation of the exact pseudo-inverses, leading to less effective corrections. As a result, the number of iterations needed to achieve convergence should be larger than using the original NN preconditioner. The section discusses how the performance of the M-NN and PC-NN preconditioners differs from the original NN approach.

The methodology for assessing the performance of each preconditioner is based on the average number of iterations that the PPCG method requires to reach the stopping criterium $\|\mathrm{r}_k\|/\|\mathrm{b}_\mathrm{S}\| < \mathrm{tol} = 1e^{-6}$ (see step 7, Algorithm 6). The reference is the number of iterations obtained using the NN preconditioner. The M-NN or PC-NN preconditioners are said to perform well if the associated number of iterations is close to the reference. The following performance index is used to facilitate the comparison between the different preconditioners:

$$\rho := \mathbb{E}\left[\frac{\#_{\text{NN}}}{\#_{\text{M-NN}} \text{ or } \#_{\text{PC-NN}}}\right], \tag{5.29}$$

were $\#_i$ for $i =$NN, PC-NN or M-NN is the number of iterations until convergence using a fixed preconditioner. Average values are obtained over $1\,000$ samples. In practice, the M-NN or PC-NN preconditioners are said to perform well if ρ is close to the unit. Throughout this section,

all preconditioners use the Nicolaides coarse space. The analysis is restricted to coefficient fields associated to centred Gaussian fields, i.e., $\mu_{\boldsymbol{g}} = 0$, and a squared exponential covariance matrix, i.e., $\gamma = 2$ in definition (5.17). The FE basis used throughout this section has P_2 triangular elements. The FE mesh uses elements of comparable size, and it is partitioned into subdomains with comparable number of elements.

5.4.1 Effect of the Field's Complexities on the Performance of the Preconditioners

This subsection investigates the performance of the M-NN and PC-NN preconditioners in the context of coefficient fields with different variances and correlation lengths. The set-up of the test case is as follows: The mesh has $N_{\text{el}} = 32\,685$ elements with $N_{\text{nod}} = 65\,858$ nodes. This mesh is partitioned into $D = 40$ subdomains, which yields a total $N_{\Gamma} = 2\,940$ nodes over the subdomains' interface.

5.4.1.1 Performance of M-NN preconditioner

Let us start by analysing the performance of the M-NN preconditioner, reported in Figure 5.1. In general, the overall performance of the M-NN preconditioner deteriorates significantly as the variance increases, and as the correlation length decreases. This behaviour of the curves reported is explained in the following.

For low variance, the sampled $\kappa^{(m)}$ remains close to the field's median. Therefore, the resulting sample-dependent pseudo-inverses are similar to the median-based ones, and the convergence rate of the M-NN preconditioner is also close to the reference. In addition, fields with higher correlation have smaller fluctuations across the spatial domain. Since the median field is constant over the domain, the M-NN preconditioner achieves higher performances for higher correlated fields. For example, the performance index corresponding to $\sigma^2_{\boldsymbol{g}} = 0.1$ and $\ell_c = 1$ is $\rho \approx 0.9$. In general, the M-NN preconditioner is a good alternative to the NN preconditioner for low variable or highly correlated coefficient fields.

As the variance increases, the set of sampled fields becomes more and more distinct from the field's median (emergence of more extreme events). Therefore, the resulting median-based

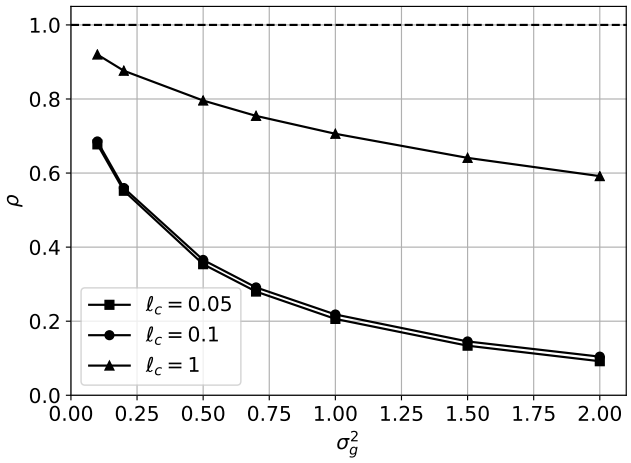


Figure 5.1: Comparison of the performance index of the M-NN preconditioner in the context of fields with different variances and correlation lengths.

performance can be very different depending on the correlation length. In the case of high correlation, the samples are almost constant in Ω, such that the sample $\kappa^{(m)}$ is nearly proportional to the field's median over the entire spatial domain. The median-based and the sample-dependent pseudo-inverses are also proportional. Hence, the residuals preconditioned with the M-NN preconditioner are scaled versions of the residuals preconditioned with the NN approach, producing similar convergence rates.

As the correlation decreases, the ratio $\kappa^{(m)}/\bar{\kappa}$ changes. This ratio might be quite constant locally over $\Omega^{(d)}$, but can differ from a subdomain to another. As a result, the global residuals obtained by applying the median-based pseudo-inverses are *not* just scaled versions from those produced with the application of the sample-dependent pseudo-inverses, resulting in poorer performances. In fact, the performance of the M-NN preconditioner degrades with the number of subdomains, as explained later in this chapter. In addition, the quality of the median field as a field's representation can only deteriorate as the variance increases, lowering, even more, the preconditioner's performance. The effect of high variance is more evident for low correlation values because the short-scale fluctuations are augmented when the variance is larger.

5.4.1.2 Performance of the PC-NN preconditioner

The performance of the PC-NN preconditioner is reported in Figure 5.2. The PC-NN preconditioner uses the same stochastic discretization parameters for all subdomains, namely, $N_{KL}^{(d)} = 4$ and $l = 3$. This produces a PC basis with 63 coefficients and 209 sparse grid points. These discretization parameters were set to fit with the available computational resources, and so that the PC-NN preconditioner performs decently with moderated cost. Note that the choice of using the same number of stochastic discretization parameters for all subdomains is adequate since the subdomains have comparable size. The effect of the stochastic discretization parameters in the performance of the PC-NN preconditioner is discussed later in this section (Section 5.4.2).

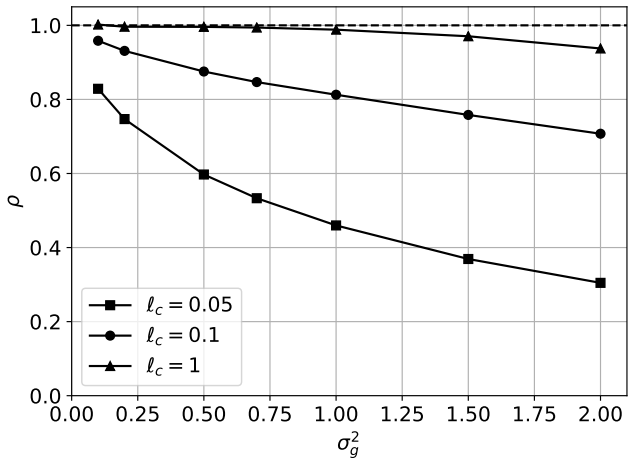


Figure 5.2: Comparison of the performance index of the PC-NN preconditioner in the context of fields with different variances and correlation lengths.

The performance of the PC-NN preconditioner depends on how well the PC-based pseudo-inverses can approximate the exact pseudo-inverses. In turn, the accuracy of the PC-based pseudo-inverses depends on how well the local KL expansions represent each sample $\kappa^{(m)}$. For low correlated fields, the samples present a slowly oscillatory character, which can be captured using a small number of random variables, resulting into accurate PC-based pseudo-inverse matrices and high performance index. As the correlation length decreases, the sample field presents more and more high frequency oscillations, which are more pronounced for high variance. Since the local truncated KL expansions cannot capture the components of the

field associated with these high frequency oscillations, the accuracy of the resulting PC-based pseudo-inverses deteriorates. As a result, the performance of the PC-NN preconditioner also deteriorates with the emergence of high frequency oscillations. This explains the deterioration of the PC-NN performance index as the variance increases and as the correlation length decreases, reported in Figure 5.2.

Comparing Figure 5.2 with Figure 5.1, one can conclude that the performance of the PC-NN preconditioner is in general much higher than the M-NN preconditioner. The reason for this is that the median-based pseudo-inverses correspond to the PC-based pseudo-inverses that use $N_{KL} = 0$.

The performance of the PC-NN preconditioner for high variance and low correlation length reported in this experiment can improved by tuning the stochastic discretization parameters. The idea is to use more accurate KL expansions and high degree PC bases so that the resulting PC-NN preconditioner is closer to the NN preconditioner and yields similar convergence rates. In the following, the impact of the stochastic discretization parameters is detailed.

5.4.2 Impact of the Stochastic Discretization Parameters on the Performance of the PC-NN Preconditioner

The performance of the PC-NN preconditioner depends on how close the PC-based pseudo-inverses are to the exact pseudo-inverses for each sample. The accuracy of the PC-based pseudo-inverse matrices is inherited by the accuracy of the local PC expansions (5.23). The stochastic discretization parameters, namely, the number of random variables and the sparse grid level, control the accuracy of these PC expansions. For that reason, these parameters play a relevant role into the overall performance of the PC-NN preconditioner. The stochastic discretization parameters are related to three sources of error to the local PC expansions, which are now summarised.

The first source of error is inherited to having a finite representation of the coefficient field over each subdomain. By truncating the local KL expansion to its $N_{KL}^{(d)}$ dominant modes, dependencies on higher order modes are disregarded. In practice, it means that the effects of the high frequency components in the pseudo-inverses were lost.

Second, the pseudo-inverses based on the local KL truncated expansion of the field are approximated using a finite dimensional polynomial basis of stochastic polynomials in the retained $N_{KL}^{(d)}$ variable. Using a finite PC basis yields a PC truncation error.

Third, and final, the PC coefficients $[\mathrm{H}^{(d)}]_\alpha$ must be estimated and this estimation involves some error such that the computed PC expansion orthogonal projection of the (KL-based) pseudo-inverse in the space of the PC basis. This error is called the projection error. The PSP method is designed to minimize the PC truncation error while controlling the projection error. As the level increases, the sparse grid is completed with new points that improve the estimation of the estimation of the projection coefficients, while the PC basis is also extended. Therefore, both the PC truncation error and the PC projection error reduce with the level.

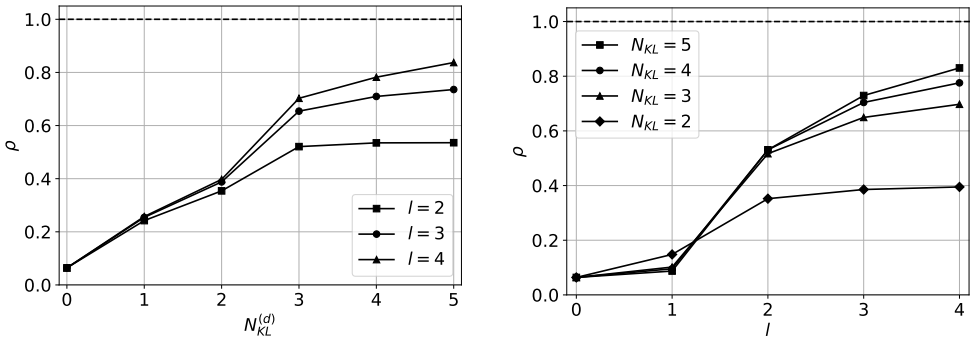
The objective of the following numerical experiments is to illustrate the impact of the stochastic discretization parameters on the performance of the PC-NN preconditioner. The following numerical experiments use a fixed number of subdomains, which implies that the accuracy of the KL expansions will be solely dependent on the number of local random variables used. The impact of changing the number of subdomains will be analysed later on, in Section 5.4.3.

The analysis is based on a test case that is set up as follows: The mesh is the same as previous experiments (with $N_{\mathrm{el}} = 32\,685$ elements and $N_{\mathrm{nod}} = 65\,858$ nodes), and partitioned into $D = 200$ subdomains, producing an interface with $N_\Gamma = 6\,823$ nodes. The field considered has $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$, values that were the most challenging in the previous numerical experiments. The results in Figure 5.3 compare the evolution of the performance index for different stochastic discretization parameters. The two plots are analysed in detail, in the following.

5.4.2.1 Impact of the number of random variables (fixed partition)

Subfigure 5.3a illustrates the variation of the performance index with the number of local random variables. As the number of local random variables increases, the local KL expansions represent the field more and more accurately. The PC-based pseudo-inverse matrices generated are then better approximations of the exact pseudo-inverses associated with the NN preconditioner. As a result, the performance of the PC-NN preconditioner improves.

Subfigure 5.3a also shows that the monotonic behaviour of the performance slows down



(a) Impact of $N_{KL}^{(d)}$.

(b) Impact of l.

Figure 5.3: Comparison of the performance index of the PC-NN preconditioner for different stochastic discretizations parameters.

or even stagnates after using a few random variables. This behaviour is explained by the progressive emergence of PC error (truncated and projection) that becomes dominant as $N_{KL}^{(d)}$ increases. Once the PC error becomes dominant, the performance can only improve by increasing the parameter l, and the stagnation of performance depends on the sparse grid level.

5.4.2.2 Impact of the sparse grid level

The impact of the sparse grid level in the performance of the PC-NN preconditioner is illustrated in Figure 5.3b. The performance index increases with the sparse grid level. The reason for this behaviour is that both the PC truncation and the projection errors decrease as l increases. For a given value of $N_{KL}^{(d)}$, the performance stagnates when l is large enough such that the KL error dominates. Consequently, using a larger l (larger PC basis and larger set of sparse grid points) becomes less and less effective, being necessary to increase $N_{KL}^{(d)}$ to improve the performance.

5.4.3 Effect of the Size of the Partition on the Performance of the PC-NN Preconditioner

The experiments reported in Figure 5.3 used a fixed partition, such that, the KL truncation error depended solely on the number of random variables used. However, for fixed number of random variables per subdomain, the fraction of the field's energy retrieved by the local KL expansions

depends on the size of the subdomains. According to the results of the previous chapter, the fraction of the field's energy retrieved by a local truncated KL expansion using a fixed number of random variables increases with D (see Figure 4.10, for instance). Therefore, the smaller the subdomains are, the more accurate the local KL expansions become, leading to a better preconditioner's performance (see Figure 4.12a, for instance). Since the performance of the PC-NN preconditioner is also directly related with the accuracy of the local KL expansions, as seen just above, the conclusions from the previous chapter suggest that reducing the size of the subdomains may also have a positive impact on the PC-NN's convergence rate. The objective of the following numerical experiments is to investigate the impact that the size of the subdomains have on the performance of the PC-NN preconditioner.

Figure 5.4 reports the variation of the PC-NN performance index with the number of subdomains for a fixed mesh with $N_{\mathrm{nod}} = 65\,858$ nodes. The experiment is based on a coefficient field with $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$. For $D \leq 200$, the performance index increases quickly with D because the KL truncation error is reduced. This reduction in the KL truncation error induces a drastic improvement of the performance of the PC-NN preconditioner. For $D > 200$, the KL truncation error is already small, and the PC truncation and projection errors become dominant, similar to the situation reported in Subfigure 5.3a. For this reason, the performance of the PC-NN preconditioner stagnates. Also, since the NN preconditioner is scalable [5], this behaviour implies that the PC-NN preconditioner is also scalable with the number of subdomains.

The results above show that the performance index converges asymptotically with the number of subdomains to some constant function of l. The performance index gets closer to 1 as both the PC truncation and projection errors get smaller, by increasing l. Therefore, provided that the sparse grid level is large enough, the performance index associated with some fixed partition and $N_{KL}^{(d)} > 1$ can also be obtained using $N_{KL}^{(d)} = 1$ and a larger number of subdomains.

Figure 5.5 reports the evolution of the performance when using only one random variable per subdomain ($N_{KL}^{(d)} = 1$). For $N_{KL}^{(d)} = 1$, one can use large level $l = 6$ (PC expansion of polynomial degree 64), ensuring low PC truncation and projection errors. The performance index increases as long as the KL truncation error is not dominated by the PC truncation and projection errors with D. Since the sparse grid level used is large ($l = 6$), the performance index converges to a

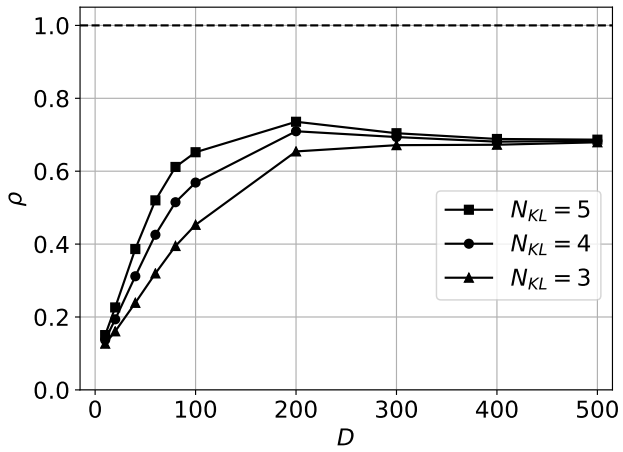


Figure 5.4: Variation of the PC-NN performance index with the number of subdomains for a fixed mesh with $N_{\text{nod}} = 65\,858$ nodes. Coefficient parameters: $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$. Sparse grid level: $l = 3$.

constant close to 1. The coefficient field considered has a slightly longer correlation and lower variance than is the rest of the section ($\sigma_{\boldsymbol{g}}^2 = 1$ and $\ell_c = 0.1$) to emphasise this effect.

An important remark concerns the dimension of the matrix $[\mathrm{W}]$ whose size is equal to the number of floating subdomains. For large numbers of subdomains the Cholesky factorization of $[\mathrm{M}]$ becomes costly, suggesting that future works should consider better strategies for the projection, as discussed in Section 5.5. In any case, this experiment demonstrated that the PC-NN preconditioner can approach the NN performance (up to PC error) for sufficiently small subdomains. In practice, there exists an optimal number of subdomains to balance the Cholesky factorization cost with the workload within each subdomain. This optimal D depends on a number of factors, including the coefficient field, the parallel architecture, the type of the partition, etc. The *a priori* definition of this optimal D remains an open problem.

5.4.4 Scalability of the PC-NN Preconditioner

The previous section showed that the performance of the PC-NN preconditioner improves by reducing the size of the subdomains. However, changing the size of the partition can also be useful for other reasons. For example, increasing D reduces the number of local degrees of freedom of the local problems, making their resolution faster. Using large D is also necessary

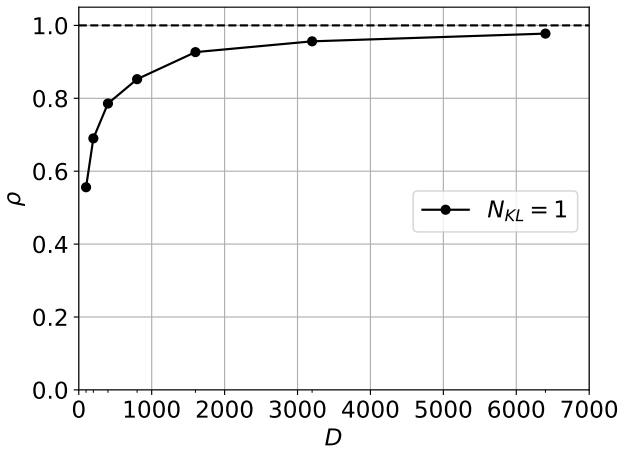


Figure 5.5: Variation of the PC-NN performance index with the number of subdomains for a fixed mesh using a single local random variable and $l = 6$. Fixed problem with $N_{\text{el}} = 32\,685$ elements and $N_{\text{nod}} = 65\,858$ nodes. The field considered has $\sigma^2_{\boldsymbol{g}} = 1$ and $\ell_c = 0.1$.

when the stochastic field and the sampled solution require a fine FE mesh. Refining the mesh using a constant D may not be an option since the number of degrees of freedom would become large and, consequently, the local operations would be excessively costly. The alternative is to refine the mesh while keeping the number of local degrees of freedom per subdomain constant. Then, the highly oscillatory field can be adequately characterized with a finer mesh while the workload associated with each subdomain is the same.

In the following, we consider two types of experiments with the partition: a) partition a fixed FE mesh using more and more subdomains (resulting in a reduction of the local problems size), and b) an increasing partition of the subdomain with a refinement of the FE mesh to maintain a fixed number of degrees of freedom per subdomain. Scenarios a) and b) are often called *strong* and *weak* scalability, respectively. These two different scenarios are relevant to access the parallel efficiency of the preconditioner.

Figure 5.6 summarizes the scalability results for PC-NN preconditioner and compares it with the one of the NN and M-NN preconditioners. The test case is based on a field with $\sigma^2_{\boldsymbol{g}} = 2$ and $\ell_c = 0.05$. The sparse grid level used to construct the PC-NN preconditioner is fixed to $l = 3$. Subfigures 5.6a and 5.6b report the strong and weak scalability, respectively, of the NN and the PC-NN preconditioners. The histograms in Subfigure 5.6c detail the distribution of the number

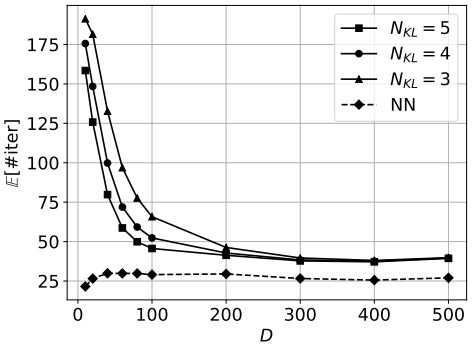
of iterations for the NN and the PC-NN preconditioners with a different numbers of subdomains. Finally, Subfigure 5.6d contrasts the evolution of the average number of PPCG iterations using M-NN preconditioner against the other two (scalable) preconditioners.

The evolution of the average number of iterations reported in Subfigures 5.6a and 5.6b is very similar. The average number of iterations using the NN preconditioner increases slightly from $D = 10$ (first point) to $D = 40$ (third point). This behaviour is explained by the increase in the size of the problem. In addition, the emergence of floating subdomains, and consequent need for projection-based iterations also contributes to the initial increase of the number of iterations. However, for $D > 40$ the number of iteration stagnates, showing that the NN preconditioner combined with the Nicolaides coarse space produces a scalable method, as expected.

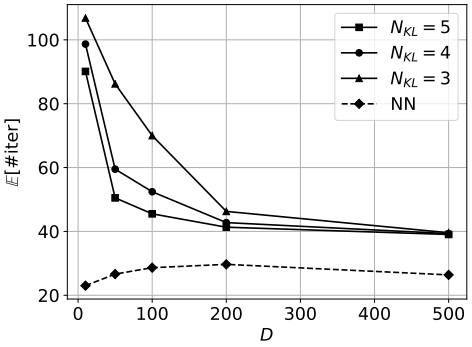
The evolution of the average number of iterations using the PC-NN reported in Subfigures 5.6a and 5.6b has two main patterns. For $D < 200$, the average number of iterations decreases significantly. This behaviour is explained by the significant reduction of the KL truncation error and consequent improvement of the PC-NN performance, which approaches the NN's one. For $D > 200$ the PC truncation and projection error become dominant and the number of iterations stagnates, to a value independent of $N_{KL}^{(d)}$. The number of iterations for the PC-NN preconditioner stagnates at very similar values to the standard approach, thanks to the sparse grid level ($l = 3$), which ensures small PC truncation and projection errors. Further, the histograms shown in Subfigure 5.6c indicate that the spread of the distribution of the number of iterations using the PC-NN approach reduces with D (strong scalability experiment). It shows that the extreme realizations of the field are better treated when D is large. This behaviour is relevant to the parallel resolution of multiple samples as a low variance in the number of iterations will ease balancing the computational load.

Finally, Subfigure 5.6d compares the evolution of the average number of iterations in the strong experiment using the M-NN preconditioner. To emphasise the M-NN preconditioner's lack of scalability, the plot also reports the evolution of the iterations number for the (scalable) NN and PC-NN preconditioners. The number of iterations associated with the M-NN preconditioner increases rapidly with the number of subdomains. When D increases further, more iterations are needed, although the degradation is not as dramatic after $D = 400$ as initially. This is consistent with the results and analysis of Section 5.4.1.1. As a result of the degradation of the

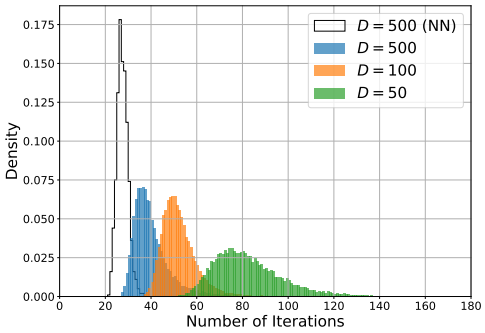
M-NN performance, this preconditioner is not scalable with D.



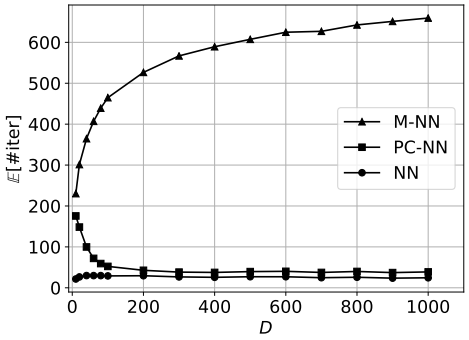
(a) Fixed problem size $N_{\text{nod}} = 65\,858$.



(b) Fixed ratio $\frac{N_{\text{nod}}}{D} \approx 330$.



(c) Dist. of iter. for fixed D (PC-NN and NN).



(d) Fixed problem size $N_{\text{nod}} = 65\,858$.

Figure 5.6: Top plots report strong (top-left) and weak (top-right) scalability analysis of the performance of the PC-NN ($l = 3$) and NN preconditioners. Histograms of the number of iterations for the PC-NN preconditioner for a different number of subdomains (strong scalability experiment), and to the NN preconditioner for just $D = 500$. Bottom-right plot shows the strong scalability analysis of the M-NN preconditioner. The field considered has $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$.

5.5 Alternative Sample-Dependent Coarse Space

As discussed before, the resolution of the Schur system by a PCG method using the straightforward application of the NN preconditioner is not a robust approach because the number of iterations increases with the number of subdomains in the partition. The origin of the problem is related to the singularities of the NN maps. To produce a robust method, the Nicolaides coarse space was introduced, and the PCG method extended to the PPCG, resulting in a scalable

method. However, for more challenging problems, the Nicolaides coarse space may not always be sufficient. For example, the singularities associated with linear elasticity models relate to rigid body motion, which are not characterized by constant basis functions [89, 60, 39]. The effect of the singularities is not the only cause of low convergence. Fields with discontinuities and highly contrasted values are particularly challenging for NN preconditioners, whose local nature tends to slow down the residual propagation across discontinuities [52, 43, 89]. Similarly to the effect of the singularities of the NN maps, a separate treatment of the slow residual components can speed up the iterative scheme.

One generalization of the Nicolaides coarse space is the so-called *Generalized Eigenproblems in the Overlap coarse space*, or in short *GenEO coarse space*. This space was introduced in [116, 118] and applied with the PPCG method in the context of linear elasticity models with heterogeneous coefficient fields, for example, [117]. The GenEO coarse space is constructed from the singular (Nicolaides) and nearly singular modes of the NN maps.

Contrary to the Nicolaides coarse space, the construction of the GenEO coarse space is sample-dependent and the nearly singular modes must be determined for each sample. The identification relies on solving a set of generalized eigenvalue problems defined for each subdomain, which represents a significant computational burden. This section proposes an alternative to the GenEO coarse space construction which is specifically adapted to the resolution of multiple sampled problems. The idea is to rely on fixed local subspaces, based on the median field, and re-use these to define the coarse space for each sample. Opposite to the median-based NN preconditioners presented before, the median-based GenEO turns out to be an effective alternative to the full coarse space construction of each sample, even for highly variable and low correlated fields, provided it is used with the PC-NN preconditioner. In particular, the performance of the median-based GenEO coarse space is close to the original coarse space for sufficiently small subdomains.

In this thesis, we restrict ourselves to the elliptic equation (2.1) with a.s. continuous fields Since the Nicolaides coarse space with the PC-NN preconditioner is already scalable for these problems, using a richer GenEO coarse space does not make such a difference regarding scalability. Therefore, this section only intends to focus on the possibility of reducing the coarse space construction time, by relying on fixed local subspaces.

The section is organized as follows. First, the GenEO problem is provided. Then, the median-based GenEO coarse space is introduced. Finally, several numerical experiments illustrate the potential of the median-based GenEO coarse space.

5.5.1 The GenEO Coarse Space

In the following, we introduce the GenEO coarse space construction. The discussion puts particular emphasis on the cost of this construction. The material presented here is based on the description in [40, Chapter 7].

The GenEO problem associated with subdomain $\Omega^{(d)}$ is the generalized eigenvalue problem given by

$$[\mathrm{S}^{(d)}]\mathrm{v}^{(d)} = \zeta^{(d)}[\mathrm{N}^{(d)}]\mathrm{v}^{(d)} \quad \text{with} \quad [\mathrm{N}^{(d)}] := [\mathrm{R}^{(d)}][\mathrm{S}][\mathrm{R}^{(d)}]^\top. \tag{5.30}$$

Problem (5.30) is symmetric, $[\mathrm{N}^{(d)}]$ is SPD such the eigenvalues ζ are non-negative, and the resulting eigenvectors $\mathrm{v}_j^{(d)}$ are $[\mathrm{S}^{(d)}]$-orthogonal and $[\mathrm{N}^{(d)}]$-orthonormal. We order the eigenvalues in increasing order: $0 \leq \zeta_0^{(d)} \leq \zeta_1^{(d)} \leq \ldots \leq \zeta_{N_\Gamma^{(d)}}^{(d)}$. In particular, if $\Omega^{(d)}$ is a floating subdomain $\zeta_0^{(d)} = 0$ and the associated eigenvector is a constant vector.

For a fixed parameter $\tau \geq 0$, we consider the local subspace of $\Omega^{(d)}$ defined as

$$\mathcal{G}_\tau^{(d)} := \mathsf{span}\{\mathrm{v}_j^{(d)} : \zeta_j^{(d)} \leq \tau\}. \tag{5.31}$$

The subspace $\mathcal{G}_\tau^{(d)}$ is the subspace of the "most singular" eigenvalues of the NN map. The total number of eigenvectors retrived is

$$N_G := \sum_{d=1}^{D} \dim \mathcal{G}_\tau^{(d)} \geq D_{\mathrm{in}}. \tag{5.32}$$

Moreover, consider an extension of the matrix $[\mathrm{W}]$ given in (5.8) through

$$[\mathrm{W}] := [\mathrm{w}_1^{(1)}|\cdots|\mathrm{w}_{\dim \mathcal{G}_\tau^{(1)}}^{(1)}|\cdots|\mathrm{w}_1^{(D)}|\cdots|\mathrm{w}_{\dim \mathcal{G}_\tau^{(D)}}^{(D)}] \in \mathbb{R}^{N_\Gamma \times N_G}. \tag{5.33}$$

where $\mathrm{w}_j^{(d)} := [\check{\mathrm{R}}^{(d)}]^\top \mathrm{v}^{(d)}$. The GenEO coarse space is the subspace with dimension N_G spanned by the columns of the matrix $[\mathrm{W}]$. The Nicolaides coarse space corresponds to the

limit $\tau \to 0$ and is the GenEO coarse space with the smallest dimension, $N_G = D_{\text{in}}$.

The resolution of each sample problem is carried out using the PPCG method defined in Algorithm 6. In particular, the definition of the projection matrix $[\Pi]$ in (5.11) is kept with the matrix $[\mathrm{W}]$ defined in (5.33). As τ increases, the GenEO coarse space dimension also increases, and the initial condition (5.14) is closer to the solution. More importantly, the projected preconditioned residual has less and less slow components and the iterations converge faster. However, the set-up step of Algorithm 6 is more expensive for the GenEO coarse space compared to the Nicolaides coarse space approach. The generalized eigenvalue problems (5.30) must be solved for all subdomains. Note that the matrices $[\mathrm{N}^{(d)}]$ depend not only on their subdomain $\Omega^{(d)}$ but also on the neighbours of $\Omega^{(d)}$. This implies communication between neighbours for the parallel resolution of these problems, which may impact parallel efficiency. The GenEO problem can be solved using a direct method if the subdomains are not too large. In this case, all eigenvectors become available. Alternatively, matrix-free strategies, such as the Arnoli iteration [130], can also be used, in which case, only a small portion of the eigenvectors associated with each problem is sought. For the case of $\tau = 0$, the resolution of (5.30) is unnecessary, and the Nicolaides procedure above is used.

Once the local eigenvectors associated with the subdomains are selected, the matrix $[\mathrm{M}]$ and its Cholesky decomposition are computed. Similarly to the Nicolaides coarse space, the computation of the product $[\mathrm{S}][\mathrm{W}]$ can be split into local contributions as in (5.13), by exploiting the local block structure of $[\mathrm{S}]$ and $[\mathrm{W}]$. However, the cost of this product is more expensive in the GenEO approach because the coarse space is larger. Another important observation is that the size of the matrix $[\mathrm{M}]$ increases with τ, which makes its Cholesky factorization more expensive and its storage possibly challenging.

For fixed τ, the number of eigenvectors retained in a subdomain tends to be smaller as the size of the subdomain decreases. For this reason, the growth of the dimension of $[\mathrm{W}]$ is sub-linear with D. Moreover, the dimension of $[\mathrm{W}]$ grows at a faster rate when associated with larger τ. As a result, τ should be kept as low as possible, achieving scalability while maintaining a minimal workload.

5.5.2 The Median-Based GenEO Coarse Space

Solving the generalized eigenvalue problems (5.30) for each sample is expensive. This subsection exploits alternatives to bypass the resolution of these problems. The idea is to consider fixed local subspaces defined as

$$\bar{\mathcal{G}}_{\tau}^{(d)} := \text{span}\{\bar{\mathrm{v}}_j^{(d)} : \bar{\zeta}_j^{(d)} \leq \tau\}, \tag{5.34}$$

where $(\bar{\mathrm{v}}_j^{(d)}, \bar{\zeta}_j^{(d)})$ are the eigenpairs of the median-based generalised eigenvalue problems (5.30). The coarse space formed by the resulting median-based eigenvectors will be called *median-based GenEO (M-GenEO) coarse space*. The framework used to perform the PPCG method with an M-GenEO coarse space is essentially the same as the one used with an original GenEO coarse space. The only difference is that the median-based GenEO problem (5.30) is solved once and for all, before the sampling stage, and re-used in (5.9) for all samples. Only the projection operator (matrix $[\mathrm{M}]$ and its decomposition) is set for each sample.

5.5.3 Numerical Comparison Between GenEO and M-GenEO

We first conduct a set of numerical experiments to compare the GenEO and the M-GenEO methods. The goal of these numerical experiments is to understand how much the performance of the M-GenEO coarse space differs from the original GenEO method in terms of performance, and in particular, if it can compute efficiently multiple solution samples.

Throughout the following experiments, the size of the coarse space is reported by its relative dimension w.r.t. the number of interface nodes (*i.e.*, size of the Schur problem), using

$$C := \frac{\mathbb{E}\left[N_G\right]}{N_{\Gamma}}. \tag{5.35}$$

The relative size C quantifies the reduction of the projection space relative to Schur problem size. Three different costs increase with C, or equivalently, with the dimension of the coarse space. Firstly, note that the number of local GenEO eigenvectors of each subdomain $\Omega^{(d)}$ corresponds to the number of local problems (columns of $[\mathrm{W}^{(d)}]$) associated with $\Omega^{(d)}$ at the

construction of $[\mathrm{M}]$. Since, the number of local GenEO eigenvectors increases with C, the construction of $[\mathrm{M}]$ becomes more expensive. Secondly, the size of $[\mathrm{M}]$ increases with N_G; therefore, its Cholesky factorization is more expensive. Finally, since the factorization $[\mathrm{M}]^{-1}$ and the local matrices $[\mathrm{W}^{(d)}]$ become larger, the application of $[\Pi]$ at each PPCG iteration also becomes more expensive. Therefore, C should be kept as small as possible.

Throughout this section, the numerical experiments are based on a mesh with $N_{\mathrm{el}} = 32\,685$ elements and $N_{\mathrm{nod}} = 65\,858$. With exception to Figure 5.10, all numerical experiments use a partition with $D = 100$ subdomains ($D_{\mathrm{in}} = 66$), producing $N_\Gamma = 4\,841$ interface nodes. Unless otherwise specified, the coefficient field with $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$ is the standard field used.

5.5.3.1 Impact of the coarse space relative dimension

As the dimension of the coarse space increases, a lower number of iterations is required to converge. We start by analysing the impact of using the approximated M-GenEO coarse space on the convergence. To this end, Figure 5.7 illustrates the average number of iterations produced using the M-GenEO and the original GenEO coarse spaces, combined with the NN preconditioner, as a function of C. The behaviour of both curves is explained in the following.

For the first point on the curves ($C = 0.014$), the GenEO and the M-GenEO coarse spaces correspond to the Nicolaides coarse space ($N_G = 66$). Therefore, both methods produce the same number of iterations. As C increases, the methods require less and less iterations, although the M-GenEO is lagging behind the GenEO, until $C = 1$, where the initial guess is the solution (no iteration needed). Naturally, the two opposite sides of the horizontal axis represent distinct cost-benefit relations, where a smaller C is associated with a higher number of iterations and lower projection and $[\mathrm{M}]$ inversion costs, and vice-versa. In any case, due to the increasing cost of the projection with the dimension of the coarse space, the practical values of C are positioned towards the left of the horizontal axis, that is $C \ll 1$.

Figure 5.7 also shows that the rate at which the number of PPCG iterations decreases with C differs between the two coarse spaces. The GenEO coarse space is more effective at treating the components of the solution that slow down the PPCG scheme, and the convergence of the method significantly improves by just slightly increasing the coarse space dimension, compared

to the convergence with the Nicolaides. For instance, from $C = 0.014$ ($\mathbb{E}[N_G] = 66$) to $C = 0.018$ ($\mathbb{E}[N_G] = 172$), the average number of iterations drops from 29 to 17. Each increase in the GenEO coarse space improves the overall PPCG convergence. The M-GenEO coarse space is not as effective as its GenEO counterpart at treating slow residual components. For instance, from $C = 0.014$ to $C = 0.018$ (same range as before) the number of iterations has a smaller drop, going from 29 to 25. It is only at $C = 0.06$ ($N_G = 338$) that the M-GenEO coarse space can produce 18 iterations, roughly the same number of iterations that the GenEO produced at $C = 0.018$ ($\mathbb{E}[N_G] = 172$). Similarly to the GenEO case, as C keeps increasing, the improvement of convergence levels-up and gains obtained in terms of iterations are progressively smaller. When $C = 1$, both coarse spaces spam $\mathbb{R}^{N_\Gamma}$ and the solution is obtained in one iteration.

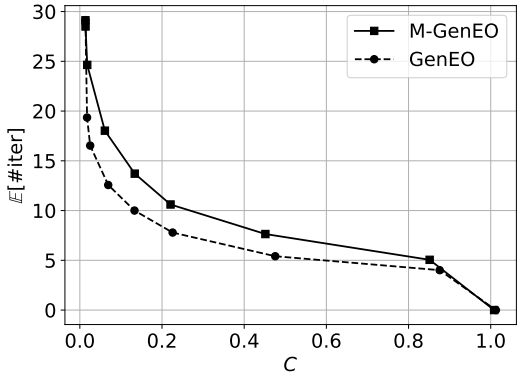


Figure 5.7: Average number of PPCG iterations using NN preconditioner as a function of the coarse space dimension. Compared are the M-GenEO and the GenEO coarse spaces.

The results in Figure 5.7 show that larger coarse spaces are associated with smaller numbers of iterations. To better appreciate the improvements due to the generalized coarse spaces, relative to the Nicolaides coarse space, we introduce the index

$$\rho_C := 1 - \mathbb{E}\left[\frac{\#_{\text{GenEO}} \text{ or } \#_{\text{M-GenEO}}}{\#_{\text{Nico}}}\right], \tag{5.36}$$

where $\#$ corresponds to the number of iterations performed using either the NN or PC-NN preconditioner (specified along with the discussion) combined with either the GenEO, the M-GenEO or the Nicolaides coarse space. Throughout this thesis, both numerator and denomerator use

the same preconditioner. The larger the performance index ρ_C is, the more effective is the introduction of the generalized coarse space.

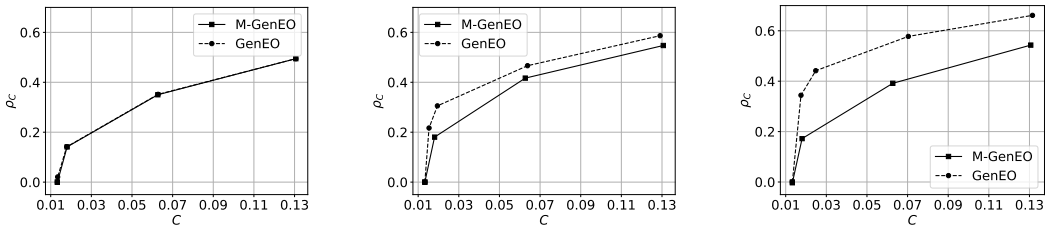
Several problem-dependent factors influence the improvement provided by the coarse spaces, namely, the variance and correlation length of the coefficient field, the partition considered, and the type of preconditioner used in association with the coarse space. These aspects are analysed in detail in the following.

5.5.3.2 Impact of the field complexity

We start by investigating the impact of the correlation length on the performance of the two coarse spaces. Figure 5.8 illustrates the variation of the performance index of the two coarse spaces for different correlation lengths, and for a fixed (high) variance $\sigma_{\boldsymbol{g}}^2 = 2$. The coarse spaces are combined with the NN preconditioner. The correlation length used for each plot and the number of iterations using the Nicolaides coarse space are provided in the subcaptions.

The two subspaces yield nearly identical performance when ℓ_c is large ($\ell_c = 1$, left plot). This is explained by the fact that for large ℓ_c, the samples $\kappa^{(m)}$ are nearly constant over $\Omega^{(d)}$ such that $\overline{\mathcal{G}}_\tau^{(d)}$ is an excellent approximation of $\mathcal{G}_\tau^{(d)}$. When the complexity of the field increases (ℓ_c decreases), using an extended coarse space becomes more interesting as it is more susceptible to improve convergence. This observation is valid for the two methods, although GenEO exhibits better performance index. This trend emphasizes the importance of adapting the coarse space construction when the field over the subdomains presents oscillations. Indeed, the median field cannot accurately account for such effects.

Having investigated the impact of the correlation length on the performance of the two coarse spaces, we now investigate the impact of the variance. Figure 5.9 illustrates the evolution of the performance index of the coarse spaces combined with the NN preconditioner, with the variance. Each subcaption gives the variance of the field and the average number of iterations performed using the Nicolaides coarse space. The plots highlight two important behaviours. Firstly, the lower efficiency of the M-GenEO coarse space (the gap between the two curves) increases with the variance. Secondly, for a fixed C, the performance index provided by each coarse space increases with the variance. These two aspects are explained separately in the

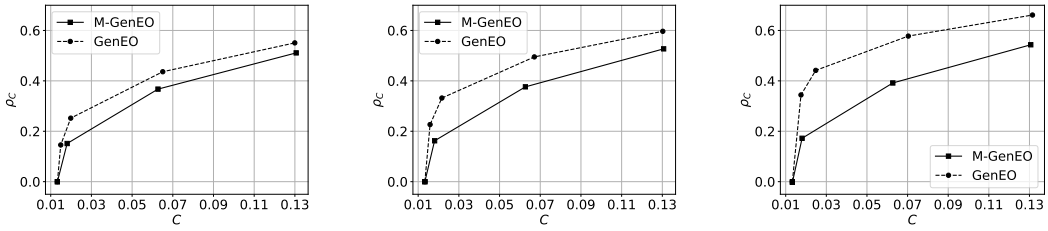


(a) $\ell_c = 1.$ (19) (b) $\ell_c = 0.1.$ (24 ± 2) (c) $\ell_c = 0.05.$ (30 ± 3)

Figure 5.8: Performance index as a function of the coarse space dimension C and for different correlation lengths. The experiment uses NN preconditioner. Coefficient field with $\sigma_{\boldsymbol{g}}^2 = 2$. The average number of iterations with one standard deviation bound using Nicolaides coarse space is indicated for each plot.

following two paragraphs.

For low variance, the amplitude of the field's oscillations is very small, even though the correlation length is very small. Therefore, the field is nearly constant and the median is a good representation, explaining the similar performances. As the variance increases, the amplitude of the field's oscillations also increases, and the median becomes a very poor representation of the field. As a result, the M-GenEO coarse space yields poorer performances compared to the original GenEO approach, that is adapted to each sample. Figure 5.9 also indicates an overall improvement of ρ_C when $\sigma_{\boldsymbol{g}}^2$ increases, particularly for the standard GenEO, and to a lesser extend for M-GenEO. This trend can be explained by the emergence of more slow residual modes as $\sigma_{\boldsymbol{g}}^2$ increases (due to the emergence of more extreme events), such that richer subspaces (compared to the Nicolaides) are even more effective.



(a) $\sigma_{\boldsymbol{g}}^2 = 0.5.$ (22 ± 1) (b) $\sigma_{\boldsymbol{g}}^2 = 1.$ (25 ± 2) (c) $\sigma_{\boldsymbol{g}}^2 = 2.$ (30 ± 3)

Figure 5.9: Performance analysis of larger coarse spaces with the variance. The experiment uses the NN preconditioner. Coefficient field with $\ell_c = 0.05$. The average number of iterations with one standard deviation bound using Nicolaides coarse space is described for each plot.

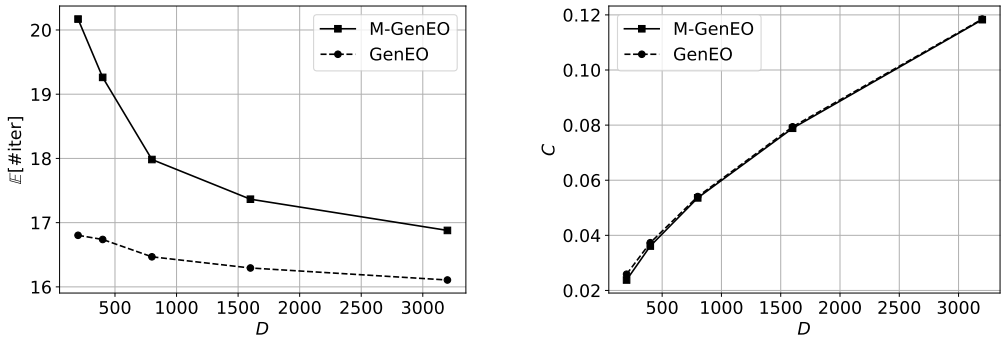
5.5.3.3 Impact of the different number of subdomains

Figure 5.8 showed the importance the field's complexity on the efficiency of the M-GenEO coarse space. We expect the M-GenEO's performance to improve as one makes the subdomains smaller, if it is the local complexity that matters. In particular, if the local correlation length is large, it implies that the local part of the field is proportional to the field's median ($\kappa^{(m)} \approx c\bar{\kappa}, c \in \mathbb{R}$). Likewise, the exact matrices $[\mathrm{S}^{(d)}]$ and $[\mathrm{N}^{(d)}]$ also become nearly proportional to their median-based pairs. As a result, the subspace spanned by the local eigenvectors should be close to the local subspace spanned by the median-based eigenvectors. Hence, provided that the subdomains are small enough, the M-GenEO and GenEO coarse spaces should yield similar convergence rates. Subfigure 5.10a illustrates this effect for $\tau = 0.3$, reporting for the two coarse spaces, the number of iterations as a function of D. In order to emphasize the effect of the apparent correlation, the correlation length considered in this experiment is $\ell_c = 0.1$ (larger than the one that has been used previously). This experiment shows that the additional number of iterations between the two coarse spaces reduces with D. As D changes, the dimensions of the coarse space also changes, as well as N_Γ. Therefore, Subfigure 5.10b reports the evolution of the relative coarse space dimension C with D (sample average C for GenEO). The curves show that C increases with D and remains close for the two subspace constructions.

From the previous experiments, one can conclude that the reduction in the number of iterations observed in Subfigure 5.10a is due to the increase in the relative subspace dimension, while the reduction of the number of additional iterations for the M-GenEO is due to the increasing apparent correlation length.

5.5.3.4 Effect of the preconditioners

The following experiments analyse the performance of the two coarse spaces when combined with the M-NN and PC-NN preconditioners. Figure 5.11 reports the evolution of the average number of iterations of both subspaces with C, using different preconditioners. The preconditioners considered are progressively more effective and eventually converge to the NN preconditioner. The test case used is based on a coefficient field with $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.05$, while PC-NN preconditioner uses $l = 4$.

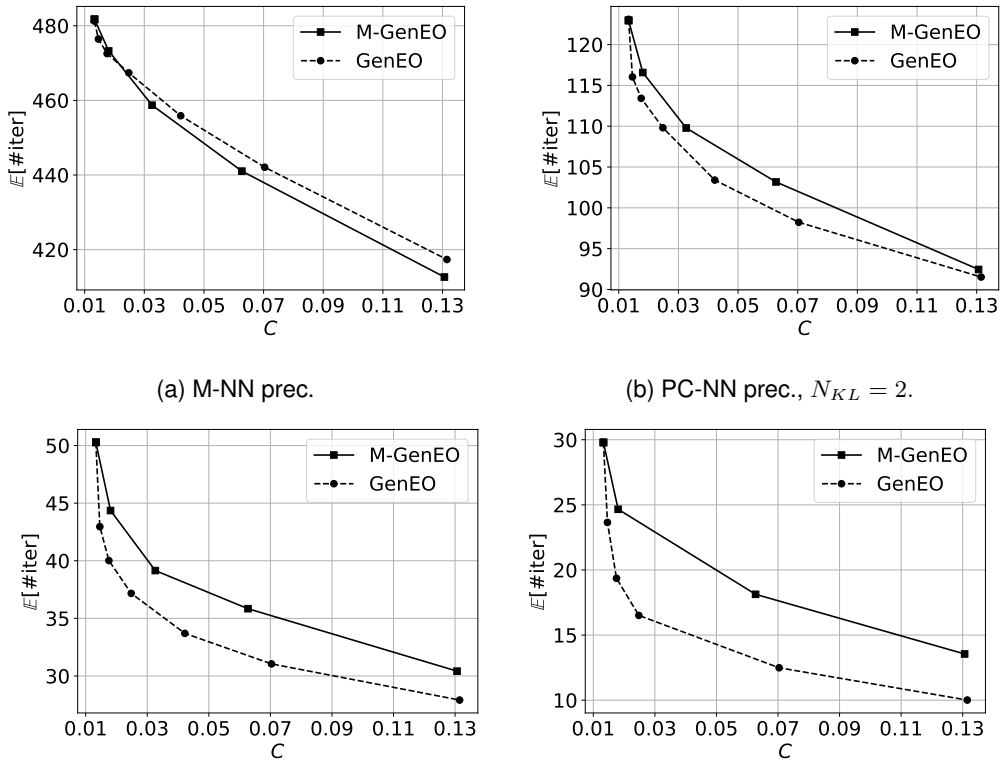


(a) Impact on the average number of iterations. (b) Impact on the relative coarse space dim.

Figure 5.10: Impact of the number of subdomains on the average number of iterations (left plot) and on relative coarse space dimension (right plot). Coefficient field with $\sigma_{\boldsymbol{g}}^2 = 2$ and $\ell_c = 0.1$. Coarse spaces with $\tau = 0.3$ and combined with the NN preconditioner.

Subfigure 5.11a reports improvement obtained with the M-NN preconditioner. Compared to the Nicolaides coarse space, the improvement brought by the two GenEO coarse spaces is limited, and as a result, their difference is negligible. For the smallest values of C, the method with M-GenEO coarse space converges faster than the original approach. As C increases, the GenEO coarse space becomes more effective. In any case, one can observe that larger dimensional coarse spaces only slightly reduce the number of iterations. For example, from $C = 0.014$ to $C = 0.018$, the size of the coarse spaces increases by 33% ($\mathbb{E}[N_G] = 66$ to $\mathbb{E}[N_G] \approx 87$). However, the average number of iterations decreases from ≈ 480 only to ≈ 470 iterations (about 2% reduction). This suggests that the M-NN preconditioner is responsible for the resulting performance. Subfigure 5.11b concerns the PC-NN preconditioner with $N_{KL} = 2$. The average number of iterations of the GenEO coarse space is consistently smaller than for the M-GenEO for all values of C reported. However, the improvements on reducing the number of iterations are poor and only slightly better than the previous M-NN preconditioning approach, as we may have anticipated for the considered number of subdomains $D = 100$ and correlation length $\ell_c = 0.05$. Indeed, from $C = 0.014$ to $C = 0.018$ (same range as before), the number of iterations decreases from 123 to 115 iterations (about 7% reduction). As the number of random variables increases, the improvement brought by the GenEO is more significant, as reported in Subfigure 5.11c, which illustrates the case of $N_{KL}^{(d)} = 4$. We notice here that the M-GenEO

starts lagging significantly behind the GenEO. For instance, from $C = 0.014$ to $C = 0.018$, the number of iterations with the GenEO coarse space goes from 50 to 40, where the M-GenEO coarse space can only achieve 40 iterations at $C = 0.033$. Eventually, the GenEO performs better than the M-GenEO, and the differences between the two approaches become more, and more pronounced as N_{KL} increases. Subfigure 5.11d reports the limiting behaviour of this trend, suggesting that the higher effectiveness of the GenEO coarse space compared to the M-GenEO is more pronounced when combined with a more effective preconditioner.



(a) M-NN prec.

(b) PC-NN prec., $N_{KL} = 2$.

(c) PC-NN prec., $N_{KL} = 2$.

(d) NN prec.

Figure 5.11: Evolution of the average number of iterations produced with the two coarse spaces combined with different preconditioners. Coefficient field with $\sigma^2_{\boldsymbol{g}} = 2$ and $\ell_c = 0.05$ and PC-NN preconditioners using $l = 4$.

5.5.4 Concluding Remarks on the GenEO-based Projection

In this section, we introduced the M-GenEO coarse space construction, which is the GenEO built on the median-based local subspaces. The key aspect of using the M-GenEO coarse space is that it does not require solving the generalized eigenvalue problems for each sample. Therefore, the M-GenEO coarse space construction is significantly cheaper than the original GenEO space construction.

This section provided several numerical experiments to analyse the performance of the two coarse spaces for different field complexities, different number of subdomains and different types of preconditioners. The results are summarised as follows:

- The differences between the M-GenEO and GenEO coarse spaces are noticeable only if combined with an effective preconditioner (either the NN or the PC-NN preconditioner with sufficiently refined stochastic discretization);
- The effect of substituting the sample-dependent local subspaces with the median-based counterparts is more and more noticeable as the apparent correlation length of the field over the subdomains increases.
- The performance of the GenEO for some fixed dimension can be achieved using a M-GenEO coarse space with larger dimension.
- Decreasing the subdomains' size reduces the difference between the GenEO and the M-GenEO, when τ is held fixed. However, this is accompanied by an increasing cost because of the growth of $[\mathrm{M}]$.

The GenEO and M-GenEO can improve the convergence of the PPCG algorithm compared with using the "minimal" Nicolaides coarse space. The problem of selecting τ as to balance the resulting reduction in the number of iterations with the additional cost of working with a higher dimensional coarse space is challenging, in particular, because it depends very much on the problem considered. In this work, we do not tackle this question, especially because the Nicolaides coarse space ($\tau = 0$) is already scalable for our problem. However, this section provided evidence that median-based coarse spaces may be a viable alternative to the standard GenEO coarse space in contexts where scalability demands more than the Nicolaides. Naturally,

the median-based approach will be more viable for partitions with small subdomains that yield locally well-correlated fields.

Note also that we tried to design a surrogate-based GenEO construction, in particular, using PC expansions of the local subspaces $\mathcal{G}_\tau^{(d)}$. One difficulty along this direction is the dependence of $\mathcal{G}_\tau^{(d)}$ on the random variables of $\Omega^{(d)}$ *and* of its neighbours. These dependencies make the PC construction more complex (not fully local), leading to higher dimensional approximations. Future work on this topic will focus on the way to tackle these dependencies such that the surrogate-based GenEO construction is viable.

5.6 Conclusion and Prospective Work

This chapter proposed a new DD method to efficiently generate samples of the solution from stochastic elliptic equations with random coefficients. Particular focus has been put on coefficients associated with large variance and low correlation length.

Each solution sample is generated by solving the Schur system of the elliptic problem associated with each sampled field using a CG-type method. The previous chapter presented a Schur-based preconditioning approach to solve each sampled problem by a PCG method using a preconditioner that is easily adapted to each sample. However, the application of this preconditioner involves the resolution of a "global" linear system with a size equal to the number of interface nodes, which can create limitations in the context of large problems. To overcome these issues, new preconditioning approaches are proposed in this chapter, which are totally based on local operations, promoting parallelism and applicability to large problems.

5.6.1 Summary of the Surrogate-based BDD method

The starting point of the proposed methods is the Neumann-Neumann (NN) method, which solves the Schur system by a PCG method using a specific preconditioner: the NN preconditioner. Contrary to the preconditioning approach of the previous chapter, the NN preconditioner involves solely local operations, namely, the resolution of a set of local problems at each iteration. Ideally, the subdomains are small enough so that the size of these local problems allows to

compute and store the local operators. However, constructing these operators for each sample can represent a significant computational burden, which should be avoided if many sampled problems are to be solved.

This chapter presented two alternatives to the NN preconditioner. One is a sample independent approach that uses the pre-computed operators corresponding to the median field for all samples, called the M-NN preconditioner. This approach leads to the median-based BDD method. Sample-independent approaches are not effective for high variable fields, as extensively illustrated throughout this thesis. To provide a more robust alternative in the context of such type of fields, this chapter proposes using PC-based approximations of the local operators, called the PC-NN preconditioner. This approach leads to the surrogate-based BDD method. The key aspect of the PC-NN preconditioner is that it bypasses most of the construction cost of the sample-dependent NN preconditioner using surrogate-based operations that are much cheaper to perform.

The PC-NN preconditioner is set-up during the pre-processing stage. It is formed of surrogates that approximate the pseudo-inverse matrices associated with the discrete NN local maps. The construction of these local surrogates relies on local PC expansions using random variables of the local KL expansion of the coefficient field over each subdomain. In fact, we construct PC expansions of the square-root of the local pseudo-inverses. This essential feature provides robustness and stability to the application of the PC-NN preconditioner.

The main drawback of the NN preconditioner is that the resulting number of PCG iterations depends on the number of subdomains and increases with the number of the interface nodes. These scalability issues are due to the singular local NN maps. A coarse space containing the singular modes of the NN maps, known as the Nicolaides coarse space, is then introduced to obtain a scalable PPCG iterative method.

The resolution of each sampled problem by a PPCG method using the NN preconditioner is known as the Balancing Domain Decomposition (BDD) method. The adaptation of the PC-NN preconditioner into the PPCG method gives rise to a surrogate-based BDD method. Compared to its original form, the surrogate-based BDD method alleviates the cost of constructing and applying the preconditioner, while preserving the scalability and parallel properties of the original approach.

Extensive numerical experiments compared the performances of the NN, M-NN and PC-NN preconditioners. The NN preconditioner is naturally the most effective in terms of the number of iterations, and for that reason, it is considered the reference. The PC-NN and M-NN preconditioners can be considered good alternatives to the NN preconditioner if they produce a similar number of iterations. The tests empirically demonstrated that the PC-NN and NN preconditioners require similar iterations to achieve a fixed tolerance, provided that the KL and PC expansions are accurate enough. In addition, results show that the M-NN preconditioner has lower performance than the surrogate-based preconditioning approach. The main mechanisms controlling the efficiency of the PC-NN preconditioner are the local KL truncation error, the PC truncation error, and the PC projection error. The KL truncation error is controlled by the number of local random variables and by the apparent correlation length over the subdomain. The PC truncation and the projection errors are controlled by the sparse grid level. Concerning the stochastic discretization, the results demonstrated that using smaller subdomains allows to reduce the number of small random variables, while the sparse grid level must remain fixed as the partition is refined. In addition, the number of iterations of the PC-NN preconditioner is scalable with the number of subdomains when associated to the Nicolaides coarse space projection.

The Nicolaides coarse space is essential for the scalability of the PPCG iterative scheme. However, for more complex models, the combination of parallel preconditioners with the Nicolaides coarse space may not produce a scalable method. The GenEO coarse space is an extension of the Nicolaides coarse space that can improve its scalability and convergence rate in general. Contrary to the Nicolaides coarse space, the construction of the GenEO coarse space requires solving a set of generalized eigenvalue problems for each sample, increasing the set-up cost for each sample. The final part of this chapter explored a less expensive alternative to the construction of the GenEO coarse space. The proposed approach uses the M-GenEO coarse space, corresponding to the local subspaces associated with the median field. By re-using these local subspaces, no eigenvalue problem needs to be solved, and the cost is significantly reduced compared with the original approach. Numerical experiments demonstrated that, in general, the M-GenEO method requires a larger dimensional space to achieve the same performance of the GenEO method. The differences between the two methods tend to reduce as the field's complexity reduces, and the partition of the domain is refined.

5.6.2 Prospective Work

This chapter raises several questions that motivate future work.

5.6.2.1 Parallel implementation

The performance of the PPCG methods was only assessed from the number of iterations needed to converge, based on trivial parallelism of the proposed algorithms. Yet, a full and detailed investigation of the parallel efficiency is missing but highly desirable to completely appreciate the value of the proposed algorithm.

Some parallel experiments have been initiated at Cerfacs, but it remains to complete the parallelization of some procedures and optimize the framework to handle large numbers of processors. When these ongoing efforts will be determined, a complete characterization of the scalability and gains brought by the M-NN, PC-NN, and M-GenEO methods will be possible. At the moment, preliminary results show encouranging CPU time reductions and scalability on moderate size problems ($\sim 10^5$ degrees of freedom), using few hundreds of processors.

5.6.2.2 Coarse-space construction

Although the problems considered in the thesis do not fully benefit from advanced coarse space constructions, we believe this topic to be an important one. Future works will have to focus on problems that challenge the scalability of the PC-NN preconditioner combined with the Nicolaides coarse space. For such problems, the median-based GenEO coarse space approach proposed could be limited. The question will then be how to construct an efficient approximation of the local subspaces of a given sample. As discussed previously, the difficulty arises from the dependences on the coefficient field, which are not restricted to the subdomains but also involve their neighbouring subdomains. A possible way to mitigate this complexity would be to account for local random variables over the subdomains while approximating the coefficient over the neighbouring subdomains with less random variables (account for spatial correlations) or even the median field.

5.6.2.3 Projection operator

If an effective approximation of the local subspaces is critical, the construction of the projection operator for each sample also involves a non-negligible cost. As we have discussed before, the computation of $[\mathrm{M}]$ can be reduced to local operations, namely, multiplications of $[\mathrm{S}^{(d)}]$ by matrices $[\mathrm{W}^{(d)}]$. This particular structure of $[\mathrm{M}]$ suggests that instead of building surrogates of the local subspace (i.e., for the $[\mathrm{W}^{(d)}]$), it would be more efficient to directly derive surrogates of the local products $[\mathrm{S}^{(d)}][\mathrm{W}^{(d)}]$. These products have similar local dependences on neighbouring subdomains than for the local subspace, and, as such, could rely on similar dimensionality reduction strategies. In any case, without a careful treatment, approximations of the products $[\mathrm{S}^{(d)}][\mathrm{W}^{(d)}]$ in the definition of (5.11) of $[\mathrm{M}]$ could lead to projection operators $[\widetilde{\Pi}]$ that are not $[\mathrm{S}]$-orthogonal for every sample, with severe degradation of the PPCG method performance as a result. Finally, all these aspects (parallelism, subspace construction, surrogate for the projection operator) will have to be combined and tested on larger (3 dimensional) and more challenging problems to fully appreciate the advantages of the surrogate BDD method.

Chapter 6

Conclusions and Perspectives

This thesis presented new Domain Decomposition (DD) approaches that are characterized by surrogate-based preconditioners specifically designed for the sampling of stochastic elliptic equations with highly variable and low correlated coefficient fields. The generation of each solution sample amounts to solve a reduced Finite Element (FE) system for the solution values on the subdomain's interfaces. We proposed three distinct methods to generate each sample (solve each reduced system), which are discussed separately in dedicated chapters.

In Chapter 3 we considered an overlapping DD setting and generate each solution sample using the additive Schwarz method (SM), which requires preconditioning. The preconditioning approach proposed amounts to construct Polynomial Chaos (PC) expansion of the reduced FE matrix based on a KL expansion of the field. This PC expansion is evaluated for each sample to generate a preconditioner for the SM scheme adapted to each sample. We called the resulting method the PC-based Preconditioned SM (PSM) method.

In Chapter 4, we considered a non-overlapping setting, in which case, the resulting reduced FE matrix is known as the Schur complement matrix. Each solution sample is obtained by solving the Schur system by a Preconditioned Conjugate Gradient (PCG) method. We proposed a stochastic Schur-based preconditioner for the PCG method. The proposed preconditioning approach amounts to construct a surrogate of the Schur complement matrix, based on independent (local) PC-based surrogates of the Schur's subdomain components depending on KL

expansions of the field over $\Omega^{(d)}$. Thanks to local parametrizations of the field, these local surrogates are evaluated according to each sample independently from one another. The resulting realizations are assembled together to generate a symmetric and positive definite (SPD) preconditioner for the PCG scheme.

Finally, in Chapter 5, we presented the PC-based Neumann-Neumann (PC-NN) preconditioner for the Balancing Domain Decomposition (BDD) method [40]. Exploiting local KL expansions of the field, the PC-NN preconditioner is based on local PC-based surrogates of the NN maps that characterize the original BDD method. The PC-NN preconditioner was combined with the Nicolaides [90] coarse space, yielding a scalable method whose performance in terms of iterations is close the the original approach. Also, we explored an extension of the Nicolaides coarse space, the so-called GenEO coarse space [89], that is used in context where the Nicolaides coarse space is not scalable. In particular, we proposed a new GenEO-type construction through median-based local subspaces, called Median-GenEO (M-GenEO), that significantly reduces the set-up cost of the original GenEO construction.

In this chapter we present a summary of the major achievements of this thesis, in Section 6.1, and limitations that motivate future works, in Section 6.2

6.1 Achievements

The approaches developed in this thesis are designed so that the accuracy of the surrogates does not impact the accuracy of the resulting solution samples. Indeed, the surrogates are only used to construct a preconditioner for solving the original problem. As a result, the accuracy of each solution is not dictated by accuracy of the surrogates involved, contrary to other approaches proposed in [32, 86, 63]. Other major achievements of the thesis are recalled below.

6.1.1 Advanced Surrogate-based Preconditioners Adapted to the Coefficient Samples

The introductory Chapter 2 identifies the preconditioning step as an essential ingredient of DD methods. State-of-the-art DD methods are very effective; however, the cost of the precondi-

tioning step represents a significant computational burden. Chapter 2 suggested DD methods based on sample-independent preconditioning (based on statistics of the field such as the median) as a possible alternative to DD methods having expensive sample-dependent preconditioners. However, sample-independent strategies can be significantly less effective compared to approaches adapted to each sample. The surrogate-based preconditioners proposed in this thesis are adapted to each sample, providing much faster convergence rates than sample-independent approaches. Several numerical experiments demonstrated the relevance of this strategy.

In Chapter 3, we developed the surrogate-based approach to accelerate the SM. Results show that the high acceleration rates that characterize the median-based PSM (preconditioner is the reduced FE matrix based on the median field) significantly deteriorates as the complexity (stochastic dimensionality) of the field increases. Opposite to that, the PC-based preconditioner remains effective provided that the number of random variables involved in the parametrization of the field is sufficiently large. Numerical experiments report that the typical average acceleration of the median-based PSM w.r.t. the non-preconditioned SM is about 3 orders of magnitude. However, using a truncated KL representation with $10 - 15$ random variables, the PC-based PSM produces average acceleration rates up to 5 orders of magnitude.

In Chapter 4, we developed a surrogate-based preconditioner to accelerate the resolution of Schur systems with CG-type algorithms. The surrogate-based preconditioner is defined according to local PC-based surrogates of the Schur matrix's subdomain components, exploiting local parameterizations of the coefficient field. The local surrogates are defined according to a factorized form that guarantees all preconditioners generated are SPD, providing robustness to the approach. For a stochastic coefficient field with high variance and low correlation, the proposed surrogate-based preconditioner allowed us to obtain the solution up to 7 times faster in terms of average iterations compared with the median-based preconditioner (Schur complement matrix based on the median field).

In chapter Chapter 5 we proposed variants of the BDD method specifically designed to solve multiple sampled problems. The BDD method is based on the NN preconditioner, whose application is totally based on local operations. We proposed two variations of the original NN preconditioner, one based on the median-based NN maps (the M-NN preconditioner) and an-

other based on local PC-based surrogates of the NN maps (the PC-NN preconditioner). Similar to the local surrogate proposed in the previous chapter, the local PC-based surrogate of the NN maps are defined according to a factorized form that is crucial to guarantee stability and higher convergence rates. The performances of the PC-NN preconditioner and the M-NN preconditioner have been compared against the standard NN preconditioner. Several mechanisms can boost the performance of the PC-NN preconditioner to approach the standard NN performance, including the number of random variables, the sparse grid level, and the number of subdomains. On the contrary, the performance of the M-NN preconditioner remains significantly below the NN reference, and little can be done to improve its convergence rate. In addition, the PC-NN preconditioner shares the same scalability properties as the NN preconditioner, which is not the case for the median-based approach.

6.1.2 Preconditioners Constructed at Negligible Cost per Sample

The generation of solution samples using DD methods amounts to perform several set-up operations for each sample that require significant computational resources. As a result, the number of samples generated can be limited.

The preconditioning operations associated with the DD approach presented in this thesis are cheaper, and split the cost of the surrogate-based local approach between the pre-processing stage and the sampling stage. Due to the efficient evaluation of each local PC expansions, the generation of the surrogate-based preconditioners is cheaper than standard preconditioners. In addition, the application of each of these surrogate-based preconditioners reduces to local matrix-vector multiplications, which weighs very little in the total cost of each iteration. In these conditions, the cost of the surrogate-based approach is essentially the cost of the pre-processing stage. The pre-processing stage is based on operations amenable to parallel implementations. The local parametrizations provide very low computational complexity associated with each subdomain (computational unit). Subsequently, the cost of the pre-processing stage is factorized over the samples. Hence, in the context of many samples, the surrogate-based approaches presented in this thesis have a low cost per sample, which represents a significant advantage over the straightforward application of other DD methods with full set-up.

6.1.3 Preconditioning Approach for Parallel Implementations

Some of the algorithms presented in this thesis were designed specifically with parallel implementations in mind. Chapter 4 introduces a global surrogate-based preconditioner for the Schur problem that is generated from local surrogates over each subdomain. The construction of these surrogates amounts to compute local PC expansions. Thanks to the local parametrizations of the field, these PC expansions are independent of each other and can be computed in parallel. If the subdomains are small and use only a few random variables, the approach induces a very low complexity for the surrogates. The resulting construction is therefore suitable for parallel implementations.

Chapter 5 uses surrogate-based preconditioning to design a DD method that is scalable with the number of subdomains and suitable for parallel implementations. Indeed, the workload of the PC-NN's pre-processing stage is identical to the pre-processing stage of the Schur-based preconditioner presented in the previous chapter. Therefore, it is also adapted to parallel implementations. At the sampling stage, many operations are also amenable to be performed in parallel. In particular, the preconditioning step is completely relying on local operations amenable to parallel execution, as oppose to the approach of the previous chapter. Indeed, the generation of the surrogate-based pseudo-inverses is independent among subdomains, and the application of the resulting surrogate-based pseudo-inverses at each iteration is also totally local.

The reach scalability, NN-type preconditioners need the introduction of coarse spaces and projection operators. We verified that the fully parallel M-GenEO approach was a possible alternative to the standard GenEO method, which involves more communication between subdomains.

6.2 Limitations

This thesis presents several perspectives that can motivate future works. This section draws the major guidelines to improve the results already obtained.

6.2.1 Parallel Implementations

The performance assessment of the methods presented in this thesis was limited to iterations count. To fully support the claim of parallel efficiency it will be necessary to rely on parallel implementations and report execution times. An effort towards parallel implementation has been engaged but a fair comparison of the parallel efficiency requires optimized implementation which takes time, in particular to target massively parallel frameworks.

When parallel implementations of the surrogate-based BDD method will be available, it will be interesting to compare it with existing approaches (for example [117, 89]), especially on larger problems. In addition, a parallel comparison of the methods of Chapter 4 and Chapter 5 will be necessary to determine which is more efficient depending on the problem size. Indeed, for moderately large problems, the effectiveness of the global preconditioning approach of Chapter 4 can potentially compensate for the cost associated with the global operations involved.

Parallel implementation of the method of Chapter 5 would also enable dealing with much larger problems, in particular, in three dimensions. Higher dimensional problems will raise some difficulties such as more involved parametrizations of the local fields that may require alternative surrogate approaches (for instance, low-rank approximations).

6.2.2 Adaptation of the Methods to Discontinuous Fields

Another extension concerns the adaptation of the proposed methods to elliptic problems with discontinuous fields (e.g. layered media). In these situations, DD methods and standard preconditioning strategies can be severely challenging [52, 43, 89]. Then, it would be interesting to test our approaches on problems with stochastic coefficient fields with localized discontinuities.

One would expect the global character of the Schur-based preconditioner used in Chapter 4 to effectively handle these situations. However, once again, the size of the problem continues to be a serious limitation, for the Schur-based preconditioner. Concerning the surrogate-based BDD method proposed, the question comes down to the type of coarse space used. The Nicolaides coarse space presents scalability issues for discontinuous fields [52, 43, 89], which demand a richer (and often more expensive) coarse space. The alternative to the Nicolaides coarse space considered in Chapter 5 was the GenEO coarse space built on local subspaces

containing the singular modes (Nicolaides subspace) and the "most singular" modes of the NN maps. The construction of the GenEO local subspaces involves the resolution of a set of generalised eigenvalue problems. The M-GenEO coarse space was introduced to bypass the resolution of these problems, and several numerical experiments for continuous fields empirically show that the median-based coarse space can provide acceleration rates comparable to the original approach. However, it is unclear how the extra part of the M-GenEO will help the convergence in the case of discontinuous fields, and the comparison of the GenEO and the M-GenEO coarse spaces should be extended to discontinuous fields.

Finally, on a longer perspective, the extension of the proposed algorithm to problem with coefficient fields supporting discontinuities at uncertain locations would be interesting. In particular, these situations will challenge the local parametrizations.

6.2.3 Generalization of Surrogate-based Preconditioning Approaches

The surrogate-based preconditioning strategies presented in this thesis are specifically designed to solve multiple sampled problems. Given the performance of the method, it would be interesting to combine surrogate-based preconditioners with advanced sampling strategies such as MLMC [27, 8, 61, 69, 111], which could require level-dependent partitions and stochastic discretization parameters.

The surrogate-based preconditioners presented in this thesis could also be applied to other DD formulations, such as the FETI method [40]. Another route is to extend the surrogate-based preconditioning approach to non-scalar elliptic problems motivated by a wide range of applications. Extension to the linear elasticity problems should be straightforward. Similar DD methods for parabolic problems should be also amenable to surrogate-based preconditioners. In contrast, DD methods for hyperbolic and non-linear problems is general constitute a much more challenging extension of surrogate-based preconditioners.

Appendix A

The additive Schwarz Method: Matrix Form

An iteration of the SM can be seen as the application of a mapping $\mathscr{S} : \mathbb{R}^{N_\Gamma} \to \mathbb{R}^{N_\Gamma}$, such that the SM aims at finding the fixed-point solution

$$\mathscr{S}(\mathrm{u}_\Gamma) = \mathrm{u}_\Gamma. \tag{A.1}$$

Consider an initial guess of the boundary values u_Γ^0, satisfying the global boundary values (i.e., $\mathrm{u}_\vdash^{(d=1),k=0} \equiv \mathrm{u}_0$ and $\mathrm{u}_\dashv^{d=D,k=0} \equiv u_1$), to construct a sequence u_Γ^k converging to u_Γ. In the additive Schwarz method, the updated vector u_Γ^{k+1} is defined by solving the local problems (3.10) using the current estimate of boundary values in u_Γ^k. To this end, in Equation (3.10) we denoted $\mathcal{P}_k^{(d)} := \mathcal{P}^{(d)}(\mathrm{u}_{\Gamma^{(d)}}^k, f, \kappa)$ the local problems for the k-th iterate of the boundary values, and $w_k^{(d)}(x)$ their solutions. Following the structure of the decomposition of the domain shown in Figure 3.1, the updated values are defined by

$$\mathrm{u}_\vdash^{(d),k+1} = \begin{cases} w_k^{(d-1)}(x_\vdash^{(d)}), & 1 < d \leq D \\ \mathrm{u}_0, & d = 1 \end{cases} \quad \text{and} \quad \mathrm{u}_\dashv^{(d),k+1} = \begin{cases} w_k^{(d+1)}(x_\dashv^{(d)}), & 1 \leq d < D \\ \mathrm{u}_1, & d = D \end{cases}. \tag{A.2}$$

Let $w_f^{(d)}(x)$ be the solution corresponding to local problem $\mathcal{P}^{(d)}((0,0), f, \kappa)$, i.e. the lo-

cal problem with *homogeneous boundary conditions*. Similarly, define the elementary solutions $w_{\vdash}^{(d)}$ and $w_{\dashv}^{(d)}$ of the *homogeneous* local problems with elementary boundary conditions $\mathcal{P}^{(d)}\left((1,0),0,\kappa\right)$ and $\mathcal{P}^{(d)}((0,1),0,\kappa)$, respectively. Each harmonic problem is the influence of each subdomain's boundary into the global solution, called as *influence problem*. A sketch of these functions is drawn in Figure A.1. Exploiting the linearity and superposition principle of the

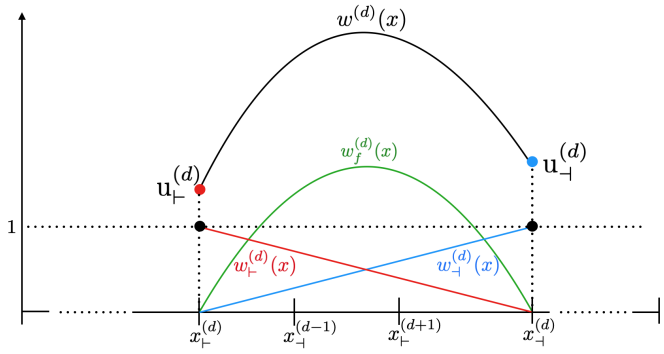


Figure A.1: Sketch of the local solutions and their corresponding interface decomposition.

elliptic problem, the local solution of $\mathcal{P}^{(d)}(\mathrm{u}_{\Gamma^{(d)}},f,\kappa)$ can be expressed as

$$w^{(d)}(x)=w_{\vdash}^{(d)}(x)\mathrm{u}_{\vdash}^{(d)}+w_{\dashv}^{(d)}(x)\mathrm{u}_{\dashv}^{(d)}+w_f^{(d)}(x). \tag{A.3}$$

The expressions in (A.2) for the update of the boundary values can be consequently rewritten in

$$\mathrm{u}_{\vdash}^{(d),k+1}=\begin{cases} w_{\vdash}^{(d-1),k}\left(x_{\vdash}^{(d)}\right)\mathrm{u}_{\vdash}^{(d-1),k}+w_{\dashv}^{(d-1),k}\left(x_{\vdash}^{(d)}\right)\mathrm{u}_{\dashv}^{(d-1),k}+w_f^{(d-1),k}\left(x_{\vdash}^{(d)}\right), & 1<d\leq D \\ \mathrm{u}_0, & d=1 \end{cases}, \tag{A.4}$$

and

$$\mathrm{u}_{\dashv}^{(d),k+1}=\begin{cases} w_{\vdash}^{(d+1),k}\left(x_{\dashv}^{(d)}\right)\mathrm{u}_{\vdash}^{(d+1),k}+w_{\dashv}^{(d+1),k}\left(x_{\dashv}^{(d)}\right)\mathrm{u}_{\dashv}^{(d+1),k}+w_f^{(d+1),k}\left(x_{\dashv}^{(d)}\right), & 1\leq d<D \\ \mathrm{u}_1, & d=D \end{cases}. \tag{A.5}$$

The fixed point iteration can be formally recast as

$$\mathrm{u}_\Gamma^{k+1} = [\mathrm{L_S}]\mathrm{u}_\Gamma^k + \mathrm{b_S}, \tag{A.6}$$

where the entries of the matrix $[\mathrm{L_S}]$ correspond to the local solutions of the influence problems (from $w_{\vdash}^{(d)}$ and $w_{\dashv}^{(d)}$), while the entries of the vector $\mathrm{b_S}$ are made of the local solutions for the homogeneous boundary values (from $w_f^{(d)}$). However, one should keep in mind that the matrix $[\mathrm{L_S}]$ and vector $\mathrm{b_S}$ are not formed in practice, and Algorithm 1 is instead employed to generate the successive iterates.

Appendix B

Global KL-expansion

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

$$C(x, x') \doteq \mathbb{E}\left[\boldsymbol{g}(x, \theta)\boldsymbol{g}(x', \theta)\right]. \tag{B.1}$$

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

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

where the eigenpairs $(\lambda_i, \phi_i(x))$ are the solutions of the eigenvalue problem [65, 79, 70]

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

The eigenvalues satisfying (B.3) are non-negative, and the eigenfunctions are normalized according to the spatial norm introduced in Section 4.2.1. The random variables $\boldsymbol{\eta}_i(\theta)$ are given by

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

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

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

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

$$\boldsymbol{g}(x,\theta) \approx \hat{g}(x,\theta) \doteq \sum_{i=1}^{N_{KL}} \sqrt{\lambda_i}\phi_i(x)\boldsymbol{\eta}_i(\theta). \tag{B.5}$$

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

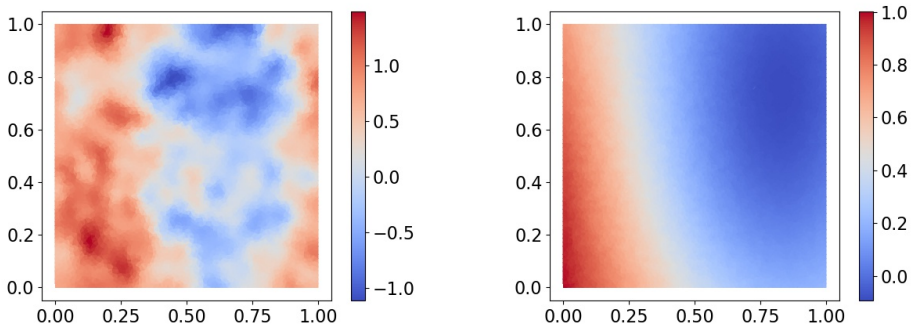
$$\mathrm{E}^2_{N_{KL}} = \mathbb{E}\left[\|\boldsymbol{g} - \hat{g}\|^2\right] = \sum_{i,j>N_{KL}} \sqrt{\lambda_i\lambda_j}\langle\phi_i,\phi_j\rangle_\Omega \mathbb{E}\left[\boldsymbol{\eta}_i\boldsymbol{\eta}_j\right] = \sum_{i>N_{KL}} \lambda_i,$$

from which the convergence follows because $\sum_i \lambda_i < \infty$ for a second-order field. In figure B.2 we illustrate the decay of $\sqrt{\lambda_i}$ for different types of covariance functions and correlation lengths.

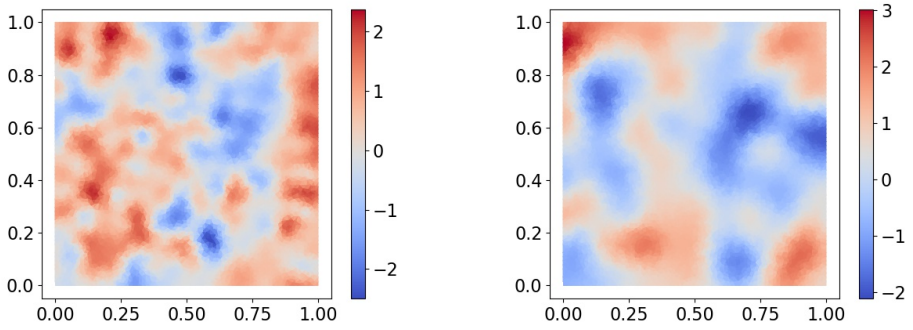
For physically relevant fields, the frequency content of the eigenfunctions $\phi_i(x)$ increases with the mode index i. The first modes account for the large scale deviations, while the higher order modes represent short scale details of the field. This observation explains why rougher fields typically 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 B.1 typical realizations of a Gaussian field in the unit square domain with covariance in (4.8), with parameter $\gamma = 1$ (left) and $\gamma = 2$ (right), and correlation length $\ell_c = 1$ (top) and $\ell_c = 0.1$ (bottom), and $\sigma^2 = 1$.

Figure B.2 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 (4.8) is much more demanding for $\gamma = 1$ than for $\gamma = 2$.

Figure (B.3) 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 (4.8), with $\sigma^2 = 1$, $\ell_c = 0.5$, and $\gamma = 1$ (left) and $\gamma = 2$ (right). Plots correspond to using $N_{KL} = 5$, 20 and 60 (from top to bottom) modes in the approximation. The normalized approximation error

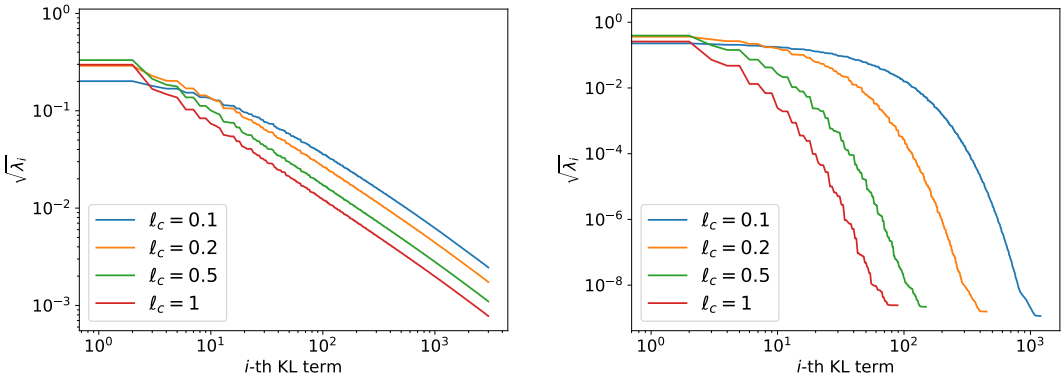


(a) $\ell_c = 1$ (b) $\ell_c = 1$



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

Figure B.1: Sample of the field $\boldsymbol{g}$ for the covariance (4.8) with parameter $\gamma = 1$ (left), $\gamma = 2$ (right), $\ell_c = 1$ (top), $\ell_c = 0.1$ (bottom) and $\sigma^2 = 1$



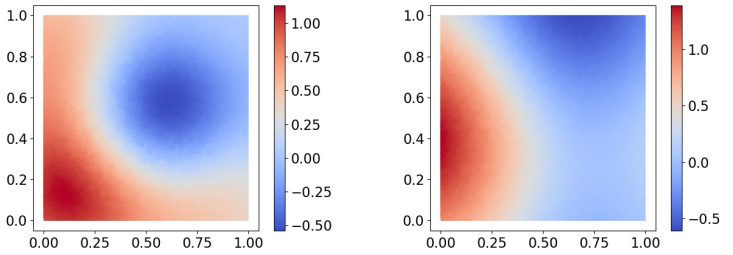
(a) Exponential convariance $(\gamma = 1)$ (b) Gaussian convariance $(\gamma = 2)$

Figure B.2: Spectral decay of the KL expansion for the covariance function C in (4.8) with $\gamma = 1$ (left) and $\gamma = 2$ (right) and several correlation lengths ℓ_c as indicated.

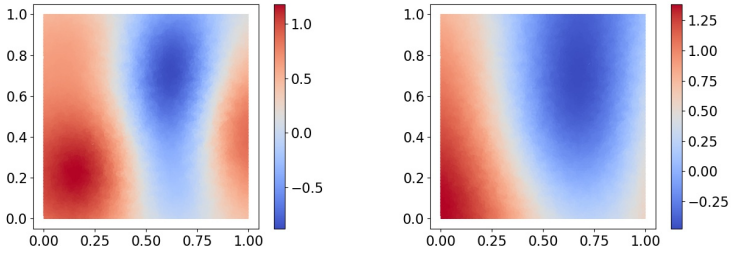
on this particular realization,

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

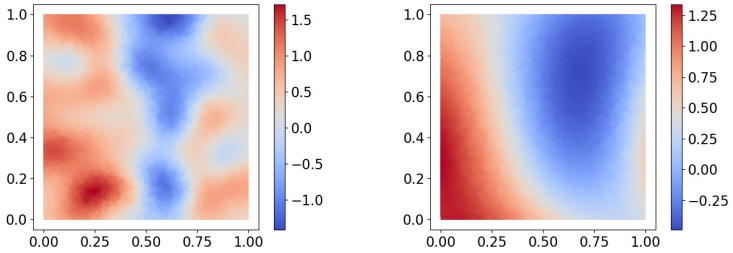
is also indicated. We see that for $\gamma = 1$, the convergence with N_{KL} is slow, reaching a normalized error of roughly 30% for $N_{KL} = 60$. In contrast, the approximation from the covariance with $\gamma = 2$ quickly converges with a normalized error lower than 10% with only $N_{KL} = 10$ modes.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



(a) $N_{KL} = 5$, $\mathrm{e_{KL}} \approx 0.73$. (b) $N_{KL} = 5$, $\mathrm{e_{KL}} \approx 0.38$.



(c) $N_{KL} = 10$, $\mathrm{e_{KL}} \approx 0.57$. (d) $N_{KL} = 10$, $\mathrm{e_{KL}} \approx 0.07$.



(e) $N_{KL} = 60$, $\mathrm{e_{KL}} \approx 0.34$. (f) $N_{KL} = 60$, $\mathrm{e_{KL}} \approx 6 \times 10^{-6}$.

Figure B.3: 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 $N_{KL} = 5$, 10 and 60 (from top to bottom). The corresponding $L^2(\Omega)$ errors $\mathrm{e_{KL}}$ are also indicated.

Bibliography

[1] Simon Abraham, Mehrdad Raisee, Ghader Ghorbaniasl, Francesco Contino, and Chris Lacor. A robust and efficient stepwise regression method for building sparse polynomial chaos expansions. *Journal of Computational Physics*, 332:461–474, 2017.

[2] Christophe Audouze and Prasanth B. Nair. Sparse approximate solutions to stochastic Galerkin equations. *Comptes Rendus Mathematique*, 357(6):561–570, 2019.

[3] Ivo Babuska, Raúl Tempone, and Georgios E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.

[4] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[5] Santiago Badia, Alberto F. Martín, and Javier Príncipe. Implementation and scalability analysis of Balancing Domain Decomposition methods. *Archives of Computational Methods in Engineering*, 2003.

[6] Santiago Badia, Alberto F. Martín, and Javier Príncipe. Enhanced balancing neumann-neumann preconditioning in computational fluid and solid mechanics. *International Journal for Numerical Methods in Engineering*, 2013.

[7] Feng Bai and Yi Wang. Reduced-Order Modeling Based on Hybrid Snapshot Simulation. *International Journal of Computational Methods*, 18(01):2050029, feb 2021.

[8] Andrea Barth, Christoph Schwab, and Nathaniel Zollinger. Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, 119(1):123–161, 2011.

[9] Joakim Beck, Raul Tempone, Fabio Nobile, and Lorenzo Tamellini. On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. *Mathematical Models and Methods in Applied Sciences*, 22(09):1250023, 2012.

[10] Joakim Beck, Fabio Nobile, Lorenzo Tamellini, and Raúl Tempone. Convergence of quasi-optimal stochastic galerkin methods for a class of PDES with random coefficients. *Computers & Mathematics with Applications*, 67(4):732 – 751, 2014. High-order Finite Element Approximation for Partial Differential Equations.

[11] Peter Benner, Albert Cohen, Mario Ohlberger, and Karen Willcox, editors. *Model reduction and approximation: theory and algorithms*. Number 15 in Computational science and engineering. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2017.

[12] Alex Bespalov, Daniel Loghin, and Rawin Youngnoi. Truncation preconditioners for stochastic Galerkin finite element discretizations. *arXiv:2006.06428 [cs, math]*, feb 2021. arXiv: 2006.06428.

[13] Géraud Blatman and Bruno Sudret. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliability Engineering & System Safety*, 95 (11):1216–1229, 2010.

[14] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345 – 2367, 2011.

[15] Jean-François Bourgat, Roland Glowinski, Patrick Le Tallec, and Marina Vidrascu. Variational formulation and algorithm for trace operation in domain decomposition calculations, 1989.

[16] Robert Bridson and Chen Greif. A multipreconditioned conjugate gradient algorithm. *SIAM Journal on Matrix Analysis and Applications*, 27(4):1056–1068, 2006.

[17] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[18] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *The Annals of Mathematics*, 48(2):385–392, 1947.

[19] F. Chaouqui, G. Ciaramella, M. J. Gander, and T. Vanzan. On the scalability of classical one-level domain-decomposition methods. *Vietnam Journal of Mathematics*, 46(4):1053–1088, 2018.

[20] Julia Charrier. *Analyse numérique d'équations aux derivées partielles à coefficients aléatoires applications à l'hydrogéologie*. Theses, ENS Cachan - Bretagne, July 2011.

[21] Julia Charrier. Strong and weak error estimates for elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, 50(1):216–246, 2012.

[22] Yi Chen, John Jakeman, Claude Gittelson, and Dongbin Xiu. Local polynomial chaos expansion for linear differential equations with high dimensional random inputs. *SIAM Journal on Scientific Computing*, 37(1):A79–A102, 2015.

[23] Mathilde Chevreuil, Régis Lebrun, Anthony Nouy, and Prashant Rai. A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):897–921, jan 2015. arXiv: 1305.0030.

[24] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric pdes. *Foundations of Computational Mathematics*, 14(4):601–633, 2014.

[25] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2002.

[26] K. A. Cliffe, I. G. Graham, R. Scheichl, and L. Stals. Parallel computation of flow in heterogeneous media modelled by mixed finite elements. *Journal of Computational Physics*, 164(2), 2000.

[27] K Andrew Cliffe, Mike B Giles, Robert Scheichl, and Aretha L Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3, 2011.

[28] Albert Cohen, Ronald DeVore, and Christoph Schwab. Convergence rates of best n-term galerkin approximations for a class of elliptic spdes. *Foundations of Computational Mathematics*, 10(6):615–646, 2010.

[29] Patrick R. Conrad and Youssef M. Marzouk. Adaptive smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing*, 35(6):A2643–A2670, 2013.

[30] Paul G. Constantine, Michael S. Eldred, and Eric T. Phipps. Sparse pseudospectral approximation method. *Computer Methods in Applied Mechanics and Engineering*, 229–232:1 – 12, 2012.

[31] Andres A Contreras, Paul Mycek, Olivier P Le Maitre, Francesco Rizzi, Bert Debusschere, and Omar M Knio. Parallel Domain Decomposition Strategies for Stochastic Elliptic Equations. Part A: Local Karhunen–Loève Representations. *SIAM Journal on Scientific Computing*, 40(4):C520–C546, 2018.

[32] Andres A Contreras, Paul Mycek, Olivier P Le Maitre, Francesco Rizzi, Bert Debusschere, and Omar 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):C547–C580, 2018.

[33] L. Beirão da Veiga, C. Lovadina, and L. F. Pavarino. Positive definite balancing Neumann–Neumann preconditioners for nearly incompressible elasticity. *Numerische Mathematik*, 104(3):271–296, 2006.

[34] Manas K Deb, Ivo M Babuška, and J Tinsley Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.

[35] Manas K. Deb, Ivo M. Babuška, and J. Tinsley Oden. Solution of stochastic partial differ-

ential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.

[36] Bert J. Debusschere, Habib N. Najm, Alain Matta, Omar M. Knio, Roger G. Ghanem, and Olivier P. Le Maitre. Protein labeling reactions in electrochemical microchannel flow: Numerical simulation and uncertainty propagation. *Physics of Fluids*, 15(8):2238–2250, August 2003.

[37] Ajit Desai, Mohammad Khalil, Chris Pettit, Dominique Poirel, and Abhijit Sarkar. Scalable domain decomposition solvers for stochastic PDEs in high performance computing. *Computer Methods in Applied Mechanics and Engineering*, 335:194–222, June 2018.

[38] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, May 2013.

[39] Victorita Dolean, Frédéric Nataf, Robert Scheichl, and Nicole Spillane. Analysis of a Two-level Schwarz Method with Coarse Spaces Based on Local Dirichlet-to-Neumann Maps. *Computational Methods in Applied Mathematics*, 12(4):391–414, 2012.

[40] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. *An introduction to domain decomposition methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015. Algorithms, theory, and parallel implementation.

[41] Alireza Doostan, Roger G. Ghanem, and John Red-Horse. Stochastic model reduction for chaos representations. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40):3951–3966, August 2007.

[42] Alireza Doostan, AbdoulAhad Validi, and Gianluca Iaccarino. Non-intrusive low-rank separated approximation of high-dimensional stochastic models. *Computer Methods in Applied Mechanics and Engineering*, 263:42–55, 2013.

[43] Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *Mathematical Modelling and Numerical Analysis*, 2012.

[44] Yalchin Efendiev and Thomas Y. Hou. *Multiscale Finite Element Methods*. Springer New York, New York, NY, 2009.

[45] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical mathematics and scientific computation. Oxford University Press, Oxford, United Kingdom, second edition edition, 2014.

[46] O.G. Ernst, C.E. Powell, D.J. Silvester, and E. Ullmann. Efficient solvers for a linear stochastic galerkin mixed formulation of diffusion problems with random data. *SIAM Journal on Scientific Computing*, 31(2):1424–1447, 2009.

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

[48] Charbel Farhat and Francois-Xavier Roux. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM Journal on Scientific and Statistical Computing*, 13(1):379–396, 1992.

[49] Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall Pierson, and Daniel Rixen. FETI-DP: a dual-primal unified FETI method?part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7):1523–1544, mar 2001.

[50] X. Feng and C. Lorton. An efficient monte carlo interior penalty discontinuous galerkin method for elastic wave scattering in random media. *Computer Methods in Applied Mechanics and Engineering*, 315:141–168, 2017.

[51] Philipp Frauenfelder, Christoph Schwab, and Radu Alexandru Todor. Finite elements for elliptic problems with stochastic coefficients. *Computer methods in applied mechanics and engineering*, 194(2):205–228, 2005.

[52] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling & Simulation*, 8(4):1461–1483, 2010.

[53] Martin J Gander. Optimized schwarz methods. *SIAM Journal on Numerical Analysis*, 44 (2):699–731, 2006.

[54] Luis García Ramos, René Kehl, and Reinhard Nabben. Projections, Deflation, and Multigrid for Nonsymmetric Matrices. *SIAM Journal on Matrix Analysis and Applications*, 41 (1):83–105, January 2020.

[55] André Gaul, Martin H. Gutknecht, Jörg Liesen, and Reinhard Nabben. A Framework for Deflated and Augmented Krylov Subspace Methods. *SIAM Journal on Matrix Analysis and Applications*, 34(2):495–518, January 2013.

[56] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[57] Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[58] Paulo Goldfeld, Luca F. Pavarino, and Olof B. Widlund. Balancing neumann-neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. *Numerische Mathematik*, 95(2):283–324, 2003.

[59] I.G. Graham, F.Y. Kuo, D. Nuyens, R. Scheichl, and I.H. Sloan. Quasi-monte carlo methods for elliptic PDEs with random coefficients and applications. *Journal of Computational Physics*, 230(10):3668–3694, 2011.

[60] Ryadh Haferssas, Pierre Jolivet, and Frédéric Nataf. A robust coarse space for optimized Schwarz methods: SORAS-GenEO-2. *Comptes Rendus Mathematique*, 353(10):959 – 963, 2015.

[61] Abdul-Lateef Haji-Ali, Fabio Nobile, and Raúl Tempone. Multi-index monte carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2016.

[62] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge ; New York, 2nd ed edition, 2012.

[63] Thomas Y Hou, Qin Li, and Pengchuan Zhang. Exploring the locally low dimensional structure in solving random elliptic PDEs. *Multiscale Modeling & Simulation*, 15(2):661–695, 2017.

[64] Chao Jin, Xiao-Chuan Cai, and Congming Li. Parallel Domain Decomposition Methods for Stochastic Elliptic Equations. *SIAM Journal on Scientific Computing*, 29(5):2096–2114, jan 2007.

[65] Kari Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. Sana, 1947.

[66] O.M. Knio and O.P. Le Maître. Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dyn. Res.*, 38:616–640, 2006.

[67] Katerina Konakli and Bruno Sudret. Global sensitivity analysis using low-rank tensor approximations. *Reliability Engineering & System Safety*, 156:64–83, 2016.

[68] Katerina Konakli and Bruno Sudret. Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions. *Journal of Computational Physics*, 321:1144–1169, sep 2016.

[69] Frances Y. Kuo and Dirk Nuyens. Application of Quasi-Monte Carlo Methods to Elliptic PDEs with Random Diffusion Coefficients: A Survey of Analysis and Implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696, dec 2016.

[70] Olivier Le Maitre and Omar M Knio. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Science & Business Media, 2010.

[71] Olivier P. Le Maitre, Matthew T. Reagan, Habib N. Najm, Roger G. Ghanem, and Omar M. Knio. A stochastic projection method for fluid flow: Ii. random process. *Journal of Computational Physics*, 181(1):9 – 44, 2002.

[72] O.P. Le Maitre, O.M. Knio, B.J. Debusschere, H.N. Najm, and R.G. Ghanem. A multigrid solver for two-dimensional stochastic diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 192(41–42):4723 – 4744, 2003.

[73] Randall J. Leveque. *Finite Volume Methods for Nonlinear Elasticity in Heterogeneous Media*. John Wiley & Sons, Ltd, 2002.

[74] Randall J. LeVeque. *Finite difference methods for differential equations*. Society of Industrial and Applied Mathematics, 2006.

[75] Pierre-Louis Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, volume 1, page 42. SIAM, Philadelphia, 1990.

[76] Alexander Litvinenko, Dmitry Logashenko, Raul Tempone, Gabriel Wittum, and David Keyes. Solution of the 3D density-driven groundwater flow problem with uncertain porosity and permeability. *GEM - International Journal on Geomathematics*, 11(1):10, December 2020.

[77] J.S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Verlag, 2001.

[78] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer New York, New York, NY, 2004.

[79] Paul Lévy and Michel Loeve. *Processus stochastiques et mouvement brownien*. Gauthier-Villars Paris, 1965.

[80] Dingjiong Ma, Wai-ki Ching, and Zhiwen Zhang. Proper orthogonal decomposition method for multiscale elliptic PDEs with random coefficients. *Journal of Computational and Applied Mathematics*, 370:112635, 2020.

[81] Jan Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993.

[82] Jan Mandel and Clark R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications*, 10 (7):639–659, October 2003.

[83] Jan Mandel, Clark R. Dohrmann, and Radek Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Applied Numerical Mathematics*, 54(2):167–193, July 2005.

[84] Tarek P. A. Mathew. *Domain decomposition methods for the numerical solution of partial differential equations*. Number 61 in Lecture notes in computational science and engineering. Springer, Berlin, 2008.

[85] Gregory H. Miller and Elbridge Gerry Puckett. A Neumann–Neumann preconditioned iterative substructuring approach for computing solutions to Poisson's equation with prescribed jumps on an embedded boundary. *Journal of Computational Physics*, 235:683–700, 2013.

[86] Lin Mu and Guannan Zhang. A Domain Decomposition Model Reduction Method for Linear Convection-Diffusion Equations with Random Coefficients. *SIAM Journal on Scientific Computing*, 41(3):A1984–A2011, 2019.

[87] Abul Mukid Mohammad Mukaddes, Masao Ogino, Hiroshi Kanayama, and Ryuji Shioya. A Scalable Balancing Domain Decomposition Based Preconditioner for Large Scale Heat Transfer Problems. *JSME International Journal Series B*, 49(2):533–540, 2006.

[88] H.N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Ann. Rev. Fluid Mech.*, 41:35–52, 2009.

[89] Frédéric Nataf, Hua Xiang, Victorita Dolean, and Nicole Spillane. A Coarse Space Construction Based on Local Dirichlet-to-Neumann Maps. *SIAM Journal on Scientific Computing*, 33(4):1623–1642, jan 2011.

[90] R. A. Nicolaides. Deflation of Conjugate Gradients with Applications to Boundary Value Problems. *SIAM Journal on Numerical Analysis*, 24(2):355–365, 4 1987.

[91] 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.

[92] A. Nouy and O.P. Le Maître. Generalized spectral decomposition for stochastic nonlinear problems. *J. Comput. Phys.*, 228:202–235, 2009.

[93] Anthony 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):4718 – 4736, 2008.

[94] Anthony 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):4718–4736, October 2008.

[95] Anthony Nouy. Proper Generalized Decompositions and Separated Representations for the Numerical Solution of High Dimensional Stochastic Problems. *Archives of Computational Methods in Engineering*, 17(4):403–434, dec 2010.

[96] Anthony Nouy and Olivier P. Le Maı^tre. Generalized spectral decomposition for stochastic nonlinear problems. *Journal of Computational Physics*, 228(1):202–235, January 2009.

[97] Erich Novak and H. Woźniakowski. *Tractability of multivariate problems*. Number 6, 12, 19 in EMS tracts in mathematics. European Mathematical Society, Zürich, 2008. OCLC: ocn272566892.

[98] Dirk Nuyens. The construction of good lattice rules and polynomial lattice rules. In Peter Kritzer, Harald Niederreiter, Friedrich Pillichshammer, and Arne Winterhof, editors, *Uniform Distribution and Quasi-Monte Carlo Methods*. DE GRUYTER, Berlin, Boston, January 2014.

[99] Manolis Papadrakakis and Vissarion Papadopoulos. Robust and efficient methods for stochastic finite element analysis using monte carlo simulation. *Computer Methods in Applied Mechanics and Engineering*, 134(3-4):325–340, 1996.

[100] Michael L. Parks, Eric de Sturler, Greg Mackey, Duane D. Johnson, and Spandan Maiti. Recycling Krylov Subspaces for Sequences of Linear Systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, January 2006.

[101] L. Pavarino and O. Widlund. Balancing Neumann–Neumann methods for incompress-

ible stokes equations. *Communications on Pure and Applied Mathematics*, 55:302–335, 2001.

[102] M.F Pellissetti and 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): 607–616, aug 2000.

[103] Catherine E Powell and Howard C Elman. Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis*, 29(2):350–375, 2009.

[104] Srikara Pranesh and Debraj 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*, 311:457–475, 2016.

[105] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[106] Prashant Rai. *Sparse Low Rank Approximation of Multivariate Functions – Applications in Uncertainty Quantification*. Theses, Ecole Centrale de Nantes (ECN), Nov 2014.

[107] M. T. Reagan, H. N. Najm, R. G. Ghanem, and O. M. Knio. Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection. *Combustion and Flame*, 132(3):545–555, 2003.

[108] João F. Reis, Olivier P. Le Maitre, Pietro M. Congedo, and Paul Mycek. Stochastic preconditioners for domain decomposition methods. *UQOP2020 Conference Proceeding*, 2020.

[109] João F. Reis, Giulio Gori, Olivier P. Le Maitre, and Pietro M. Congedo. Introduction to spectral methods for uncertainty quantification. In Massimiliano Vasile, editor, *Optimization Under Uncertainty with Applications to Aerospace Engineering*, chapter 1, pages 1–34. Springer Nature Switzerland, Switzerland, 2021.

[110] João F. Reis, Olivier P. Le Maître, Pietro M. Congedo, and Paul Mycek. Stochastic preconditioning of domain decomposition methods for elliptic equations with random coefficients. *Computer Methods in Applied Mechanics and Engineering*, 381:113845, 2021.

[111] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. A multi-index quasi-monte carlo algorithm for lognormal diffusion problems. *SIAM Journal on Scientific Computing*, 39(5): S851–S872, 2017.

[112] Eveline Rosseel and Stefan Vandewalle. Iterative solvers for the stochastic finite element method. *SIAM Journal on Scientific Computing*, 32(1):372–397, 2010.

[113] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[114] Maher Salloum, Alen Alexanderian, Olivier P. Le Maitre, Habib N. Najm, and Omar M. Knio. Simplified CSP analysis of a stiff stochastic ODE system. *Computer Methods in Applied Mechanics and Engineering*, 217-220:121 – 138, 2012.

[115] Kirk M. Soodhalter, Eric Sturler, and Misha E. Kilmer. A survey of subspace recycling iterative methods. *GAMM-Mitteilungen*, 43(4), November 2020.

[116] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numerische Mathematik*, 126(4):741–770, apr 2014.

[117] Nicole Spillane. An adaptive multipreconditioned conjugate gradient algorithm. *SIAM journal on Scientific Computing*, 38(3):A1896–A1918, 2016.

[118] Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédéric Nataf, Clemens Pechstein, and Robert Scheichl. A robust two-level domain decomposition preconditioner for systems of PDEs. *Comptes Rendus Mathematique*, 349(23):1255–1259, 2011.

[119] Waad Subber and Sébastien Loisel. Schwarz preconditioners for stochastic elliptic PDEs. *Computer Methods in Applied Mechanics and Engineering*, 272:34–57, 2014.

[120] Waad Subber and Abhijit Sarkar. Domain decomposition method of stochastic PDEs: a two-level scalable preconditioner. In *Journal of Physics: Conference Series*, volume 341, page 012033. IOP Publishing, 2012.

[121] Waad Subber and Abhijit Sarkar. A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner. *Journal of Computational Physics*, 257:298–317, 2014.

[122] L. Tamellini, O.P. 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.

[123] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of Two-Level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods. *Journal of Scientific Computing*, 39(3):340–370, June 2009.

[124] Kunkun Tang, Pietro M. Congedo, and Rémi Abgrall. Adaptive surrogate modeling by ANOVA and sparse polynomial dimensional decomposition for global sensitivity analysis in fluid simulation. *Journal of Computational Physics*, 314:557–589, 2016.

[125] Aretha Leonore Teckentrup. *Multilevel Monte Carlo methods and uncertainty quantification*. Dissertation, Department of Mathematical Sciences, Unversity of Bath, June 2013.

[126] Ramakrishna Tipireddy, Panos Stinis, and Alexandre M Tartakovsky. Basis adaptation and domain decomposition for steady-state partial differential equations with random coefficients. *Journal of Computational Physics*, 351:203–215, 2017.

[127] Andrea Toselli and Olof B. Widlund. *Domain decomposition methods–algorithms and theory*. Number 34 in Springer series in computational mathematics. Springer, Berlin, 2005.

[128] Luca Traverso and Timothy Nigel Phillips. Efficient stochastic finite element methods for flow in heterogeneous porous media. Part 2: Random lognormal permeability. *International Journal for Numerical Methods in Fluids*, 92(11):1626–1652, November 2020.

[129] Lloyd N. Trefethen. *Spectral Methods in MATLAB*. Society for Industrial and Applied Mathematics, 2000.

[130] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

[131] Chong Wang, Zhiping Qiu, and Yaowen Yang. Uncertainty propagation of heat conduction problem with multiple random inputs. *International Journal of Heat and Mass Transfer*, 99: 95–101, 2016.

[132] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4): 897–936, October 1938.

[133] Justin Winokur, Daesang Kim, Fabrizio Bisetti, Olivier P Le Maitre, and Omar M Knio. Sparse pseudo spectral projection methods with directional adaptation for uncertainty quantification. *Journal of Scientific Computing*, 68(2):596–623, 2016.

[134] Dongbin Xiu. *Numerical methods for stochastic computations: A spectral method approach*. Princeton university press, 2010.

[135] Xiaojing Xu, Wenwen Fan, and Xiaoping Xie. Hybrid stress quadrilateral finite element approximation for stochastic plane elasticity equations: FINITE ELEMENT APPROXIMATION FOR STOCHASTIC ELASTICITY. *International Journal for Numerical Methods in Engineering*, 109(10):1418–1438, 2017.

[136] Qing-He Yao and Xin Pan. Large scale simulation of hydrogen dispersion by a stabilized balancing domain decomposition method. *Journal of Applied Mathematics*, 2014:1–10, 2014.

[137] Petr Zakharov, Oleg Iliev, Jan Mohring, and Nikolay Shegunov. Parallel Multilevel Monte Carlo Algorithms for Elliptic PDEs with Random Coefficients. In Ivan Lirkov and Svetozar Margenov, editors, *Large-Scale Scientific Computing*, volume 11958, pages 463–472. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[138] Elena Zampieri and Luca F. Pavarino. Implicit spectral element methods and Neumann–Neumann preconditioners for acoustic waves. *Computer Methods in Applied Mechanics and Engineering*, 195(19):2649–2673, 2006.

[139] Stefano Zampini. Balancing Neumann–Neumann methods for the cardiac Bidomain model. *Numerische Mathematik*, 123(2):363–393, 2013.

INSTITUT POLYTECHNIQUE DE PARIS

École doctorale de mathématiques Hadamard (EDMH)

Titre: Préconditionnement de Méthodes de Décomposition de Domaine pour les Équations Elliptiques Stochastiques.

Mots clés: Décomposition de Domaine; Quantification d'Incertitude; Méthode de Monte-Carlo; Preconditionneur Stochastique; Preconditionement de Method de Gradient Conjugué; Préconditionneur Neumann-Neumann.

Résumé:
Cette thèse présente une nouvelle méthode numérique pour générer efficacement des échantillons de la solution d'équations elliptiques stochastiques avec des coefficients aléatoires. Un accent particulier est mis sur les coefficients avec une variance élevée et des corrélation spatiales trés courtes. Ce travail concerne l'adaptation de certaines Méthodes de Décomposition de Domaine (DD) classiques à l'échantillonnage de problèmes stochastiques.
Les méthodes DD déterministes classiques reposent sur des approches itératives qui nécessitent des stratégies de préconditionnement capable de maintenir un taux de convergence élevé lorsque le nombre de sous-domaines augmente. Dans notre contexte stochastique, la détermination d'un préconditionneur classique adapté à chaque échantillon peut être coûteuse, et des stratégies alternatives peuvent être plus efficaces. Chaque échantillon revient à résoudre un système linéaire réduit pour les valeurs de solution aux interfaces des sous-domaines, selon une discrétisation par éléments finis. Ce système réduit est ensuite résolu par une méthode itérative. Cette thèse proposait trois contributions principales au préconditionnement efficace, en introduisant des métamodèles de 1) l'opérateur global réduit, 2) la contribution de chaque sous-domaine à l'opérateur global réduit, et 3) les préconditionneurs locaux (multi-préconditionnement).
La première contribution se concentre sur la méthode itérative de Schwarz et introduit un préconditionneur stochastique consistant en un métamodèle du système de Schwarz pour les valeurs inconnues sur la interface des sous-domains. La deuxième contribution étend l'idée précédente aux méthodes DD en construisant les métamodèles des composantes locales du complément de Schur. Finalment, la troisième contribution concerne un préconditionneur totalement local: le préconditionneur Neumann-Neumann à deux niveaux. Tout au long de chaque contribution, un grand nombre d'expériences numériques montrent l'efficacité de le préconditionnement basé sur métamodèles.

Title: Preconditioning of Domain Decomposition Methods for Stochastic Elliptic Equations.

Keywords: Domain Decomposition; Uncertainty Quantification; Sampling Method; Stochastic Preconditioner; Preconditioned Conjugate Gradient Method; Neumann-Neumann Preconditioner.

Abstract: This thesis presents a new numerical method to efficiently generate samples of the solution of stochastic elliptical equations with random coefficients. Particular emphasis is placed on coefficients with high variance and short correlation length. This work concerns the adaptation of some classical Domain Decomposition (DD) Methods to the sampling of stochastic problems. Classical deterministic DD methods are based on iterative approaches which require preconditioning strategies capable of maintaining a high rate of convergence when the number of subdomains increases. In our stochastic context, determining a classical preconditioner suitable for each sample can be expensive, and alternative strategies can be more efficient. Each sample amounts to solving a reduced linear system for the values of the solution at the interfaces of the subdomains, according to a finite element discretization. This reduced system is then solved by an iterative method. This thesis proposed three main contributions to efficient preconditioning, by introducing surrogates of 1) the reduced global operator, 2) the contribution of each subdomain to the reduced global operator, and 3) local preconditioners (multi-preconditioning).
The first contribution focuses on the iterative Schwarz method and introduces a stochastic preconditioner consisting of a surrogate of the Schwarz system for the unknown values on the interface of the subdomains. The second contribution extends the previous idea to non-overlapping DD methods by building surrogates of the local components of the Schur complement. Finally, the third contribution concerns a totally local preconditioner: the two-level Neumann-Neumann preconditioner. Throughout each contribution, a large number of numerical experiments show the effectiveness of surrogate-based preconditioning.

Institut Polytechnique de Paris

91120 Palaiseau, France

INSTITUT POLYTECHNIQUE DE PARIS