PRECONDITIONING TECHNIQUES FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH RANDOM DATA

by

RAWIN YOUNGNOI

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

Supervisor: Dr. Daniel Loghin
Co-Supervisor: Dr. Alex Bespalov

School of Mathematics
College of Engineering and Physical Sciences
The University of Birmingham
August 2020

Abstract

The stochastic Galerkin finite element method (SGFEM) is a well-established numerical method for approximating solutions of partial differential equations with parametric or random data. The advantage of SGFEM over traditional sampling methods is its convergence rate. However, this approach yields large-scale, often intractable systems of linear equations. Therefore, a powerful iterative solver equipped with a suitable preconditioner is required to approximate the solution for such linear systems.

In this thesis, we focus on designing preconditioners for stochastic Galerkin matrices that arise when solving the steady-state diffusion equation with random data. We consider two parametric representations of the diffusion coefficient: affine and non-affine.

For the case of affine-parametric diffusion coefficient, we present two preconditioners. Truncation preconditioners for affine-parametric diffusion problems form a new class of preconditioners that generalise the mean-based preconditioner by including additional information from the diffusion coefficient. Next, the domain decomposition technique for the parametric domain is introduced. This technique provides a framework for designing preconditioners which are capable of parallelism. We present a new concept of parametric mesh to represent the structure of the parametric space. Moreover, a so-called even-odd partitioning strategy for the parametric mesh is introduced. This strategy results in three versions of the even-odd preconditioners.

We provide spectral analyses of the preconditioned systems both for the truncation preconditioners and domain decomposition preconditioners, which confirm the optimality of the preconditioners with respect to discretisation parameters.

For the case of non-affine parametric diffusion coefficient, the truncation preconditioners and domain decomposition preconditioners are presented. They generalise the

idea of truncation preconditioners and domain decomposition preconditioners for affine-parametric coefficients by capturing the important terms and finding a structure which can utilise parallelism. We also design a preconditioner for log-transformed coefficients.

Finally, the performance of each preconditioner is illustrated by numerical experiments. We compare the efficiency (in terms of iteration counts and total complexity) of our purposed preconditioners with some existing preconditioners such as the mean-based preconditioner and the Kronecker product preconditioner.

Acknowledgments

I would like to express my sincere gratitude to two supervisors, Daniel Loghin and Alex Bespalov, for their invaluable suggestions and patience throughout my PhD study at the University of Birmingham. Without their knowledgeable and consistent guidance, my research would never go this far, and this thesis would never be successfully completed. Finishing this thesis is very important, but all the invaluable research experience at the University of Birmingham is more meaningful. I am very grateful to their attentiveness and support, particularly during the pandemic of COVID 19, which is a very difficult time for everyone.

I would like to extend my thanks to Thammasat University for providing me with the scholarship throughout this PhD study. This also includes the Office of Educational Affairs in London for supporting Thai students in many aspects.

I would like to express my great gratitude to my parents and Worasak Charungrattanapong for encouragement and supporting through thick and thin during my doctoral study and the pandemic. If this is one of my achievement, this is also theirs.

Last but not least, I would like to thank Jay Vicker, who listened to my rehearsals several times before my first talk. Thanks should also go to Pilantarat Allen, some colleagues at Thammasat University and ex-colleagues in London from Reuters Software Thailand Limited. I sincerely appreciate for always standing by me and supporting me during the difficult moments.

CONTENTS

CHAPTER 1

INTRODUCTION

Large-scale simulations are crucial for understanding complex physical processes and phenomena, e.g., in weather forecasting or subsurface hydrology. These simulations of these phenomena are usually associated with sophisticated mathematical models and employ numerical algorithms combined with powerful computing resources to approximate the solution of the model. In general, the mathematical models are usually represented via partial differential equations (PDEs) and an approximation of outcomes or quantities of interest is calculated via the solution of the PDEs to describe the underlying phenomena. The higher resolution mathematical models require more computational resources [8, 84, 7] and more efficient numerical methods such as parallel algorithms to reduce the discretisation error. In many cases, the inputs of the model, such as initial conditions or coefficient functions, are uncertain due to lack of knowledge or inherent variability. Uncertainties due to incomplete knowledge are referred to as being epistemic [113, 70], whereas the others are referred to as aleatoric uncertainties [113, 70]. The uncertain inputs are called random data and PDEs with random data are often called stochastic partial differential equations (SPDEs) which can be represented in a probabilistic framework. The randomness of the inputs adversely affects the accuracy of predictions.

Uncertainty quantification (UQ) deals with identifying, quantifying and reducing uncertainties in both computational simulations and real-world applications [61]. It helps to improve the accuracy in estimating the quantities of interest [84, 94, 53, 64]. Probability

theory provides a mathematical framework for UQ. To analyse uncertainties in the inputs, random variables represent the uncertainties in the input and the input with uncertainties are represented via stochastic processes or random fields with some statistical quantities of input data such as probability density functions (pdf), expected values [8, 94, 64]. In the SPDE setting, the quantities of interest are usually derived from the solution to an SPDE, for example, the spatial average or variance of the solution to an SPDE over the spatial domain. As a result, effective numerical methods are crucial in order to obtain the solution of an SPDE, in particular, they have to be adapted to the type of the problem under consideration [8, 94, 7].

In the past decades, many efficient numerical methods for SPDEs were introduced. These methods can be categorised into sampling methods and non-sampling methods [84, 94, 131, 133, 7]. Another approach to categorising the methods is based on how they are implemented [70]. Intrusive methods [8, 64, 62, 37, 7, 60] are the methods where the existing implementations cannot be used directly, e.g., the associated code needs a modification. On the other hand, numerical methods are called non-intrusive [102, 129, 9] if the existing code can be utilised without modification.

Monte Carlo methods (see [84, 131, 81, 105]) are some of the most popular sampling methods because they are simple and can be easily parallelised. They are non-intrusive methods. However, the main drawback of these methods is the rate of convergence, i.e., the statistical error, which is the error from the sampling, is proportional to $Q^{-1/2}$ where Q denotes the number of samples. Consequently, the approximate solution by Monte Carlo methods converges to the solution of the SPDE very slowly. As a result, they are not suitable for large-scale problems [136, 96]. Multilevel Monte Carlo methods ([11, 35, 66, 123]) perform the sampling on physical meshes with different mesh sizes. They result in faster convergence rate than Monte Carlo methods. More importantly, multilevel Monte Carlo methods require fewer samples on a finer grid. Therefore, multilevel Monte Carlo methods are more efficient for large-scale problems than Monte Carlo methods. Other numerical methods for SPDEs include quasi-Monte Carlo methods (see [38, 68, 67, 82, 30])

or multi-index Monte Carlo methods (see [72]).

Perturbation methods and Neumann series expansions are examples of non-sampling methods. While the former approximate the input and the solution of an SPDE by Taylor expansion [94, 64, 131, 6], the latter approximate the inverse of the stochastic stiffness matrix by the Neumann series expansion. Both these methods suit problems with small variations of uncertainties [94, 9, 64, 6, 76], due to properties of Taylor series and Neumann series.

Recently, stochastic spectral methods (SSMs) [64, 62, 7, 70], such as stochastic Galerkin methods or stochastic collocation methods, have gained considerable attention, particularly in mechanical engineering [57, 109, 65, 117], fluid dynamics [84, 80, 39, 92, 114, 135] and transport in porous medium [57, 58]. Furthermore, they are also successfully applied to many applications in chemistry [101, 93], biomedical engineering [56], acoustic scattering [47], deep excavations [31], earthquake engineering [3], civil engineering [1], medical imaging [86] and electromagnetics [18]. Each random variable representing the randomness in the model adds another dimension [63] to the problem. Furthermore, a vector of random variables induces a parametric space which is a Hilbert space associated with a random vector [84, 81, 59]. SSMs transform the stochastic problem into a coupled deterministic problem [2, 113]. The deterministic problem associated with the spatial domain can be approximated by any standard numerical method such as finite element methods [28, 27] or finite difference methods whereas a global polynomial approximation is employed in the parameter domain. The main disadvantage of SSMs is the computational effort required to solve the resulting large coupled system of linear equations.

Stochastic Galerkin methods (SGMs) are projection methods which are non-sampling and intrusive. These methods transform the SPDE to a variational formulation. To obtain an approximate solution, the parametric space and spatial space are represented by finite dimensional subspaces. The Galerkin methods approximate the solution of the SPDE by a function in the coupled finite dimensional subspaces [45, 131]. Moreover, the approximate solution by SGMs converges to the solution very fast [7, 84, 23] which is a

key advantage of these methods.

Stochastic collocation methods [9, 132] are sampling and non-intrusive methods. The sampling is performed at the collocation points selected in the parameter domain. It can be done in parallel which is one benefit of stochastic collocation methods. The approximate solution of the SPDE is then obtained using interpolation techniques. To be precise, the solution is approximated by an interpolant based on those samples. The convergence rates of stochastic collocation methods depend on the choice of polynomial basis [84, 131, 92, 10] and the computational cost may be higher than for intrusive methods [48, 130]. Nevertheless, they are non-intrusive methods that can achieve the fast convergence rate as intrusive methods such as SGMs. Note that solving the large coupled system of linear equations in the case of stochastic collocation methods may be avoided if Lagrange polynomials are employed. Thus, the solutions at the collocation points are the coefficients of the basis functions [45, 132, 48, 9].

Other techniques can be employed to reduce the computational cost or improve the accuracy of SSMs, such as stochastic reduced basis methods (SRBMs) and adaptive stochastic Galerkin finite element methods. The purpose of SRBMs [91, 107, 108, 90] is to reduce the computational cost of solving a large coupled linear system which arises from SSMs while maintaining the accuracy of the approximate solution. These methods represent the solution by a linear combination of basis vectors of preconditioned stochastic Krylov subspace. The Bubnov-Galerkin projection scheme is applied to obtain the coefficients of the approximate solution. The computational cost to obtain the approximate solution is much lower than that for SSMs because the dimension of basis vectors of preconditioned stochastic Krylov subspace is usually selected to be much smaller than the number of polynomial basis elements associated with the parametric space.

An adaptive stochastic Galerkin finite element method [22, 21, 36, 41, 42, 43] is a technique designed to achieve the desired accuracy with minimum of computational cost. Adaptive techniques rely on a posteriori error estimation, and they can be applied to both the spatial (finite element) and the parametric (polynomial) components of the SGFEM

approximations. An adaptive algorithm consists of four main procedures: solve, estimate, mark and refine. The algorithm will start by solving for the approximate solution. Then, the error estimators are utilised as indicators to mark the components of the approximate solution that require refinement. These procedures will be repeated until the prescribed tolerance is met.

1.1 Statement of the Problem

In this thesis, we are interested in preconditioning techniques for stochastic Galerkin matrices that arise when solving the steady-state diffusion equation with random data as follows:

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)) &= f(\boldsymbol{x}) && \text{in } D \times \Omega, \\ u(\boldsymbol{x},\omega) &= 0 && \text{on } \partial D \times \Omega, \end{aligned}$$

where $D \subset \mathbb{R}^d$, $d = 1, 2, 3$ is a bounded spatial domain, Ω is a sample space and $f \in L^2(D)$. The two representations of diffusion coefficient a are explored: affine and non-affine parametric.

This research focuses now just on SGFEM. SGFEM discretisation of the elliptic problems with random data generates an approximating tensor product spaces $X \otimes S$, where X is a finite element space associated with the domain D and S is a space of complete polynomial. This yields a linear system with a block coefficient matrix A, i.e., (see [89, 113])

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N_{\boldsymbol{y}}} \\ A_{21} & A_{22} & \cdots & A_{2N_{\boldsymbol{y}}} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_{\boldsymbol{y}}1} & A_{N_{\boldsymbol{y}}2} & \cdots & A_{N_{\boldsymbol{y}}N_{\boldsymbol{y}}} \end{bmatrix},$$

where A_{ij} denotes an $N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}$ matrix. $N_{\boldsymbol{x}}$ and $N_{\boldsymbol{y}}$ are the dimensions of the spatial and parametric spaces. Moreover, the system matrix can be written as the sum of Kronecker

products

$$A = \sum_{m=0}^{L} G_m \otimes K_m$$

where G_m and K_m denote $N_{\boldsymbol{y}} \times N_{\boldsymbol{y}}$ sparse stochastic Galerkin matrices (see [51]) and $N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}$ stiffness matrices, respectively. The number of terms L in the system matrix A depends on the representation of the diffusion coefficient.

The dimension of the linear system depends on three discretisation parameters: the mesh size, the number of active random variables and the degree of polynomial basis. Generally, the approximating space in the physical domain is represented by the space of piecewise polynomials. In contrast, there are two typical choices to represent the approximating space in the parameter domain: the space of complete polynomials and the tensor product polynomial space. The space of complete polynomials of degree n is a set of all polynomials with the total degree less or equal than n, whereas tensor product polynomial space of degree n is a set of all polynomials of degree less or equal than n in each parameter. The cardinality of the basis for both polynomial spaces, i.e., $N_{\boldsymbol{y}}$, increases rapidly with the degree of polynomials and the number of parameters. Consequently, the dimension of the linear system grows exponentially with these discretisation parameters. In turn, the growth in the dimension of the linear system as the number of parameters increases leads to an ill-conditioned system which affects significantly the computational effort to obtain the approximate solution. This situation is referred to as the curse of dimensionality.

Solving the coupled linear system which arises from the SGFEM is very challenging. Due to complexity of the problem, a direct solver is not an option for this type of problem. There are three types of iterative solvers for this task: multigrid methods [83, 46, 103, 29, 85, 100, 49], domain decomposition methods and Krylov subspace methods [111, 126, 127]. Multigrid methods are iterative solvers for solving a linear system that arises in the discretisation of partial differential equations, including SGFEM. The idea of multigrid methods is to reduce the error in some components of the approximate solution by projecting the error in the approximate solution to the grids of different sizes.

Then, the multigrid methods smoothen the approximate solution and project back to the original grid. Moreover, the approximate solution is obtained by a direct solver if the grid is sufficiently coarse. Theoretically, the convergence rate of multigrid methods does not depend on the mesh size. However, the parameters in the methods need to be selected carefully to achieve a fast convergence rate.

Domain decomposition methods [106] represent a powerful technique for large-scale problems. The domain decomposition approach is a divide and conquer algorithm by dividing the problem into many subproblems. The subproblems should be sufficiently small so that they can be solved by a direct method in parallel. Then, the solutions from each subproblem are merged by solving a linear system with a certain Schur complement. However, solving a linear system with the Schur complement is very challenging due to the structure of the Schur complement. By construction, this approach is suitable for parallel computing. Typically, the domain decomposition methods are not a popular choice to use as a solver.

Krylov subspace methods [106] are projection methods for solving linear systems. These methods are constructed by projecting the residual vector to a suitable subspace. They approximate the inverse of the coefficient matrix by constructing a polynomial of the coefficient matrix. The choices of the subspace give a variety of methods, such as conjugate gradient method (CG), minimal residual method (MINRES) or generalized minimal residual method (GMRES) which suit different situations. In many problems, Krylov subspace methods converge to the solution very fast, but the efficiency of the methods deteriorates when solving ill-conditioned system.

Preconditioning techniques are techniques aimed at improving the efficiency and robustness of a Krylov solver by solving an equivalent linear system. To improve the performance of the solvers, the complexity for solving a linear system with the preconditioner should not be high while the number of Krylov subspace iteration is reduced. Moreover, optimality of the preconditioner is also important, so the performance of the solver is independent of the problem size. More importantly, good preconditioner is a key to improve

the efficiency of the solver when solving the ill-conditioned system. For SGFEM, there are several preconditioners to apply with Krylov subspace methods [63, 95, 97, 73, 104, 125].

The mean-based preconditioner was introduced more than two decades ago in [63]. It was analysed in [97]. The mean-based preconditioner is one of the popular choices employed to improve the efficiency of a Krylov solver such as the CG method. It employs the mean of the random field to design the preconditioner. In addition, the block-diagonal structure of the mean-based preconditioner provides opportunities for parallelism. It was shown in [63] that the performance of the solver equipped with the mean-based preconditioner does not depend on the size of the problem.

The Kronecker product preconditioner was introduced and analysed in [125]. The preconditioner employs all the components arising in the stochastic Galerkin matrix. It preserves the Kronecker product structure as the mean-based preconditioner. The idea of the Kronecker product preconditioner is to find the best approximation of the coefficient matrix in the Frobenius norm while the right Kronecker factor is chosen to be the stiffness matrix associated with the mean term of the diffusion coefficient. Although the spectral analysis of the Kronecker product preconditioner cannot reflect its performance, it still outperforms the mean-based preconditioner.

Although the mean-based preconditioner and the Kronecker product preconditioner are studied for the steady-state diffusion problem, they can successfully apply to other problems. For example, they have been applied to the saddle point problems in [50, 99] and the steady-state Navier–Stokes equations in [98, 114]. In addition, they also perform very well in optimal control problems [19], for the nearly incompressible linear elasticity problem [78] and for the steady-state diffusion problem where the diffusion coefficient is the exponent of a random field [126].

There are other techniques to design an efficient preconditioner such as the domain decomposition or the hierarchical approach. It is possible to use domain decomposition as a solver but in many applications, the domain decomposition methods can be used to design a preconditioner for Krylov subspace methods [118, 121, 74, 120, 124, 119].

As a result, the preconditioner suits parallelism, which is the benefit of the domain decomposition methods. When it comes to the preconditioner design, approximating the Schur complement is thus the main challenge in designing a preconditioner via the domain decomposition technique. In addition, hierarchical approach [95, 115, 116] utilises the hierarchical structure of the problem by using a certain lower-order approximation. The solution from the lower-order approximation can be used to obtain the solution of the higher-order approximation, or it can be used as an initial guess of the iterative methods. Furthermore, hierarchical matrices ($\mathcal{H}$-matrices) [71, 15, 14, 26] is another hierarchical technique to approximate a sparse or dense matrix. A cluster tree is built and used to form a hierarchical structure of the matrix and ensure that each block's rank in the structure is less than a certain number. This low-rank approximation leads to reducing the usage of memory to store the matrix and also reduce the complexities for matrix-matrix operations. $\mathcal{H}$-matrices can be used to design a preconditioner and approximate the Schur complement for domain decomposition [13, 12, 54, 16, 17, 69].

1.2 Research Objective and Scope

The main aim of the thesis is to design fast iterative solvers for linear systems that arise from SGFEM discretizations with the space of complete polynomials of elliptic partial differential equations with random coefficients. We consider the random coefficient to be affine-parametric and non-affine-parametric, which are inspired, respectively, by the Karhunen-Loève expansion and the generalised polynomial chaos expansion of random fields. The elliptic partial differential equations with random data can be found in many real-world applications such as groundwater flow [79, 32, 4, 122] or biomedical engineering [56].

The choice of iterative solver is the CG method equipped with a preconditioner. We are interested in efficient preconditioners for the stochastic Galerkin linear system designed to improve the efficiency of the solver and thus reduce the total computational cost for

approximating the solution of the SPDE. Furthermore, we aim to make the convergence rate of the solver independent of the discretisation parameters of SGFEM.

Our first class of preconditioners comprises the truncation preconditioners and the modified truncation preconditioners (see [20]). Both preconditioners are designed for the case of affine-parametric diffusion coefficients. The truncation preconditioner is designed and analysed via a truncation of the diffusion coefficient. Our analysis shows that convergence rate is independent of the size of the problem. Due to the high complexity of solving a linear system with truncation preconditioners, the modified truncation preconditioners, which are approximations of the truncation preconditioner, are introduced. Their spectral analysis shows optimality with respect to discretisation parameters.

Next, we introduce the domain decomposition technique on the parametric domain for affine-parametric diffusion coefficients. The key idea of this technique is to identify a permutation so that the system matrix can be written as a 2-by-2 block matrix with the $(1,1)$-block having a diagonal block structure. This leads to the preconditioners which are suitable for parallelism. Additionally, we present a procedure for designing a suitable permutation. The new concept of a parametric mesh is introduced to represent the structure of the system matrix. The parametric mesh is defined via graph theory. We introduce the even-odd partitioning strategy which is one strategy to partition the parametric mesh. It results in several versions of the so-called even-odd preconditioners. Moreover, we also perform the spectral analysis of these preconditioners which indicates optimality with respect to problem size.

For the case of non-affine-parametric diffusion coefficients, we introduced several preconditioners by extending the above ideas from the preconditioners for affine-parametric diffusion coefficients. First, the diffusion coefficient is truncated and the truncation preconditioner for non-affine-parametric diffusion coefficients is introduced. It is defined by a bilinear form via the truncated coefficient. We then discuss the modified truncation preconditioner aimed at preserving the symmetry and positivity and reducing the complexity when solving a linear system with the preconditioner. In addition, we also generalise the

idea of domain decomposition preconditioner to the case of non-affine parametric coefficients to enhance the suitability for parallel computing. Specifically, we design a preconditioner for log-transformed diffusion coefficients and provide the spectral analysis for the case of the bounded parametric domain.

Finally, we report on the results from numerical experiments to compare the performance of our preconditioners with that of some existing preconditioners. We also perform the experiments to confirm the optimality of our preconditioners.

1.3 Thesis Organisation

This thesis is organised into seven chapters. In the first three chapters, we provide some background materials. Next, we present our main results in Chapters 4-6. The conclusions of this research are summarised in the last chapter of this thesis. The outlines for each chapter are included below.

In Chapter 1, we provide an introduction to UQ. The problem that we are interested in is discussed here, and we give the scope and objectives of the research.

The background concepts and results, such as random fields and their representations, are given in Chapter 2. The model problem is also stated together with the SGFEM formulation including the stochastic Galerkin linear system.

In Chapter 3, we review the Conjugate Gradient method and some properties related to our problem.

In Chapter 4, we review some existing preconditioners and their spectral analysis. Our first proposed preconditioning technique is introduced in this chapter. We outline our truncation preconditioners and present results for the case of affine-parametric diffusion coefficients. The complexity and spectral analysis of these preconditioners are provided.

Next, a domain decomposition technique on the parameter domain is presented in Chapter 5. This technique is outlined for the case of affine-parametric diffusion coefficients. Moreover, the concept of parametric mesh is introduced. We also describe how to

design a preconditioner based on this technique. Furthermore, we present three versions of even-odd preconditioners based on the domain decomposition technique on the parameter domain. The computational cost and the eigenvalue bounds of the preconditioned system for even-odd preconditioners are reported.

In Chapter 6, the preconditioners for the case of non-affine-parametric coefficients are introduced. They generalise the idea of truncation preconditioners and domain decomposition for affine-parametric coefficients. In addition, we present preconditioners for log-transformed diffusion coefficients. Again, the complexities for each preconditioner are also derived.

Finally, we summarise the present work and provide some suggestions for future research in Chapter 7.

CHAPTER 2

ELLIPTIC PDES WITH RANDOM DATA

Elliptic PDEs with random data represent a model that can be found in many research areas. In this case, random data means uncertainty in input data such as diffusion coefficients, boundary conditions, or forcing terms. In a probabilistic framework, uncertainties in the model can be represented via statistical quantities such as random fields or random variables.

The stochastic Galerkin finite element method is a powerful tool to approximate the solution of the model due to its fast convergence rate. The spatial space and parametric space are represented by finite-dimensional subspaces, i.e., piecewise polynomial space and the space of complete polynomials, respectively. This leads to a large coupled stochastic Galerkin linear system.

In this chapter, we provide some preliminary knowledge in connection with our problem. First, the definition of second-order random fields and some relevant properties are stated in section 2.1. Then, we discuss the representations of random fields, namely the Karhunen-Loève expansion and the generalised polynomial chaos expansion. In section 2.2, our model problem is stated, which is an elliptic PDE with random data. Subsequently, we derive a variational formulation for the model problem and then transform it to the variational formulation in parametric space in section 2.3. In order to discretise the variational formulation, the space of complete polynomials is introduced; we then construct orthogonal polynomials via a three-term recurrence relation before obtaining

the discrete formulation in section 2.4. Finally, in section 2.5, we discuss some basic properties of the linear system arising from our discretisation.

2.1 Random Fields

In this section, several definitions and relevant properties of random fields ([89, 87]) are introduced. Random fields extend the idea of random variables by taking values in Euclidean space $\mathbb{R}^d$ with $d = 1, 2, 3$. Let $(\Omega, \mathcal{F}(\Omega), \mathbb{P})$ be a complete probability space, where Ω is a set of all outcomes, $\mathcal{F}(\Omega)$ is a σ-algebra of events and $\mathbb{P} : \mathcal{F}(\Omega) \to [0, 1]$ is a probability measure, i.e., $\mathbb{P}(\Omega) = 1$. Before introducing the definition of random fields, let us introduce a useful definition of *almost surely*. Its concept is exactly the same as the concept of *almost everywhere* in measure theory, but *almost surely* is used in probability theory.

Definition 2.1. Let $F \in \mathcal{F}$ be an event. If $\mathbb{P}(F) = 1$, we say the event F happens almost surely (a.s. or $\mathbb{P}$-a.s.) with respect to a probability measure $\mathbb{P}$.

If an event F happens almost surely, there might be an event F' in the sample space Ω but $\mathbb{P}(F') = 0$. We will use this to state our problem in the next section.

Now, we define random fields as follows.

Definition 2.2. Let $D \subset \mathbb{R}^d$ be a bounded spatial domain with $d = 1, 2, 3$. A set $\{a(\boldsymbol{x}, \cdot) \mid \boldsymbol{x} \in D\}$ is called a random field if it is a set of real-valued random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Note that the function a maps $D \times \Omega$ to $\mathbb{R}$.

According to the definition of random fields, we can see that if $\boldsymbol{x} \in D$ is fixed, $a(\boldsymbol{x}, \cdot)$ can be viewed as a random variable in Ω. However, if $\omega \in \Omega$ is fixed, $a(\cdot, \omega)$ is a realisation of the random field a in D.

Additionally, there are some statistical quantities such as mean or variance, which are used to describe the behaviour of random fields. To ensure that these two quantities are well defined, we need the following definition.

Definition 2.3. For $D \subset \mathbb{R}^d$, a random field a is said to be second-order if $a(\boldsymbol{x}, \cdot) \in L^2(\Omega)$, for all $\boldsymbol{x} \in D$, i.e.,

$$\int_\Omega |a(\boldsymbol{x}, \cdot)|^2 \, \mathrm{d}\mathbb{P}(\omega) < \infty \quad \text{for all } \boldsymbol{x} \in D.$$

The mean and covariance functions of the random field a are well defined if a is second-order. That is, the mean function of the random field a is

$$\mathbb{E}\left[a(\boldsymbol{x}, \cdot)\right] := \int_\Omega a(\boldsymbol{x}, \omega) \mathrm{d}\mathbb{P}(\omega) \qquad \text{for each } \boldsymbol{x} \in D,$$

and its covariance function is

$$\mathrm{Cov}(\boldsymbol{x}_1, \boldsymbol{x}_2) \coloneqq \mathbb{E}\left[\left(a(\boldsymbol{x}_1, \cdot) - \mathbb{E}\left[a(\boldsymbol{x}_1, \cdot)\right]\right)\left(a(\boldsymbol{x}_2, \cdot) - \mathbb{E}\left[a(\boldsymbol{x}_2, \cdot)\right]\right)\right] \qquad \text{for all } \boldsymbol{x}_1, \boldsymbol{x}_2 \in D.$$

Moreover, the variance of the random field a is denoted by σ^2 and is defined by

$$\sigma^2 = \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{x}).$$

Next, we introduce a Bochner space which is related to second-order random fields. If $W(D)$ is a Banach space of real-valued functions, we define the Bochner space by

$$L^2_{\mathbb{P}}\left(\Omega, W(D)\right) = \left\{ v : \Omega \to W(D) \mid \int_\Omega \|v(\cdot, \omega)\|^2_{W(D)} \, \mathrm{d}\mathbb{P}(\omega) < \infty \right\},$$

with the corresponding norm $\|\cdot\|_{L^2_{\mathbb{P}}(\Omega, W(D))}$ defined by

$$\|v\|_{L^2_{\mathbb{P}}(\Omega, W(D))} = \left(\int_\Omega \|v(\cdot, \omega)\|^2_{W(D)} \, \mathrm{d}\mathbb{P}(\omega) \right)^{\frac{1}{2}} \qquad \text{for all } v \in L^2_{\mathbb{P}}\left(\Omega, W(D)\right).$$

In our case, we want all realisations of a to be functions in $L^2(D)$. Consequently, the random field a is a member of $L^2_{\mathbb{P}}\left(\Omega, L^2(D)\right)$. Note also that $L^2_{\mathbb{P}}\left(\Omega, L^2(D)\right)$ is a Banach space.

In the next section, we use all these definitions to establish representations of random

fields, i.e., the Karhunen-Loève expansion and the generalised polynomial chaos expansion. These representations are important because they separate the physical domain's data and the information on the stochastic part. The key difference between these two expansions is the choice of basis functions. The Karhunen-Loève expansion employs an orthogonal basis of the Bochner space $L^2_{\mathbb{P}}(\Omega, L^2(D))$ while the other employs an orthogonal basis of $L^2_{\mathbb{P}}(\Omega)$.

2.1.1 The Karhunen-Loève Expansion

The Karhunen-Loève expansion (KL expansion) [87, 89] is a popular representation of second-order random fields. However, the random field a can be expanded by KL expansion if the mean value and covariance function of the random field a are provided. The KL expansion is obtained through the spectral theorem for compact operators.

Let $a_0(\boldsymbol{x})$ and $\mathrm{Cov}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ be the mean and the covariance function of the random field a, respectively. We start with the Fredholm integral operator $C : L^2(D) \to L^2(D)$ whose kernel is the covariance function of the random field a. The integral operator C is defined by

$$[Cu](\boldsymbol{x}) = \int_D \mathrm{Cov}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) u(\tilde{\boldsymbol{x}}) \mathrm{d}\tilde{\boldsymbol{x}}. \tag{2.1}$$

The following theorem shows the relation between the integral operator C and the covariance function via the eigenpairs for the operator C.

Theorem 2.4 (Mercer [89, Theorem 1.80])**.** *Let D be a bounded spatial domain in $\mathbb{R}^d$ with $d = 1, 2, 3$ and $\mathrm{Cov} : D \times D \to \mathbb{R}$ be a symmetric and non-negative definite function. Let C be the corresponding integral operator defined by (2.1). If $\{\lambda_m\}_{m=1}^{\infty}$ and $\{\phi_m\}_{m=1}^{\infty}$ are the eigenvalues and eigenfunctions of the integral operator C with $\lambda_m > 0$ and $\|\phi_m\|_{L^2(D)} = 1$ for all $m = 1, 2, \ldots$, then, for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in D$,*

$$\mathrm{Cov}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{m=1}^{\infty} \lambda_m \phi_m(\boldsymbol{x}_1) \phi_m(\boldsymbol{x}_2).$$

Additionally, the series absolutely and uniformly converges on $D \times D$.

By Mercer's theorem, there exist positive eigenvalues λ_m and normalised eigenfunctions ϕ_m, $m = 1, 2, 3, ...$ for the operator C. Hence, the covariance function Cov can be expanded in a convergent series. The Karhunen-Loève expansion utilises the eigenpairs of the integral operator C as stated in the following theorem.

Theorem 2.5 (Karhunen-Loève Expansion [89, Theorem 7.52])**.** *Let $D \subset \mathbb{R}^d$ be a bounded spatial domain and $a \in L^2_{\mathbb{P}}(\Omega, L^2(D))$ be a second-order random field. Then, there exist $\{\phi_m\}_{m=1}^{\infty} \subset L^2(D)$ and $\{\lambda_m\}_{m=1}^{\infty} \subset \mathbb{R}^+$ such that*

$$a(\boldsymbol{x}, \omega) = a_0(\boldsymbol{x}) + \sum_{m=1}^{\infty} \sqrt{\lambda_m}\phi_m(\boldsymbol{x})Y_m(\omega), \tag{2.2}$$

where $a_0(\boldsymbol{x}) = \mathbb{E}[a(\boldsymbol{x}, \cdot)]$ and Y_m are random variables with zero mean and unit variance,

$$Y_m(\omega) = \frac{1}{\sqrt{\lambda_m}}(a(\boldsymbol{x}, \omega) - a_0(\boldsymbol{x}), \phi_m(\boldsymbol{x}))_{L^2(D)}.$$

Moreover, $\{Y_m\}_{m=1}^{\infty}$ are pairwise uncorrelated.

Note that $\{\phi_m Y_m\}_{m=1}^{\infty}$ is an orthogonal set with respect to the inner product $\langle \cdot, \cdot \rangle$ defined by

$$\langle u, v \rangle = \mathbb{E}\left[\int_D u(\boldsymbol{x}, \cdot)v(\boldsymbol{x}, \cdot)\mathrm{d}\boldsymbol{x}\right], \quad \text{for } u, v \in L^2_{\mathbb{P}}\left(\Omega, L^2(D)\right).$$

Also note that if the random field a is Gaussian, then the random variables Y_m are independent and identically distributed (iid) Gaussian distributions $\mathcal{N}(0, 1)$. In practice, a truncated KL expansion is used for computational purposes. Therefore, if we choose $M \in \mathbb{N}$, then we have

$$a(\boldsymbol{x}, \omega) \approx a_M(\boldsymbol{x}, \omega) = a_0(\boldsymbol{x}) + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(\boldsymbol{x})Y_m(\omega), \tag{2.3}$$

and the error from the truncation is

$$\|a - a_M\|^2_{L^2_{\mathbb{P}}(\Omega, L^2(D))} = \mathbb{E}\left[\int_D \left(\sum_{m=M+1}^{\infty} \sqrt{\lambda_m}\phi_m(\boldsymbol{x})Y_m\right)^2 \mathrm{d}\boldsymbol{x}\right] = \sum_{m=M+1}^{\infty} \lambda_m.$$

Thus, to reduce the error from the truncation, it is reasonable to reorder the terms in the random field by the magnitude of λ_m.

2.1.2 Generalised Polynomial Chaos Expansion

The polynomial chaos expansion (or Wiener chaos expansion) was introduced several decades ago by N. Wiener in [128]. It utilised Hermite polynomials in Gaussian random variables with zero mean and unit variance to expand a second-order random field $X \in L^2_{\mathbb{P}}(\Omega, L^2(D))$ with normal distribution. Note that we call Hermite polynomials in Gaussian random variables *the Hermite chaos* [134, 128]. One has

$$\begin{aligned} X(\boldsymbol{x}, \omega) =& c_0(\boldsymbol{x})H_0 + \sum_{m_1=1}^{\infty} c_{m_1}(\boldsymbol{x})H_1\left(Y_{m_1}(\omega)\right) \\ &+ \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{m_1} c_{m_1m_2}(\boldsymbol{x})H_2\left(Y_{m_1}(\omega), Y_{m_2}(\omega)\right) \\ &+ \sum_{m_1=1}^{\infty}\sum_{m_2=1}^{m_1}\sum_{m_3=1}^{m_2} c_{m_1m_2m_3}(\boldsymbol{x})H_3\left(Y_{m_1}(\omega), Y_{m_2}(\omega), Y_{m_3}(\omega)\right) + \ldots, \end{aligned}$$

where $c_{m_1\cdots m_k}(\boldsymbol{x})$ is a real-valued coefficient and $H_n\left(Y_{m_1}(\omega), \ldots, Y_{m_k}(\omega)\right)$ denotes the Hermite chaos of exact degree n in the random variables $Y_{m_1}(\omega), \ldots, Y_{m_k}(\omega)$. Moreover, Y_m are independent Gaussian random variables with zero mean and unit variance. The Hermite chaos H_n is a multivariate polynomial

$$H_n\left(Y_{m_1}(\omega), \ldots, Y_{m_k}(\omega)\right) = \prod_{i=1}^{k} h_{n_i}\left(Y_{m_i}\right),$$

where h_{n_i} is the univariate Hermite polynomial of degree n_i with $n = \sum_{i=1}^{k} n_i$. For convenience, we rewrite $X(\boldsymbol{x}, \omega)$ in the following form

$$X(\boldsymbol{x}, \omega) = \sum_{m=0}^{\infty} \hat{c}_m \psi_m \left(\boldsymbol{Y}(\omega) \right),$$

where $\boldsymbol{Y}(\omega) = (Y_1(\omega), Y_2(\omega), \ldots)$ and there is a one to one correspondence between $\psi_m \left(\boldsymbol{Y}(\omega) \right)$ and $H_n \left(Y_{m_1}(\omega), \ldots, Y_{m_k}(\omega) \right)$. Furthermore, $\{\psi_m\}_{m=0}^{\infty}$ forms a complete orthogonal basis in $L^2_{\mathbb{P}}(\Omega)$.

Note that the polynomial chaos expansion converges in $L^2_{\mathbb{P}}(\Omega, L^2(D))$. Moreover, we can expand a random field by the polynomial chaos expansion without providing the mean and the covariance of the random field, which is the benefits of the polynomial chaos expansion over the KL expansion.

In 2002, the polynomial chaos expansion was generalised by D. Xiu and G. Karniadakis in [134] to generalised polynomial chaos expansion (gPC) or Wiener-Askey chaos, in order to deal with general random inputs such as random fields with uniform distribution. They employ orthogonal polynomials which are generated by the probability density functions of the random fields as shown in Table 2.1.

Distribution	Wiener-Askey chaos	Support
Gaussian	Hermite chaos	$(-\infty, \infty)$
uniform	Legendre chaos	$[a, b]$
beta	Jacobi chaos	$[a, b]$
gamma	Laguerre chaos	$[0, \infty)$

Table 2.1: The correspondence between the type of distribution and the set of orthogonal polynomials.

The polynomial chaos forms a basis for the space $L^2_{\mathbb{P}}(\Omega)$ which is orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathbb{P}}$ defined by

$$\langle u, v \rangle_{\mathbb{P}} = \mathbb{E}[uv];$$

thus, a second-order random field a can be represented in the form

$$a(\boldsymbol{x},\omega) = \sum_{m=0}^{\infty} a_m(\boldsymbol{x})\psi_m\left(\boldsymbol{Y}(\omega)\right),$$

where ψ_m are orthogonal polynomials corresponding to the distribution of the random field a and a_m can be obtained by

$$a_m(\boldsymbol{x}) = \mathbb{E}\left[a(\boldsymbol{x},\omega)\psi_m\left(\boldsymbol{Y}(\omega)\right)\right].$$

However, in practice, the gPC expansion needs to be truncated to finite summation. There are several truncation schemes for the gPC expansion such as the total order expansion or the tensor-product expansion (see [44]). The total order expansion is a traditional truncation strategy for gPC expansion. It considers M random variables and orthogonal polynomials up to degree k. Thus, the random field a can be approximated by

$$a_k^M(\boldsymbol{x},\omega) = \sum_{m=0}^{N} a_m(\boldsymbol{x})\psi_m\left(\boldsymbol{Y}(\omega)\right),$$

where $\boldsymbol{Y}(\omega) = (Y_1(\omega), Y_2(\omega), \ldots, Y_M(\omega))$ and ψ_m are orthogonal polynomials whose degrees are less than or equal to k. Additionally, the truncated polynomial chaos expansion has $N+1$ terms where

$$N+1 = \begin{pmatrix} M+k \\ k \end{pmatrix}.$$

Another truncation strategy is the tensor-product expansion. This strategy also selects M random variables but the degree of the orthonormal polynomials in the ith random variable does not exceed k_i for $i = 1, 2, ..., M$. Thus, the number of terms by tensor-product expansion is

$$N+1 = \prod_{i=1}^{M} (k_i+1).$$

We can see that the number of terms in the truncated polynomial chaos expansion grow rapidly with the number of random variables and the maximum degree of orthonormal

polynomials.

Note that there are more truncation schemes to reduce the number of terms in gPC expansion such as the sparse gPC expansion based on the least angle regression (see [25]).

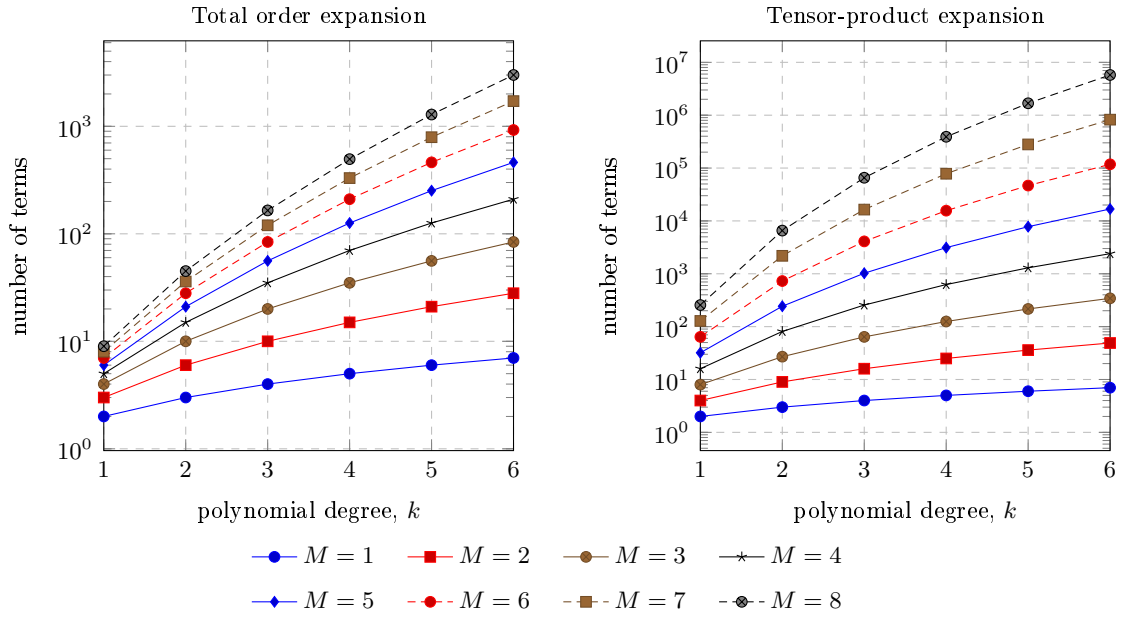


Figure 2.1: The number of terms in a truncated polynomial chaos expansion by total order expansion and tensor-product expansion grows rapidly with respect to M and k. (We assume $k = k_1 = ... = k_M$ for tensor-product expansion.)

Mean and Variance of Generalised Polynomial Chaos Expansion

Mean and variance are two important quantities in statistics. In this section, we will show how to find the mean and the variance of a random field via the gPC expansion. Suppose the random field a is represented by a gPC expansion as

$$a(\boldsymbol{x}, \omega) = \sum_{m=0}^{\infty} a_m(\boldsymbol{x}) \psi_m \left(\boldsymbol{Y}(\omega) \right),$$

where ψ_m are orthogonal polynomials and assume that $\psi_0 = 1$.

We start with the mean value of the random field a. That is

$$\mathbb{E}\left[a\right] = \sum_{m=0}^{\infty} a_m(\boldsymbol{x}) \mathbb{E}\left[\psi_m \left(\boldsymbol{Y}(\omega) \right)\right] = a_0(\boldsymbol{x}),$$

since $\mathbb{E}\left[\psi_m\left(\boldsymbol{Y}(\omega)\right)\right] = 0$ for all ψ_m with degree greater than 0.

Subsequently, before moving to the variance of the random field a, let us consider

$$\begin{aligned}\mathbb{E}\left[a^2\right] &= \mathbb{E}\left[\sum_{m=0}^{\infty}\sum_{m'=0}^{\infty} a_m(\boldsymbol{x})a_{m'}(\boldsymbol{x})\psi_m\left(\boldsymbol{Y}(\omega)\right)\psi_{m'}\left(\boldsymbol{Y}(\omega)\right)\right] \\ &= \sum_{m=0}^{\infty}\sum_{m'=0}^{\infty} a_m(\boldsymbol{x})a_{m'}(\boldsymbol{x})\mathbb{E}\left[\psi_m\left(\boldsymbol{Y}(\omega)\right)\psi_{m'}\left(\boldsymbol{Y}(\omega)\right)\right].\end{aligned}$$

Again, by the orthogonality of $\{\psi_m\}_{m=0}^{\infty}$, we have

$$\mathbb{E}\left[a^2\right] = \sum_{m=0}^{\infty}\sum_{m'=0}^{\infty} a_m(\boldsymbol{x})a_{m'}(\boldsymbol{x})\delta_{mm'} = \sum_{m=0}^{\infty} a_m^2(\boldsymbol{x}). \tag{2.4}$$

Hence,

$$\mathrm{Var}(a) = \mathbb{E}\left[a^2\right] - \mathbb{E}\left[a\right]^2 = \sum_{m=0}^{\infty} a_m^2(\boldsymbol{x}) - a_0^2(\boldsymbol{x}) = \sum_{m=1}^{\infty} a_m^2(\boldsymbol{x}). \tag{2.5}$$

To summarise, KL-expansion requires the mean and covariance of the random field. After that, we need to solve an eigenvalue problem to obtain the expansion. Solving this eigenvalue problem consumes high computational effort, which is the main drawback for KL-expansion. Also, KL-expansion cannot represent some random fields. For example, if a lognormal random field is expanded by KL expansion, the random variables will not be iid. However, the advantage of KL-expansion is that the truncated KL-expansion is optimal in the L^2 sense. On the other hand, gPC expansion does not need the mean or covariance of the random field. The number of terms in truncated gPC expansion grows rapidly with the truncation parameters, which could affect computation performance.

2.2 Elliptic Partial Differential Equations with Random Data

Let $D \subset \mathbb{R}^d$ be a bounded spatial domain with $d = 1, 2, 3$ and let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space. To state our model problem, let $a \in L^2_{\mathbb{P}}(\Omega, L^2(D))$ be a second-order random field and assume that $a(\cdot, \omega)$ is strictly positive, bounded and has a positive lower bound for almost all $\omega \in \Omega$. That is, there exist positive real numbers $a_{\min}$ and $a_{\max}$ such that

$$0 < a_{\min} \leq a(\boldsymbol{x}, \omega) \leq a_{\max} < \infty \qquad \text{a.e. in } D \times \Omega. \tag{2.6}$$

In particular, $a \in L^\infty_{\mathbb{P}}(\Omega, L^2(D))$. Note that there are some cases that the assumption (2.6) does not hold but there exists a unique solution (see Section 6.1 for more details). Now, we consider the steady-state diffusion problem with uncertainty in the diffusion coefficient a. We assume f to be a forcing term without uncertainty, i.e., $f \in L^2(D)$, and also assume homogeneous Dirichlet boundary conditions on ∂D. We want to find the function $u : \bar{D} \times \Omega \to \mathbb{R}$ such that $\mathbb{P}$-a.s.

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x}, \omega) \nabla u(\boldsymbol{x}, \omega)) &= f(\boldsymbol{x}) && \text{in } D \times \Omega, \\ u(\boldsymbol{x}, \omega) &= 0 && \text{on } \partial D \times \Omega. \end{aligned} \tag{2.7}$$

If $\omega \in \Omega$ is fixed in (2.7), the problem becomes a deterministic elliptic problem for which we can find the approximate solution by numerical methods for PDEs such as the finite element method. By the Lax-Milgram theorem, it is well known that the variational formulation of the elliptic partial differential equation (2.7) for fixed ω has a unique solution $u(\cdot, \omega)$, if assumption (2.6) holds.

In the next section, we derive the variational form of problem (2.7) and then change it to parametric form. Furthermore, we provide an existence and uniqueness theorem for the variational form before applying the stochastic Galerkin finite element methods to the problem.

2.3 Variational Formulation

Let $H_0^1(D)$ be the Sobolev space of the functions in H^1 which vanish on ∂D and let v be a function in $L^2_{\mathbb{P}}(\Omega, H_0^1(D))$. In order to obtain the variational form of problem (2.7), we multiply both sides of the equation in (2.7) by the function v and then integrate by parts over the spatial domain D before taking expectations. It leads to the variational form

$$\begin{cases} \text{Find } u \in L^2_{\mathbb{P}}(\Omega, H_0^1(D)) \text{ such that} \\ \mathbb{B}(u,v) = \mathbb{F}(v) \qquad \text{for all } v \in L^2_{\mathbb{P}}\left(\Omega, H_0^1(D)\right), \end{cases} \tag{2.8}$$

where the bilinear form $\mathbb{B}$ and the linear functional $\mathbb{F}$ are defined by

$$\begin{aligned} \mathbb{B}(u,v) &= \mathbb{E}\left[\int_D a(\boldsymbol{x},\cdot)\nabla u(\boldsymbol{x},\cdot)\cdot\nabla v(\boldsymbol{x},\cdot)\mathrm{d}\boldsymbol{x}\right], \\ \mathbb{F}(v) &= \mathbb{E}\left[\int_D f(\boldsymbol{x})v(\boldsymbol{x},\cdot)\mathrm{d}\boldsymbol{x}\right]. \end{aligned}$$

Theorem 2.6 ([89, Theorem 9.25]). *For $f \in L^2(D)$, assume that the diffusion coefficient a is positive and bounded as in (2.6). Then, there exists a unique solution $u \in L^2_{\mathbb{P}}\left(\Omega, H_0^1(D)\right)$ which satisfies the variational formulation (2.8).*

Theorem 2.6 shows that the variational formulation of our problem has a unique solution in $L^2_{\mathbb{P}}(\Omega, H_0^1(D))$. However, this variational form is still not convenient in order to find an approximation of u, because it involves integration over the set of outcomes Ω and also the probability measure $\mathbb{P}$. Recall that, by the representations of a random field, the second-order random field a can be expanded by either the KL expansion

$$a(\boldsymbol{x},\omega) = a_0(\boldsymbol{x}) + \sum_{m=1}^{\infty}\sqrt{\lambda_m}\phi_m(\boldsymbol{x})Y_m(\omega),$$

or by the generalised polynomial chaos expansion

$$a(\boldsymbol{x},\omega) = \sum_{m=0}^{\infty} a_m(\boldsymbol{x})\psi_m\left(Y_1(\omega), Y_2(\omega), \ldots\right).$$

Thus, the random field a can be represented as a function of independent random variables $Y_m(\omega)$ for $m \in \mathbb{N}$. Hence, we replace these random variables by parametric variables which represent the range of random variables.

Let $y_m = Y_m(\omega)$ and $\Gamma_m \coloneqq Y_m(\Omega) \subseteq \mathbb{R}$ for all $m \in \mathbb{N}$. Let π_m be a probability measure on $(\Gamma_m, \mathcal{B}(\Gamma_m))$ and ρ_m be the probability density function for the random variable Y_m, i.e., $\pi_m(B) = \int_B \rho_m(y_m)\mathrm{d}y_m$ for any $B \in \mathcal{B}(\Gamma_m)$. Additionally, ρ_m are assumed to be even for all $m \in \mathbb{N}$. By our assumption on random variables, Y_m is a function from the sample space Ω to Γ_m, and $y_m \in \Gamma_m$ for all $m \in \mathbb{N}$. As a result, the random field a can be written in terms of $\boldsymbol{y} = (y_1, y_2, \ldots)$ as follows

$$a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{m=1}^{\infty} \sqrt{\lambda_m}\phi_m(\boldsymbol{x})y_m,$$

or using the generalised polynomial chaos expansion

$$a(\boldsymbol{x}, \boldsymbol{y}) = \sum_{m=0}^{\infty} a_m(\boldsymbol{x})\psi_m(\boldsymbol{y}).$$

Let $\boldsymbol{\Gamma} \coloneqq \prod_{m=1}^{\infty} \Gamma_m$ and $\boldsymbol{\pi} \coloneqq \prod_{m=1}^{\infty} \pi_m$ be a probability measure on $(\boldsymbol{\Gamma}, \mathcal{B}(\boldsymbol{\Gamma}))$. Recall that $\{Y_m\}_{m=1}^{\infty}$ are pairwise uncorrelated. The joint density function $\rho : \boldsymbol{\Gamma} \to \mathbb{R}$ of the associated multivariate random variable $\boldsymbol{y} \in \boldsymbol{\Gamma}$ is defined by

$$\rho(\boldsymbol{y}) = \prod_{m=1}^{\infty} \rho_m(y_m). \tag{2.9}$$

Additionally, it is obvious that if the random field a satisfies condition (2.6), then

$$0 < a_{\min} \leq a(\boldsymbol{x}, \boldsymbol{y}) \leq a_{\max} < \infty \qquad \text{a.e. in } D \times \boldsymbol{\Gamma}. \tag{2.10}$$

Next, define a weighted space $L^2_\rho(\boldsymbol{\Gamma}, W(D))$ by

$$L^2_\rho(\boldsymbol{\Gamma}, W(D)) = \left\{ v : D \times \boldsymbol{\Gamma} \to \mathbb{R} \mid \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \|v(\cdot, \boldsymbol{y})\|^2_{W(D)} \,\mathrm{d}\boldsymbol{y} < \infty \right\},$$

and the norm $\|\cdot\|_{L^2_\rho(\mathbf{\Gamma},W(D))}$ by

$$\|v\|^2_{L^2_\rho(\mathbf{\Gamma},W(D))} = \int_{\mathbf{\Gamma}} \rho(\boldsymbol{y}) \, \|v(\cdot,\boldsymbol{y})\|^2_{W(D)} \, \mathrm{d}\boldsymbol{y}.$$

Note that for a function $v(\boldsymbol{x},\omega) \in L^2_{\mathbb{P}}(\Omega, W(D))$, where $W(D)$ is a Banach space, if v can be represented in $\boldsymbol{x}$ and $Y_m(\omega)$, where $Y_m(\omega)$ are independent for $m \in \mathbb{N}$, we have

$$\|v\|_{L^2_{\mathbb{P}}(\Omega,W(D))} = \|v\|_{L^2_\rho(\mathbf{\Gamma},W(D))}.$$

Next, we replace $a(\boldsymbol{x},\omega)$ and $L^2_{\mathbb{P}}(\Omega, H^1_0(D))$ by $a(\boldsymbol{x},\boldsymbol{y})$ and $V := L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$, respectively, in the variational formulation (2.8) and get the variational formulation on $D \times \mathbf{\Gamma}$:

$$\begin{cases} \text{Find } u \in V \text{ such that} \\ B(u,v) = F(v), \qquad \forall v \in V, \end{cases} \tag{2.11}$$

where the bilinear form $B : V \times V \to \mathbb{R}$ and the functional $F : V \to \mathbb{R}$ are defined by

$$B(u,v) = \int_{\mathbf{\Gamma}} \rho(\boldsymbol{y}) \int_D a(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}, \tag{2.12}$$

$$F(v) = \int_{\mathbf{\Gamma}} \rho(\boldsymbol{y}) \int_D f(\boldsymbol{x}) v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \tag{2.13}$$

Theorem 2.7 ([89, Theorem 9.43]). *Assume that condition (2.10) holds for the random field a. If $f \in L^2(D)$ then there exists a unique solution $u \in L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$ which satisfies problem (2.11).*

In the next section, we provide the discrete formulation of the weak formulation (2.11). After that, the stochastic Galerkin finite element method is applied to the discrete formulation. We also introduce the space of complete polynomials in the variable $\boldsymbol{y}$ and explain how to construct a finite dimensional subspace of the Bochner space $L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$ to obtain a fully-discrete finite-dimensional problem.

2.4 Discrete Formulation

Let $\tilde{V} \subset L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$ be a finite dimensional subspace. A stochastic Galerkin solution $\tilde{u}$ of problem (2.7) is a function in $\tilde{V}$. Thus, we get the discrete formulation as follows.

$$\begin{cases} \text{Find } \tilde{u} \in \tilde{V} \text{ such that} \\ B(\tilde{u}, v) = F(v) \qquad \text{for all } v \in \tilde{V}, \end{cases} \tag{2.14}$$

where the bilinear form B and the functional F are stated in (2.12) and (2.13), respectively.

Theorem 2.8 ([89, Theorem 9.50]). *Let $f \in L^2(D)$ and $\tilde{V}$ be a finite subspace of the space $L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$. If assumption (2.10) holds, then there exists a unique solution $\tilde{u} \in \tilde{V}$ satisfying the fully-discrete weak problem (2.14).*

In order to apply the stochastic Galerkin FEM to our model problem, we approximate the solution u by using a gPC expansion with M random variables and the degree of the orthogonal polynomials no greater than k. Each term of the polynomial chaos expansion consists of a coefficient function in the spatial domain and an orthogonal polynomial in the parameters. Consider now the weighted space $L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$. It is known that the space $L^2_\rho(\mathbf{\Gamma}, H^1_0(D))$ is isometrically isomorphic to the space $L^2_\rho(\mathbf{\Gamma}) \otimes H^1_0(D)$ (see [110, Remark C.24]) where $L^2_\rho(\mathbf{\Gamma})$ denotes

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

with the associated inner product

$$\langle u, v \rangle_\rho = \int_{\mathbf{\Gamma}} \rho(\boldsymbol{y}) u(\boldsymbol{y}) v(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}.$$

We need a finite dimensional subspace of the space $L^2_\rho(\mathbf{\Gamma}) \otimes H^1_0(D)$. Firstly, define the space X_h of continuous piecewise linear polynomials defined on a shape-regular and conforming triangulation $\mathcal{T}_h$ of the domain D, where h denotes mesh size. More precisely, we

define

$$X_h \coloneqq \left\{ v \in H_0^1(D) \mid v|_K \in \mathbb{P}_1(K) \text{ for all } K \in \mathcal{T}_h \right\},$$

where $\mathbb{P}_1(K)$ denotes the set of all polynomials of degree 1 or less on K. We choose nodal basis functions ϕ_j for X_h. Therefore,

$$X_h \coloneqq \operatorname{span}\{\phi_1, \phi_2, \ldots, \phi_{N_{\boldsymbol{x}}}\} \subset H_0^1(D).$$

We now need to generate a subspace of $L^2_\rho(\boldsymbol{\Gamma})$. First, we start with the definition of multi-indices.

Definition 2.9. A multi-index $\boldsymbol{\alpha} \in \mathbb{N}_0^{\mathbb{N}}$ is a sequence of non-negative integers $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \ldots)$ with compact support, that is $\max(\operatorname{supp}\boldsymbol{\alpha}) < \infty$, where $\operatorname{supp}\boldsymbol{\alpha} \coloneqq \{m \in \mathbb{N} \mid \alpha_m \neq 0\}$.

Additionally, we define

$$|\boldsymbol{\alpha}| \coloneqq \sum_{m \in \operatorname{supp}\boldsymbol{\alpha}} \alpha_m, \qquad \boldsymbol{\alpha}! \coloneqq \prod_{m \in \operatorname{supp}\boldsymbol{\alpha}} \alpha_m!$$

and

$$\mathbb{I}_k^M \coloneqq \left\{ \boldsymbol{\alpha} \in \mathbb{N}_0^{\mathbb{N}} \mid \max(\operatorname{supp}\boldsymbol{\alpha}) \leq M \text{ and } |\boldsymbol{\alpha}| \leq k \right\}.$$

We choose as a subspace of $L^2_\rho(\boldsymbol{\Gamma})$ the space of complete polynomials S_k^M, which is the set of all polynomials of total degree less than or equal to k in M variables. That is;

$$S_k^M \coloneqq \operatorname{span}\left\{ \prod_{m=1}^{M} y_m^{\alpha_m} \mid \boldsymbol{\alpha} \in \mathbb{I}_k^M \right\}.$$

Note that the dimension of the space of complete polynomials is

$$N_{\boldsymbol{y}} \coloneqq \dim(S_k^M) = \frac{(M+k)!}{M!k!}.$$

Consequently, $S_k^M \otimes X_h$ is a finite subspace of $L^2_\rho(\boldsymbol{\Gamma}) \otimes H_0^1(D)$.

Let us now introduce an orthogonal basis of $L^2_{\rho_m}(\Gamma_m)$ and recall that the probability

density function ρ_m is assumed to be even for all $m = 1, 2, ..., M$. We construct a sequence of orthonormal polynomials P_j with respect to the inner product

$$\langle u, v \rangle_{\rho_m} = \int_{\Gamma_m} \rho_m(y_m) u(y_m) v(y_m) \mathrm{d}y_m,$$

via a three-term recurrence, as shown in the following theorem.

Theorem 2.10 ([55, Theorem 1.29]). *Let P_j^m, $j = 0, 1, 2, ...,$ be orthonormal polynomials with respect to the inner product $\langle \cdot, \cdot \rangle_{\rho_m}$. Then,*

$$c_{j+1}^m P_{j+1}^m(y_m) = \left(y_m - b_j^m\right) P_j^m(y_m) - c_j^m P_{j-1}^m(y_m), \qquad \text{for } j = 0, 1, 2, \ldots, \tag{2.15}$$

with $P_{-1}^m = 0$ and $P_0^m = 1$. The constant c_j^m is a normalising factor such that $\left\|P_j^m\right\|_{\rho_m}^2 = \left\langle P_j^m, P_j^m \right\rangle_{\rho_m} = 1$.

Now, multiply the equation (2.15) by P_j^m and then apply the inner product $\langle \cdot, \cdot \rangle_{\rho_m}$. It results in

$$b_j^m = -\left\langle y_m P_j^m, P_j^m \right\rangle_{\rho_m}.$$

Since ρ_m is assumed to be an even function and Γ_m is symmetric, so $b_j^m = 0$. Hence, the three-term recurrence can be simplified to

$$c_{j+1}^m P_{j+1}^m(y_m) = y_m P_j^m(y_m) - c_j^m P_{j-1}^m(y_m), \qquad \text{for } j = 0, 1, 2, \ldots, \tag{2.16}$$

with $P_{-1}^m = 0$ and $P_0^m = 1$.

Example 2.1. Let $\Gamma = [-1, 1]$. Suppose y is a uniform random variable from a sample space Ω to Γ with mean zero and unit variance. Thus, the probability density function of the random variable y is $\rho(y) = \frac{1}{2}$. The three-term recurrence (2.16) yields an orthogonal polynomials sequence $\{L_j\}_{j=0}^{\infty}$ defined by

$$c_{j+1} L_{j+1}(y) = y L_j(y) - c_j L_{j-1}(y), \qquad \text{for } j = 0, 1, 2, \ldots,$$

with $L_{-1} = 0$ and $L_0 = 1$ where $c_j = \frac{j}{\sqrt{2j+1}\sqrt{2j-1}}$. Therefore,

$$L_1(y) = \sqrt{3}y, \qquad L_2(y) = \frac{\sqrt{5}}{2}\left(3y^2 - 1\right), \qquad L_3(y) = \frac{\sqrt{7}}{3}\left(5y^3 - 3y\right), \qquad \cdots$$
$$c_1 = \frac{1}{\sqrt{3}}, \qquad c_2 = \frac{2}{\sqrt{15}}, \qquad c_3 = \frac{3}{\sqrt{35}}. \qquad \cdots$$

The polynomials $L_j(y)$ are called the normalized Legendre polynomials.

Furthermore, for $k \in \mathbb{N}$ and $j = 1, ..., k-1$, we have

$$y_m P_j^m(y_m) = c_j^m P_{j-1}^m(y_m) + c_{j+1}^m P_{j+1}^m(y_m).$$

If $\mathbf{v} = [\ P_0^m(y_m) \quad P_1^m(y_m) \quad \cdots \quad P_{k-1}^m(y_m)\ ]^T$, the three-term recurrence gives the following identity,

$$y_m \mathbf{v} = T_k^m \mathbf{v} + c_k^m P_k^m(y_m)\mathbf{e}_k,$$

where

$$T_k^m = \begin{bmatrix} 0 & c_1^m & & & \\ c_1^m & 0 & c_2^m & & \\ & \ddots & \ddots & \ddots & \\ & & c_{k-2}^m & 0 & c_{k-1}^m \\ & & & c_{k-1}^m & 0 \end{bmatrix} \text{ and } \mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{2.17}$$

The above identity provides a connection between the roots of P_k^m and the matrix T_k^m, that is the eigenvalues of T_k^m are the roots of P_k^m.

Define $\psi_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathbb{I}$ by

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}) = \prod_{m=1}^{\infty} P_{\alpha_m}^m(y_m).$$

We can see that $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathbb{I}}$ is an orthonormal set of polynomials with respect to the inner product $\langle\cdot,\cdot\rangle_\rho$. That is $\langle\psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}\rangle_\rho = 1$ if $\boldsymbol{\alpha} = \boldsymbol{\beta}$, otherwise $\langle\psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}\rangle_\rho = 0$. Thus, we define an orthonormal basis $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathbb{I}}$ for the space $L^2_\rho(\boldsymbol{\Gamma})$ and use $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha}\in\mathbb{I}_k^M}$ as the basis for $S_k^M \subset L^2_\rho(\boldsymbol{\Gamma})$.

It is obvious that there exists a bijection $q : \{1, 2, \ldots, N_{\boldsymbol{y}}\} \to \mathbb{I}_k^M$. Let us illustrate

this with an example.

Example 2.2. Consider the space of complete polynomials of degree less or equal to 2 ($k = 2$) with 3 uniform random variables ($M = 3$) with zero mean and unit variance on $\Gamma_m = [-1, 1]$, $m = 1, 2, 3$. The dimension of S_2^3 is $N_{\boldsymbol{y}} = \dim(S_2^3) = (3+2)!/(3!2!) = 10$. To be precise,

$$\begin{aligned}\mathbb{I}_2^3 = \{&(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0),\\ &(1,0,1), (0,1,1), (2,0,0), (0,2,0), (0,0,2)\}.\end{aligned}$$

Hence, all the orthogonal basis functions for S_2^3 can be written as

$$\psi_{(\alpha_1,\alpha_2,\alpha_3)}(y_1, y_2, y_3) = L_{\alpha_1}(y_1)L_{\alpha_2}(y_2)L_{\alpha_3}(y_3), \qquad \text{for all } \boldsymbol{\alpha} \in \mathbb{I}_2^3.$$

Remark. Theorem 2.10 is used to create a set of orthogonal polynomials when a random field's distribution law or probability density function is provided. Because the random field's distribution law induces the orthogonal polynomials (see Table 2.1) in its representation and we also use this distribution law to define the basis of the space S_k^M. As a result, the orthogonal polynomials in the random field's representation are the same orthogonal polynomials in the basis for the space of complete polynomials.

2.5 The Stochastic Linear System

First, we assume that the random field a can be written as

$$a(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \mathbb{I}} a_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}$$

where $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha} \in \mathbb{I}}$ is the orthonormal polynomial basis of $L^2_\rho(\boldsymbol{\Gamma})$. It is clear that this representation of a is in the form of a gPC expansion. Define $\mathbb{I}_k := \{\boldsymbol{\alpha} \in \mathbb{I} \mid |\boldsymbol{\alpha}| \leq k\}$. This representation of a can also represent the KL expansion by setting $a_{\boldsymbol{\alpha}} = 0$ for each $\boldsymbol{\alpha}$

outside $\mathbb{I}_1$.

In this section we derive the stochastic linear system corresponding to the discrete formulation (2.14). Define

$$V_{hk}^M \coloneqq S_k^M \otimes X_h = \operatorname{span}\left\{\varphi_{ij}(\boldsymbol{x},\boldsymbol{y}) \coloneqq \phi_i(\boldsymbol{x})\psi_{q(j)}(\boldsymbol{y}) \text{ for } 1 \le i \le N_{\boldsymbol{x}} \text{ and } 1 \le j \le N_{\boldsymbol{y}}\right\},$$

and choose $\tilde{V}$ in discrete formulation (2.14) to be V_{hk}^M. Since $\tilde{u} \in S_k^M \otimes X_h \subset L_p^2\left(\boldsymbol{\Gamma}, H_0^1(D)\right)$, we denote an approximate solution $\tilde{u}$ corresponding to the space V_{hk}^M by u_{hk}^M. Thus,

$$u_{hk}^M(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{N_{\boldsymbol{x}}}\sum_{j=1}^{N_{\boldsymbol{y}}} u_{ij}\varphi_{ij}(\boldsymbol{x},\boldsymbol{y}) \tag{2.18}$$

and select the test function $v(\boldsymbol{x},\boldsymbol{y}) = \varphi_{rs}(\boldsymbol{x},\boldsymbol{y})$ for $r = 1,2,\ldots,N_{\boldsymbol{x}}$ and $s = 1,2,\ldots,N_{\boldsymbol{y}}$. The discrete formulation (2.14) leads to

$$\begin{aligned}\sum_{\boldsymbol{\alpha}\in\mathbb{I}}\sum_{i=1}^{N_{\boldsymbol{x}}}\sum_{j=1}^{N_{\boldsymbol{y}}} u_{ij}\int_D a_{\boldsymbol{\alpha}}(\boldsymbol{x})\nabla\phi_i(\boldsymbol{x})\cdot\nabla\phi_r(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\psi_{q(j)}(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}&\\ = \int_D f(\boldsymbol{x})\phi_r(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}.&\end{aligned}$$

By the representation of a, the left hand side of the above equation is an infinite summation over $\mathbb{I}$. Fortunately, since $q(j), q(s) \in \mathbb{I}_k^M$, one has $\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\psi_{q(j)}(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = 0$ if $\operatorname{supp}\boldsymbol{\alpha} > M$. Moreover, by [77, Theorem 4.1],

$$\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\psi_{q(j)}(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = 0, \quad \text{if } |\boldsymbol{\alpha}| > 2k.$$

Hence, the infinite summation is implicitly truncated to the finitely many terms as follows

$$\begin{aligned}\sum_{\boldsymbol{\alpha}\in\mathbb{I}_{2k}^M}\sum_{i=1}^{N_{\boldsymbol{x}}}\sum_{j=1}^{N_{\boldsymbol{y}}} u_{ij}\int_D a_{\boldsymbol{\alpha}}(\boldsymbol{x})\nabla\phi_i(\boldsymbol{x})\cdot\nabla\phi_r(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\psi_{q(j)}(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}&\\ = \int_D f(\boldsymbol{x})\phi_r(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\int_{\boldsymbol{\Gamma}}\rho(\boldsymbol{y})\psi_{q(s)}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}.&\end{aligned}$$

This yields the linear system $A\mathbf{u} = \mathbf{b}$. The coefficient matrix A and also vectors $\mathbf{u}$ and $\mathbf{b}$ are written in block form as follows

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N_{\boldsymbol{y}}} \\ A_{21} & A_{22} & \cdots & A_{2N_{\boldsymbol{y}}} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_{\boldsymbol{y}}1} & A_{N_{\boldsymbol{y}}2} & \cdots & A_{N_{\boldsymbol{y}}N_{\boldsymbol{y}}} \end{bmatrix}, \ \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N_{\boldsymbol{y}}} \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_{N_{\boldsymbol{y}}} \end{bmatrix},$$

where

$$\begin{aligned} A_{js} &= \sum_{\boldsymbol{\alpha} \in \mathbb{I}_{2k}^M} \left\langle \psi_{\boldsymbol{\alpha}} \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho K_{\boldsymbol{\alpha}}, && j, s = 1, 2, \ldots, N_{\boldsymbol{y}}, \\ [K_{\boldsymbol{\alpha}}]_{ir} &= \int_D a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \nabla \phi_i(\boldsymbol{x}) \cdot \nabla \phi_r(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, && \boldsymbol{\alpha} \in \mathbb{I}_{2k}^M \text{ and } i, r = 1, 2, \ldots, N_{\boldsymbol{x}}, \end{aligned}$$

and

$$\begin{aligned} \mathbf{u}_j &= [\ u_{1j} \quad u_{2j} \quad \cdots \quad u_{N_{\boldsymbol{x}}j}\ ]^T, && j = 1, 2, \ldots, N_{\boldsymbol{y}}, \\ [\mathbf{b}_s]_r &= \int_\Gamma \rho(\boldsymbol{y}) \psi_{q(s)}(\boldsymbol{y}) \mathrm{d}\boldsymbol{y} \cdot \int_D f(\boldsymbol{x}) \phi_r(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, && r = 1, 2, \ldots, N_{\boldsymbol{x}}, \text{ and } s = 1, 2, \ldots, N_{\boldsymbol{y}}. \end{aligned}$$

If we define the matrix $G_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathbb{I}_{2k}^M$ by

$$[G_{\boldsymbol{\alpha}}]_{js} \coloneqq \left\langle \psi_{\boldsymbol{\alpha}} \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho, \qquad j, s = 1, 2, \ldots, N_{\boldsymbol{y}}, \tag{2.19}$$

the coefficient matrix A can be expressed as the summation of Kronecker products

$$A = \sum_{\boldsymbol{\alpha} \in \mathbb{I}_{2k}^M} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}. \tag{2.20}$$

Note that, by (2.4) and (2.5), if $\psi_{q(1)} = 1$, the mean and variance of the solution u are given by

$$\mathbb{E}\left[u_{hk}^M(\boldsymbol{x}, \cdot)\right] = u_1(\boldsymbol{x}), \quad \mathrm{Var}(u_{hk}^M(\boldsymbol{x}, \cdot)) = \sum_{j=2}^{N_{\boldsymbol{y}}} u_j^2(\boldsymbol{x}),$$

respectively, where $u_j(\boldsymbol{x}) \coloneqq \sum_{i=1}^{N_{\boldsymbol{x}}} u_{ij} \phi_i(\boldsymbol{x})$.

It is obvious that the stochastic Galerkin matrix A is symmetric. Moreover, by (2.10), the matrix A is positive definite. Generally, the matrix A is block dense; however, in the case of KL expansions, the matrix A can be written as a summation over the set $\mathbb{I}_1^M$ whose cardinality is $M+1$. For convenience, we use natural numbers instead of multi-indices, i.e.

$$A = \sum_{m=0}^{M} G_m \otimes K_m. \tag{2.21}$$

Moreover, the following theorem guarantees that the Galerkin matrix A is a block-sparse matrix as there are no more than $2M+1$ block matrices per row.

Theorem 2.11 ([89, Theorem 9.58, 9.59])**.** *Suppose that ρ satisfies (2.9), and ρ_m is even for $m = 1, 2, \ldots, M$. Then,*

$$G_0 = I_{N_{\boldsymbol{y}}},$$

while for $m = 1, 2, \ldots, M$, the matrix G_m whose entries are (cf. (2.19)) has 2 non-zero entries per row. More precisely,

$$\left\langle y_m \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho = \begin{cases} c^m_{\beta_m+1}, & \beta_{m'} = \beta'_{m'} \text{ for all } m' \in \{1,\ldots,M\} \setminus \{m\} \text{ and } \beta_m = \beta'_m - 1, \\ c^m_{\beta_m}, & \beta_{m'} = \beta'_{m'} \text{ for all } m' \in \{1,\ldots,M\} \setminus \{m\} \text{ and } \beta_m = \beta'_m + 1, \\ 0, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\beta} = q(j), \boldsymbol{\beta}' = q(s)$ and $c^m_{\beta_m}$ is a normalising constant in (2.15).

Example 2.3. Consider the space S_k^M of complete polynomials of degree less than or equal to 2 in 2 uniform random variables. That is $M = k = 2$ and $\rho(y_m) = \frac{1}{2}$ for $m = 1, 2$. Therefore,

$$\mathbb{I}_2^2 = \{(0,0), (1,0), (0,1), (1,1), (2,0), (0,2)\}.$$

Since $P_1^m(y_m) = \sqrt{3} y_m$ and

$$[G_m]_{js} = \left\langle \frac{1}{c_1^m} y_m \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho = \sqrt{3} \left\langle y_m \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho,$$

for $m = 1, 2$, we obtain G_1 and G_2 as follows,

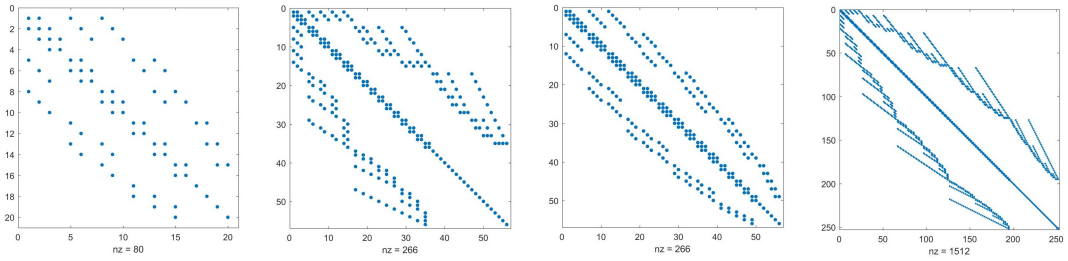
$$G_1 = \sqrt{3}\begin{bmatrix} 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & \frac{2}{\sqrt{15}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 \\ 0 & \frac{2}{\sqrt{15}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_2 = \sqrt{3}\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{3}} & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & \frac{2}{\sqrt{15}} \\ 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\sqrt{15}} & 0 & 0 & 0 \end{bmatrix}.$$

Since G_m induces the sparsity pattern of the matrix A, in order to observe the sparsity pattern of the matrix A (see [89, p. 409]), we can consider the pattern of the matrix

$$\sum_{m=0}^{M} G_m.$$

Example 2.4. Consider the spaces of complete polynomials S_3^3, S_5^3, S_3^5 and S_5^5. The patterns of the stochastic Galerkin matrix A in these cases are shown in Figure 2.2. Each plot represents the block pattern of the Galerkin matrix A of size $N_{\boldsymbol{y}} \times N_{\boldsymbol{y}}$, where a blue dot is a non-zero block matrix of size $N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}$.

Furthermore, according to Theorem 2.11, the pattern of the Galerkin matrix A also



(a) (b) (c) (d)

Figure 2.2: These four figures illustrate the pattern of the stochastic Galerkin matrix A when the complete polynomial space S_k^M is constructed by an orthonormal basis. The figures (a) and (b) correspond to the case $M = 3$, $k = 3$ and $M = 3$, $k = 5$, respectively, and the figures (c) and (d) correspond to the case $M = 5$, $k = 3$ and $M = 5$, $k = 5$, respectively.

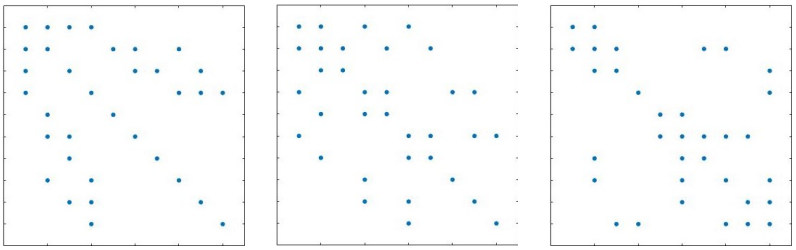
depends on the bijection map q. Thus, if the map q is changed to a new bijection map q', then the pattern of the matrix A will also be different.

Example 2.5. Consider the space S_k^M of complete polynomials degree 2 with 3 random variables, i.e., $M = 3$ and $k = 2$, so that A is a block matrix of size 10×10. Define maps q_1, q_2 and q_3 from $\{1, 2, ..., 10\}$ to $\mathbb{I}_2^3$ as in Table 2.2.

n	$q_1(n)$	$q_2(n)$	$q_3(n)$
1	$(0,0,0)$	$(0,0,0)$	$(2,0,0)$
2	$(0,0,1)$	$(1,0,0)$	$(1,0,0)$
3	$(0,1,0)$	$(2,0,0)$	$(1,0,1)$
4	$(1,0,0)$	$(0,1,0)$	$(0,0,2)$
5	$(0,0,2)$	$(1,1,0)$	$(0,2,0)$
6	$(0,1,1)$	$(0,0,1)$	$(0,1,0)$
7	$(0,2,0)$	$(1,0,1)$	$(1,1,0)$
8	$(1,0,1)$	$(0,2,0)$	$(0,0,0)$
9	$(1,1,0)$	$(0,1,1)$	$(0,1,1)$
10	$(2,0,0)$	$(0,0,2)$	$(0,0,1)$

Table 2.2: The maps q_1, q_2 and q_3 from $\{1, 2, ..., 10\}$ to $\mathbb{I}_2^3$.

Each plot in Figure 2.3 represents the pattern of the matrix A for a different map q but with the same number of non-zero entries.



(a) (b) (c)

Figure 2.3: The figures (a), (b) and (c) show the patterns of the Galerkin matrix A for the case of the complete polynomial space S_2^3 for the maps q_1, q_2 and q_3, respectively.

We can see that the function q plays a vital role to manipulate the pattern of the matrix A. Some patterns of the coefficient matrix such as block-diagonal or block-tridiagonal can improve the efficiency of a solver. For instance, parallel computing can be applied to solve a linear system with block-diagonal coefficient.

Generally, applying a numerical method such as the finite element method to a deterministic problem leads to a linear system with a stiffness matrix that contains the information from the physical domain. We have seen in this chapter that the SGFEM combines stochastic Galerkin matrices, which include the data from the stochastic part, with stiffness matrices and introduces a large coupled linear system. The dimensions of these stochastic matrices and stiffness matrices grow exponentially with the discretisation parameters. As a result, the linear system that arises from SGFEM is much larger than the linear system from the deterministic problem. Consequently, this adversely affects the computational effort for solving such a linear system. Additionally, the pattern of the coupled linear system is very complex. For the case of KL expansions, the coefficient matrix is block-sparse, whereas the coefficient matrix is block-dense in the case of gPC expansions. Hence, to deal with this problem, a powerful solver equipped with an efficient preconditioner is needed.

CHAPTER 3

THE CONJUGATE GRADIENT METHOD

According to the construction of the stochastic Galerkin matrix A, the matrix A is symmetric and positive definite. Additionally, the matrix A is block-sparse by Theorem 2.11 if the random field a is given in the form of the KL expansion. Thus, the conjugate gradient method is a suitable option for solving a linear system with our symmetric and positive definite coefficient matrix A. The important feature of the conjugate gradient method is that it guarantees convergence of the solution.

In this chapter, we present the conjugate gradient method (CG) together with its preconditioned version ([106]). They are iterative methods for solving a linear system $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. We choose an iterative method as a solver for our problem because of the sparsity and size of matrix A which is usually large because it is a summation of the Kronecker product between stochastic matrices and stiffness matrices. First, we provide a brief review of Krylov subspace methods and the definition of the Krylov subspace in section 3.1. Next, we review the conjugate gradient method and its algorithm. We also provide some properties, and most importantly, a well-known convergence theorem. Finally, we discuss the preconditioned CG in section 3.2 and why preconditioners are important to our problem. The preconditioned CG algorithm is also provided.

3.1 The Conjugate Gradient Method

Krylov subspace methods are iterative methods based on a projection technique for solving the linear system

$$A\mathbf{x} = \mathbf{b}, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ with initial guess $\mathbf{x}_0$. Let $\mathcal{K}_m$ and $\mathcal{L}_m$ be subspaces of $\mathbb{R}^n$ with $\dim \mathcal{K}_m = \dim \mathcal{L}_m = m$. A projection technique will find an approximation $\mathbf{x}_m$ to $\mathbf{x}$ as a member of $\mathbf{x}_0 + \mathcal{K}_m$ and with residual vector $\mathbf{r}_m \coloneqq \mathbf{b} - A\mathbf{x}_m$ orthogonal to $\mathcal{L}_m$. That is

$$\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m,$$

and

$$\mathbf{b} - A\mathbf{x}_m \perp \mathbf{w} \qquad \text{for all } \mathbf{w} \in \mathcal{L}_m.$$

For Krylov subspace methods, different choices of the subspace $\mathcal{L}_m$ yield a variety of Krylov subspace methods. However, the subspace $\mathcal{K}_m$ is fixed and is defined as follows.

Definition 3.1. The Krylov subspace associated with or generated by A and $\mathbf{r}_0$ is denoted by $\mathcal{K}_m(A, \mathbf{r}_0)$ and is defined by

$$\mathcal{K}_m(A, \mathbf{r}_0) = \operatorname{span}\left\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{m-1}\mathbf{r}_0\right\}$$

where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x_0}$. Note that $\mathcal{K}_m\left(A, \mathbf{r}_0\right)$ will be denoted by $\mathcal{K}_m$.

The conjugate gradient method, which is probably the best-known Krylov subspace method, employs $\mathcal{L}_m = \mathcal{K}_m$ where $\mathcal{K}_m$ is the mth Krylov subspace introduced in Definition 3.1.

Since the approximate solution $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m$, then $\mathbf{x}_m = \mathbf{x}_0 + q_{m-1}(A)\mathbf{r}_0$ where q_{m-1} is a polynomial of degree $m-1$. Since $\mathbf{x} = \mathbf{x}_0 + A^{-1}\mathbf{r}_0$, the choice of approximation $\mathbf{x}_m$ indicates that A^{-1} is approximated by $q_{m-1}(A)$.

Assuming the coefficient matrix A in (3.1) is symmetric and positive definite, it induces

the so-called A-inner product $(\cdot,\cdot)_A$ defined via

$$(\mathbf{u},\mathbf{v})_A = (A\mathbf{u},\mathbf{v}) \qquad \mathbf{u},\mathbf{v}\in\mathbb{R}^n,$$

where $(\cdot,\cdot)$ denotes the Euclidean inner product.

Note that we can define the A-norm $\|\cdot\|_A$ by $\|\mathbf{v}\|_A^2 = (\mathbf{v},\mathbf{v})_A$ for $\mathbf{v}\in\mathbb{R}^n$. We now set $\mathbf{p}_0 = \mathbf{r}_0$ and construct a set of vectors $\{\mathbf{p}_i\}_{i=0}^m$ orthogonal with respect to the inner product $(\cdot,\cdot)_A$ by choosing the next search direction to be

$$\mathbf{p}_{m+1} = \mathbf{r}_{m+1} + \beta_m \mathbf{p}_m.$$

Hence, $\mathbf{x}_m$ can be written as

$$\mathbf{x}_{m+1} = \mathbf{x}_m + \alpha_m \mathbf{p}_m = \mathbf{x}_0 + \sum_{i=0}^{m} \alpha_i \mathbf{p}_i,$$

where $\alpha_0,\alpha_1,\ldots,\alpha_m\in\mathbb{R}$. Since $\mathbf{r}_{m+1} = \mathbf{r}_m - \alpha_m A\mathbf{p}_m$, we choose α_m such that $\mathbf{r}_{m+1}$ is orthogonal to $\mathbf{r}_m$ with respect to the standard inner product $(\cdot,\cdot)$. Thus,

$$\alpha_m = \frac{(\mathbf{r}_m,\mathbf{r}_m)}{(\mathbf{p}_m,\mathbf{p}_m)_A}.$$

Again, since $\mathbf{p}_{m+1}$ is orthogonal to $\mathbf{p}_m$ with respect to the A-inner product, we have

$$\beta_m = \frac{(\mathbf{r}_{m+1},\mathbf{r}_{m+1})}{(\mathbf{r}_m,\mathbf{r}_m)}.$$

We combine these results and get the pseudocode for the CG algorithm in Algorithm 1.

As we can see, each iteration of the the CG algorithm mainly consists of an inner product and a matrix-vector multiplication. As a result, the cost of matrix-vector multiplication dominates the cost of one iteration of the CG algorithm. The CG method has some important properties as the following theorem shows.

Algorithm 1 The CG algorithm for solving the linear system $A\mathbf{x} = \mathbf{b}$

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\mathbf{p}_0 = \mathbf{r}_0$
For $m = 0, 1, 2, \ldots$ until convergence
    $\alpha_m = (\mathbf{r}_m, \mathbf{r}_m) \,/\, (\mathbf{p}_m, \mathbf{p}_m)_A$
    $\mathbf{x}_{m+1} = \mathbf{x}_m + \alpha_m \mathbf{p}_m$
    $\mathbf{r}_{m+1} = \mathbf{r}_m - \alpha_m A\mathbf{p}_m$
    $\beta_m = (\mathbf{r}_{m+1}, \mathbf{r}_{m+1}) \,/\, (\mathbf{r}_m, \mathbf{r}_m)$
    $\mathbf{p}_{m+1} = \mathbf{r}_{m+1} + \beta_m \mathbf{p}_m$
End For

Theorem 3.2 ([106, Proposition 6.20]). *Let* $\{\mathbf{r}_m\}$ *be the set of residual error vectors and* $\{\mathbf{p}_m\}$ *be the set of auxiliary vectors produced by Algorithm 1 for* $m = 0, 1, 2, ..., n$. *Then,* $(\mathbf{r}_{m_1}, \mathbf{r}_{m_2}) = 0$ *and* $(\mathbf{p}_{m_1}, \mathbf{p}_{m_2})_A = 0$ *for all* $m_1 \neq m_2$.

An immediate consequence of this result is that the conjugate gradient algorithm will terminate or converge within n iterations. However, we expect the Algorithm 1 is terminated in m iterations where $m \ll n$. In addition, the sequence $(\mathbf{x}_m)_{m\in\mathbb{N}_0}$ in Algorithm 1 converges to the exact solution with the rate of convergence as shown in the following results.

Theorem 3.3 ([106, Theorem 6.29]). *Let* $\mathbf{x}_m$ *be the approximate solution obtained at the* m*th step of the CG algorithm and let* $\mathbf{x}$ *be the exact solution. Then*

$$\|\mathbf{x} - \mathbf{x}_m\|_A \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^m \|\mathbf{x} - \mathbf{x}_0\|_A$$

where $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$.

By Theorem 3.3, it is easy to see that the rate of convergence depends on the eigenvalues of the coefficient matrix A. If $\lambda_{\min}$ is close to $\lambda_{\max}$, then the number of iterations is likely to be small. Moreover, to ensure that the relative error with respect to the norm

$\|\cdot\|_A$ is less than a given $tol \ll 1$, it is sufficient to require that

$$2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^m \leq tol.$$
$$m\log\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right) \leq \log\frac{tol}{2}.$$

Since $\log\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right) \approx \frac{-2}{\sqrt{\kappa(A)}}$, we have

$$m \geq \frac{1}{2}\sqrt{\kappa(A)}\left|\log\frac{tol}{2}\right|.$$

Remark. *tol* can be any positive real numbers. However, a discretisation of the model problem causes an approximation error that cannot be removed. Thus, it is sufficient to solve the linear system from the discretisation formulation with an accuracy in the same order as the discretisation error. If *tol* is too small, it may cause an overfitted model.

To conclude, we can see that the number of iterations appears to grow linearly with $\sqrt{\kappa(A)}$. However, it will not exceed the dimension of the matrix A due to $\mathbf{x}$ being approximated by a linear combination of orthogonal vectors $\{\mathbf{p}_i\}_{i=0}^m$.

3.2 Preconditioned Conjugate Gradient Method

If the coefficient matrix A is ill-conditioned, i.e., $\lambda(A)$ is close to zero or very large, it results in requiring many iteration counts of the CG algorithm before convergence. A preconditioner plays a vital role in the convergence of iteration methods. Generally speaking, a preconditioner is needed to improve convergence of the CG algorithm if the coefficient matrix A is ill-conditioned.

Moreover, according to the convergence Theorem 3.3 of the CG algorithm, the rate of convergence depends only on the coefficient matrix A. Then, the linear system $A\mathbf{x} = \mathbf{b}$

Algorithm 2 PCG algorithm for solving Ax=b with a preconditioner P.

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\mathbf{z}_0 = P^{-1}\mathbf{r}_0$
$\mathbf{p}_0 = \mathbf{z}_0$
For $m = 0, 1, 2, \ldots$ until convergence
    $\alpha_m = (\mathbf{r}_m, \mathbf{z}_m) \,/\, (\mathbf{p}_m, \mathbf{p}_m)_A$
    $\mathbf{x}_{m+1} = \mathbf{x}_m + \alpha_m \mathbf{p}_m$
    $\mathbf{r}_{m+1} = \mathbf{r}_m - \alpha_m A\mathbf{p}_m$
    $\mathbf{z}_{m+1} = P^{-1}\mathbf{r}_{m+1}$
    $\beta_m = (\mathbf{r}_{m+1}, \mathbf{z}_{m+1}) \,/\, (\mathbf{r}_m, \mathbf{z}_m)$
    $\mathbf{p}_{m+1} = \mathbf{z}_{m+1} + \beta_m \mathbf{p}_m$
End For

can be changed to a preconditioned linear system

$$\hat{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}. \tag{3.2}$$

such that $\kappa(\hat{A}) < \kappa(A)$. Since the CG method requires the coefficient matrix $\hat{A}$ to be symmetric and positive definite, we set

$$\hat{A} = P^{-\frac{1}{2}} A P^{-\frac{1}{2}}, \;\; \hat{\mathbf{x}} = P^{\frac{1}{2}}\mathbf{x} \text{ and } \hat{\mathbf{b}} = P^{-\frac{1}{2}}\mathbf{b},$$

where the matrix P is a symmetric and positive definite matrix such that the condition number is smaller. Hence, the number of iterations of the CG algorithm might be decreased if the CG is applied to the preconditioned linear system instead.

If the CG method is applied to the preconditioned linear system (3.2), we can derive the CG algorithm with a preconditioner P as shown in the Algorithm 2.

The preconditioned Conjugate Gradient algorithm (PCG) performs an inner-product and a matrix-vector multiplication but also solves the linear system

$$P\mathbf{z}_m = \mathbf{r}_m \tag{3.3}$$

at each iteration. Therefore, the cost for solving such a linear system should not be very high. Additionally, there is one important feature required in a preconditioner: optimality

with respect to problem size. This concept is described below.

Definition 3.4. Let $\{A_n\}$, $\{P_n\}$ be the sets of $n \times n$ matrices with $n \in \{n_1, n_2, ..., n_f\}$ and let θ, Θ be positive real numbers independent of n such that for all $n \in \{n_1, n_2, ..., n_f\}$

$$\theta \leq \lambda \left(P_n^{-1} A_n\right) \leq \Theta.$$

We call the set $\{P_n\}$ an optimal preconditioning set for $\{A_n\}$ with respect to the size n.

If the matrix P is an optimal preconditioner, then the condition number $\kappa(\hat{A})$ is bounded by a constant independent of n. As a result, the iteration counts by PCG are also bounded with respect to the problem size.

Equivalent Bilinear Forms and Optimal Preconditioners

Optimal preconditioners may be designed via equivalent operators (see [40], [52]) or equivalent norms (see [88]). It is well known that operators and bilinear forms are connected. Thus, we introduce the following definitions.

Definition 3.5. Let V be a vector space and B,$\tilde{B} : V \times V \to \mathbb{R}$ be positive definite symmetric bilinear forms. The bilinear forms B and $\tilde{B}$ are said to be equivalent if there exist positive numbers θ, Θ such that

$$\theta \tilde{B}(v, v) \leq B(v, v) \leq \Theta \tilde{B}(v, v) \qquad \text{for all } v \in V.$$

Definition 3.6. Let A, $P \in \mathbb{R}^{n \times n}$ be symmetric and positive definite matrices. The matrices A and P are spectrally equivalent if there exists positive numbers θ, Θ such that

$$\theta \mathbf{v}^T P \mathbf{v} \leq \mathbf{v}^T A \mathbf{v} \leq \Theta \mathbf{v}^T P \mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^n.$$

The following proposition shows a connection between spectrally equivalence of matrices A and P and boundedness of eigenvalues of $P^{-1}A$.

Proposition 3.7. *Let A, $P \in \mathbb{R}^{n\times n}$ be symmetric and positive definite matrices. If the matrices A and P are spectrally equivalent, then*

$$\theta \leq \lambda(P^{-1}A) \leq \Theta,$$

where θ and Θ are the constants of equivalence.

Proof. Let $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. Since A and P are spectrally equivalent, we have

$$\theta \leq \frac{\mathbf{v}^T A\mathbf{v}}{\mathbf{v}^T P\mathbf{v}} \leq \Theta.$$

Since P is symmetric and positive definite, we set $\mathbf{w} = P^{\frac{1}{2}}\mathbf{v}$ and get

$$\theta \leq \frac{\mathbf{w}^T P^{-\frac{1}{2}} A P^{-\frac{1}{2}}\mathbf{w}}{\mathbf{w}^T\mathbf{w}} \leq \Theta.$$

By Rayleigh quotient, we have

$$\theta \leq \lambda\left(P^{-\frac{1}{2}} A P^{-\frac{1}{2}}\right) \leq \Theta.$$

We finish the proof by applying the fact that $\lambda\left(P^{-\frac{1}{2}} A P^{-\frac{1}{2}}\right) = \lambda(P^{-1}A)$.

□

Since the coefficient matrix A in (2.20) is induced by the bilinear form (2.12), we may design an optimal preconditioner by finding an equivalent bilinear form as shown in the following proposition.

Proposition 3.8. *Let B,$\tilde{B} : V \times V \to \mathbb{R}$ be positive definite symmetric bilinear forms which are equivalent. Let $\tilde{V} = \operatorname{span}\{\varphi_1, \ldots, \varphi_n\} \subset V$. Define matrices A and P by*

$$A_{ij} = B(\varphi_i, \varphi_j) \text{ and } P_{ij} = \tilde{B}(\varphi_i, \varphi_j).$$

Then, A and P are spectrally equivalent.

Additionally, P is an optimal preconditioner for A.

Proof. Since the bilinear forms B and $\tilde{B}$ are equivalent and $\tilde{V}$ is a subspace of V, there exist positive numbers θ, Θ such that

$$\theta\tilde{B}(v,v) \leq B(v,v) \leq \Theta\tilde{B}(v,v) \qquad \text{for all } v \in \tilde{V}.$$

Let $\mathbf{v} \in \mathbb{R}^n$ and set $v = \sum_{i=1}^{n} \mathbf{v}_i\varphi_i$. Then, $B(v,v) = \mathbf{v}^T A\mathbf{v}$ and $\tilde{B}(v,v) = \mathbf{v}^T P\mathbf{v}$. We have

$$\theta\mathbf{v}^T P\mathbf{v} \leq \mathbf{v}^T A\mathbf{v} \leq \Theta\mathbf{v}^T P\mathbf{v} \qquad \text{for all } \mathbf{v} \in \mathbb{R}^n.$$

Hence, P is spectrally equivalent to the matrix A and also an optimal preconditioner for A.

□

To summarise, the CG method is a powerful iterative method for the symmetric and positive definite linear system due to a fast convergence rate and guarantee convergence. A preconditioner is key to improving the efficiency of Krylov subspace methods. In the case of the CG method, a preconditioner should cluster the eigenvalues of the preconditioned linear system, and the cost of the action of its inverse on a vector should not be very high. Additionally, optimality of a preconditioner is also important in order to bound the PCG iteration counts. In the next chapter, we will introduce a class of optimal preconditioner for SGFEM to improve the efficiency of the CG method.

CHAPTER 4

TRUNCATION PRECONDITIONERS FOR AFFINE PARAMETRIC DIFFUSION COEFFICIENTS

The SGFEM is a powerful method for PDEs with random data. However, the method results in a large coupled linear system of equations which can be represented in block form with a specific structure. Additionally, the coefficient matrix is ill-conditioned with respect to the spatial and parametric discretisation parameters (see [97, Lemma 3.7]). Consequently, this affects the efficiency of the Conjugate Gradient method. Thus, a preconditioning strategy is mandatory.

Block preconditioning is a natural technique for a block matrix. An effective preconditioner should be similar to the matrix A in some sense, and the computational cost of the action of its inverse on a vector should not be expensive. Moreover, if the size of the problem is very large, applying parallel computing when solving a linear system with the system matrix as a preconditioner would be beneficial.

Truncation preconditioners are one of the possible generalisations of the mean-based preconditioner (see [97]). The mean-based preconditioner employs the most important term, which is the block diagonal part corresponding to the mean of the diffusion coefficient. In contrast, truncation preconditioners are constructed via the most significant $r+1$ terms of the coefficient function where $r \in \mathbb{N}_0$.

This chapter will introduce a family of truncation preconditioners for the stochastic

Galerkin matrix A for the case where a is an affine parametric diffusion coefficient. First, we introduce the representation of the diffusion coefficient and some necessary conditions in section 4.1. Then, we will review some existing preconditioners such as the mean-based preconditioner and the Kronecker product preconditioner in section 4.2. Furthermore, their spectral analysis is also provided to compare with truncation preconditioners. Next, in section 4.3, we define a family of bilinear forms and show that the bilinear forms are equivalent to the bilinear form B in (2.12). We use these equivalent bilinear forms to construct a family of preconditioners. In section 4.4 we present an approximation of the preconditioner by a symmetric block Gauss-Seidel preconditioner, so that the computational cost is acceptable and we also show that our preconditioners are optimal with respect to discretisation parameters. Finally, the performance of the truncation preconditioner is compared with other preconditioners in section 4.6.

4.1 The Affine Parametric Diffusion Coefficient

Assume $\Gamma_m := [-1, 1]$ for all $m \in \mathbb{N}$. Let a be a random field represented as

$$a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{m=1}^{\infty} a_m(\boldsymbol{x}) y_m, \qquad \boldsymbol{x} \in D, \boldsymbol{y} \in \boldsymbol{\Gamma}, \tag{4.1}$$

where $a_m \in L^\infty(D)$ for all $m \in \mathbb{N}$.

In order to preserve positivity of the random field a, we also assume that there exist positive numbers $a_0^{\min}$ and $a_0^{\max}$ such that (see [110, Proposition 2.22])

$$0 < a_0^{\min} \leq a_0(\boldsymbol{x}) \leq a_0^{\max} \quad \text{for almost all } \boldsymbol{x} \in D \tag{4.2}$$

and

$$\tau := \frac{1}{a_0^{\min}} \left\| \sum_{m=1}^{\infty} |a_m| \right\|_{L^\infty(D)} < 1. \tag{4.3}$$

This implies that the random field a satisfies the assumption (2.10) with

$$a_{\min} \coloneqq a_0^{\min}(1-\tau) \text{ and } a_{\max} \coloneqq a_0^{\max} + a_0^{\min}\tau. \tag{4.4}$$

By the representation of the random field a, applying SGFEM to the SPDE (2.7) leads to the linear system $A\mathbf{u} = \mathbf{b}$, where

$$A = \sum_{m=0}^{M} G_m \otimes K_m \tag{4.5}$$

and, for $m = 1, 2, ..., M$,

$$\begin{aligned} [G_m]_{js} &= \left\langle y_m \psi_{q(j)}, \psi_{q(s)} \right\rangle_\rho, && j, s = 1, 2, \ldots, N_{\boldsymbol{y}}, \\ [K_m]_{ir} &= \int_D a_m(\boldsymbol{x}) \nabla \phi_i(\boldsymbol{x}) \cdot \nabla \phi_r(\boldsymbol{x}) d\boldsymbol{x}, && i, r = 1, 2, \ldots, N_{\boldsymbol{x}}. \end{aligned} \tag{4.6}$$

Unfortunately, the following theorem shows that the stochastic Galerkin matrix A associated with the mesh size h as defined in (4.5) is ill-conditioned if h is very small.

In order to study the eigenvalue bounds of the system matrix A, we define

$$\eta(\boldsymbol{x}) = \frac{1}{a_0(\boldsymbol{x})} \sum_{m=1}^{\infty} |a_m(\boldsymbol{x})|$$

and note that

$$1 - \eta(\boldsymbol{x}) \leq \frac{a(\boldsymbol{x}, \boldsymbol{y})}{a_0(\boldsymbol{x})} \leq 1 + \eta(\boldsymbol{x}), \qquad \text{for all } (\boldsymbol{x}, \boldsymbol{y}) \in D \times \boldsymbol{\Gamma},$$

and obviously, by the definition of τ, we have $0 < \eta(\boldsymbol{x}) \leq \tau < 1$ for all $\boldsymbol{x} \in D$. Additionally, we define the bilinear form $B_0 : V \times V \to \mathbb{R}$ by

$$B_0(u, v) = \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_0(\boldsymbol{x}) \nabla u(\boldsymbol{x}, \boldsymbol{y}) \cdot \nabla v(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y}. \tag{4.7}$$

Theorem 4.1. *Let a be a function in the form (4.1) and satisfying the assumption (4.2)*

and (4.3). Suppose the matrices G_m are defined as in (4.6) with normalised Legendre polynomial in uniform random variables on a bounded interval $[-1,1]$, and a piecewise linear approximation is used for the spatial domain. If the matrix A is the stochastic Galerkin matrix defined in (4.5), then

$$\Lambda(A) \subset [\alpha_1 a_{\min} h^2, \alpha_2 a_{\max}],$$

where $a_{\min}$ and $a_{\max}$ are defined in (4.4), h is the discretisation parameter for the spatial domain, and α_1 and α_2 are constants which are independent of h, M and k.

Proof. Let $v \in V_k^M$ and consider

$$\begin{aligned} B(v,v) &= \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a(\boldsymbol{x},\boldsymbol{y}) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_0(\boldsymbol{x}) \left(1+\eta(\boldsymbol{x})\right) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &\leq (1+\tau) B_0(v,v). \end{aligned}$$

On the other hand, we consider

$$\begin{aligned} B_0(v,v) &= \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_0(\boldsymbol{x}) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \frac{1}{(1-\eta(\boldsymbol{x}))} a(\boldsymbol{x},\boldsymbol{y}) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &\leq \frac{1}{1-\tau} B(v,v). \end{aligned}$$

Therefore, we get

$$(1-\tau)\, B_0(v,v) \leq B(v,v) \leq (1+\tau)\, B_0(v,v). \tag{4.8}$$

Since the bilinear forms B and B_0 induce the matrices A and $G_0 \otimes K_0$, respectively, then we have

$$(1-\tau)\, \mathbf{v}^T (G_0 \otimes K_0)\, \mathbf{v} \leq \mathbf{v}^T A \mathbf{v} \leq (1+\tau)\, \mathbf{v}^T (G_0 \otimes K_0)\, \mathbf{v},$$

where the vector $\mathbf{v}$ corresponds to the function v. By [97, Lemma 3.5], we know that

$$\Lambda(G_0 \otimes K_0) \subset [a_0^{\min}\alpha_1 h^2, a_0^{\max}\alpha_2],$$

where α_1 and α_2 are constants which are independent of h, M and k. As a result,

$$(1-\tau)\, a_0^{\min}\alpha_1 h^2 \mathbf{v}^T\mathbf{v} \leq \mathbf{v}^T A\mathbf{v} \leq (1+\tau)\, a_0^{\max}\alpha_2 \mathbf{v}^T\mathbf{v}.$$

□

We can see from Theorem 4.1 that if the mesh size is too small, the minimum eigenvalue of the coefficient matrix A approaches 0. This results in a deterioration of the bound on convergence of the CG method in Theorem 3.3, in practice, CG requires many iterations before convergence. For example, reducing the mesh size by half will double the number of CG iterations approximately. Therefore, a preconditioner is needed.

4.2 Some Existing Block Preconditioners

In this section, we review some important preconditioners for the stochastic Galerkin matrix, which are the mean-based preconditioner [63, 97] and the Kronecker product preconditioner [125]. These preconditioners can also be applied with other iterative methods such as multigrid methods or domain decomposition methods as we discuss in section 1.2. The mean-based preconditioner uses only the most dominant term in the coefficient matrix as a preconditioner, whereas the Kronecker product preconditioner approximates the coefficient matrix in Kronecker product form and uses all the data, i.e., all the matrices G_m and K_m, to construct a preconditioner.

4.2.1 Mean-based Preconditioners

By the KL expansion, the random field a can be written as an infinite series by starting with the mean value of the random field a, namely $a_0(\boldsymbol{x}) = \mathbb{E}[a(\boldsymbol{x}, \cdot)]$. The key idea of mean-based preconditioners is to choose the most significant term in the coefficient matrix A. We can obtain the mean-based preconditioner, which is denoted by P_0, via the bilinear form B_0. The mean-based preconditioner is defined by

$$P_0 = G_0 \otimes K_0.$$

Note that the assumption (4.2) on a_0 leads to the positivity of the preconditioner P_0. We can see that P_0 is also the first term of the stochastic Galerkin matrix A. It implies that P_0 is also the most important term of the matrix A.

For mean-based preconditioners, the eigenvalue bounds of the matrix $P_0^{-1}A$ are provided in the following theorem.

Theorem 4.2. *Let a be a random diffusion coefficient in the form (4.1). Suppose that the random variables are pairwise uncorrelated with even probability density functions. Then, the following eigenvalue bounds hold*

$$\Lambda\left(P_0^{-1}A\right) \subseteq [1-\tau, 1+\tau],$$

where τ denotes the constant in (4.3).

Proof. Since we have that the bilinear forms B and B_0 are equivalent in (4.8) and the bilinear forms B and B_0 induce the matrices A and P_0, respectively, then we have that

$$1-\tau \leq \lambda\left(P_0^{-1}A\right) \leq 1+\tau.$$

□

According to the above theorem, the eigenvalue bounds are independent of discretisa-

tion parameters, that is, the mesh size h, the number of active random variables M and the degree k of the space of complete polynomials. Thus, P_0 is an optimal preconditioner. The eigenvalues of the preconditioned linear system cluster around one. Therefore, we can expect the number of iterations by the CG method to be reduced. In each CG iteration, it is required to solve a system with P_0 as a coefficient matrix.

To analyse its complexity, we denote the number of flops for a sequential *operation* and a parallel *operation* by $\mathcal{F}\ell\,(operation)$ and $\mathcal{F}\ell p\,(operation)$, respectively. Furthermore, $\mathrm{nnz}(A)$ denotes the number of non-zeros in the matrix A.

Due to the structure of G_0 which is a diagonal matrix, i.e., $G_0 = I_{N_y}$, P_0 is a block diagonal matrix with K_0 along the main diagonal. Thus, we need to solve $N_{\boldsymbol{y}}$ linear systems with the coefficient matrix K_0. Hence,

$$\mathcal{F}\ell\left(P_0^{-1}\mathbf{v}\right) = N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{b}\right).$$

Moreover, we may tackle these linear systems in parallel, which would require $N_{\boldsymbol{y}}$ processors. For instance, for $\mathbf{v} \in \mathbb{R}^{N_x N_y}$, we solve $P_0\mathbf{x} = \mathbf{v}$ by

$$K_0\mathbf{x}_i = \mathbf{v}_i,$$

where $\mathbf{v}_i = \mathbf{v}\left(((i-1)N_{\boldsymbol{x}}+1) : iN_x\right)$ and $\mathbf{x}_i = \mathbf{x}\left(((i-1)N_{\boldsymbol{x}}+1) : iN_{\boldsymbol{x}}\right)$ for $i = 1, 2, \ldots, N_{\boldsymbol{y}}$. For this reason, the parallel complexity for solving a linear system with the system matrix P_0 is

$$\mathcal{F}\ell p\left(P_0^{-1}\mathbf{v}\right) = \mathcal{F}\ell\left(K_0^{-1}\mathbf{b}\right).$$

Overall, the mean-based preconditioner is an efficient preconditioner for many reasons. First, it is an optimal preconditioner. Thus, the number of PCG iterations does not depend on any discretisation parameters in the system. More importantly, the spectral bounds show that the eigenvalues of the preconditioned system $P_0^{-1}A$ are packed around one if τ is small. That results in reducing the number of CG iterations. Moreover, the

computational cost of solving systems with P_0 is not high. Nevertheless, P_0 uses only very limited information. Hence, if the remainder terms, i.e., $A - P_0$, are significantly large, then P_0 may be unsuitable.

4.2.2 Kronecker Product Preconditioners

The main disadvantage of mean-based preconditioners is omitting information about both the matrices G_m and the matrices K_m for all $m = 1, 2, \ldots, M$ by using only the first term of the Galerkin matrix A. The Kronecker product preconditioner introduced in [125] is constructed by using all the matrices G_m and also K_m. The main idea of the Kronecker product preconditioner is to find the best approximation of the matrix A by preserving the Kronecker product form. That is, suppose the positive definite matrices A and B are given. We can then find a positive definite matrix C to minimise $\|A - C \otimes B\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. For the Kronecker product preconditioner denoted by $P_\otimes$, the matrix B is chosen to be K_0, and $P_\otimes$ is defined by

$$P_\otimes = G \otimes K_0,$$

where

$$G = \sum_{m=0}^{M} \frac{\mathrm{tr}(K_m^T K_0)}{\mathrm{tr}(K_0^T K_0)} G_m,$$

and $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$ for $A = (a_{ij})_{n\times n} \in \mathbb{R}^{n\times n}$.

Note that, as a result of symmetry and positivity of G, $P_\otimes$ is symmetric and positive definite. Hence, the CG method can be used as a solver for the preconditioned system with preconditioner $P_\otimes$. Furthermore, the author of [125] also mentions the relations between the Kronecker product preconditioner $P_\otimes$ and the mean-based preconditioner P_0 in 2 ways. Firstly, if $P_\otimes$ is expanded, it can be seen that the matrix G is a linear combination of all the Galerkin matrices G_m for $m = 0, 1, 2, \ldots, M$ whereas P_0 is formed

by using only $G_0 = I_N$. Hence, we can view P_0 as an approximation of $P_\otimes$ as follows,

$$P_\otimes = G \otimes K_0 = I_{N_{\boldsymbol{y}}} \otimes K_0 + \sum_{m=1}^{M} \frac{\operatorname{tr}(K_m^T K_0)}{\operatorname{tr}(K_0^T K_0)} G_m \otimes K_0.$$

Secondly, by a property of Kronecker products, P_0 can be seen as a factor of $P_\otimes$:

$$P_\otimes = G \otimes K_0 = (G \otimes I_{N_{\boldsymbol{x}}}) \left(I_{N_{\boldsymbol{y}}} \otimes K_0\right) = (G \otimes I_{N_{\boldsymbol{x}}})\, P_0. \tag{4.9}$$

Moreover, the following lemma shows that $P_\otimes$ is the best approximation of the stochastic Galerkin matrix A under some additional conditions.

Lemma 4.3 ([125, Lemma 5.2]). *Let a be a random diffusion coefficient in the form (4.1). Assume that a_m are constant for each $m = 0, 1, 2, \ldots, M$. Then, the Kronecker preconditioner $P_\otimes$ is identical to the Galerkin matrix A, that is $P_\otimes = A$.*

Therefore, if the conditions of Lemma 4.3 hold for the diffusion coefficient function a, then we can obtain the solution in just one iteration of the CG method. Most importantly, an eigenvalue analysis is available.

Theorem 4.4 ([125, Corrollary 5.4]). *Let a be a random diffusion coefficient in the form (4.1). Suppose that the random variables are pairwise uncorrelated with even probability density functions. Then, the following eigenvalue bounds hold*

$$\begin{aligned} \lambda_{\min}(P_\otimes^{-1} A) &\geq \frac{1}{1+\tau_2} - \frac{\tau_1}{1-\tau_2}, \quad 0 < \tau_2 < 1, \\ \lambda_{\max}(P_\otimes^{-1} A) &\leq \frac{1+\tau_1}{1-\tau_2}, \end{aligned}$$

where

$$\begin{aligned} \tau_1 &= \frac{1}{a_0^{\min}} \sum_{m=1}^{M} \bar{\mu}_{k+1}^{(m)} \left\| a_m \right\|_{L^\infty(D)}, \\ \tau_2 &= \sum_{m=1}^{M} \bar{\mu}_{k+1}^{(m)} \frac{\left\| K_m \right\|_F}{\left\| K_0 \right\|_F}, \end{aligned}$$

$\bar{\mu}_{k+1}^{(m)}$ *denotes the largest root of the* $(\cdot,\cdot)_{\rho_m}$*-orthogonal polynomial* P_{k+1}.

Unfortunately, the theoretical eigenvalue bounds of $P_{\otimes}^{-1}A$ are not sufficiently sharp, therefore, they cannot reflect or predict the performance of $P_{\otimes}$. Moreover, they are worse than those established for the mean-based preconditioned Galerkin matrix in Theorem 4.2. However, according to experiments in [125, section 6], the Kronecker product preconditioner reduces the number of iterations when compared to the mean-based preconditioner in many test problems. This is because the spectrum of the preconditioned matrix $P_{\otimes}^{-1}A$ clusters around one more than that of $P_0^{-1}A$. Moreover, the setup time for the preconditioner $P_{\otimes}$ depends on the dimension of S_k^M, i.e., $N_{\boldsymbol{y}}$, whereas the construction time of P_0 remains virtually constant. However, the setup time for the preconditioner $P_{\otimes}$ is negligible compared to the time taken by the iterative solver.

In terms of computational cost of the action of $P_{\otimes}^{-1}$, it can be seen that the solution of a linear system with the coefficient matrix $P_{\otimes}$ is more expensive than solving with the preconditioner P_0 due to the relation between P_0 and $P_{\otimes}$ in (4.9). For the action of $P_{\otimes}^{-1}$ on a vector $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}}$, we start by solving $N_{\boldsymbol{x}}$ linear systems with the system matrix G. For example, we solve the linear system $(G \otimes I_{N_x})\,\mathbf{x} = \mathbf{v}$ by splitting into subsystems

$$G\mathbf{x}_i = \mathbf{v}_i,$$

where $\mathbf{v}_i = \mathbf{v}(i : N_{\boldsymbol{x}} : N_{\boldsymbol{x}}N_{\boldsymbol{y}})$ and $\mathbf{x}_i = \mathbf{x}(i : N_{\boldsymbol{x}} : N_{\boldsymbol{x}}N_{\boldsymbol{y}})$ for $i = 1, 2, \ldots, N_{\boldsymbol{x}}$. We may assign each linear system to a processor. We can then obtain $P_{\otimes}^{-1}\mathbf{v}$ by solving a linear system with the system matrix P_0 which takes $N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{b}\right)$ operations. To conclude, the action of $P_{\otimes}^{-1}$ has the following complexities

$$\mathcal{F}\ell\left(P_{\otimes}^{-1}\mathbf{v}\right) = N_{\boldsymbol{x}}\mathcal{F}\ell\left(G^{-1}\mathbf{d}\right) + N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{b}\right)$$

and

$$\mathcal{F}\ell p\left(P_{\otimes}^{-1}\mathbf{v}\right) = \mathcal{F}\ell\left(G^{-1}\mathbf{d}\right) + \mathcal{F}\ell\left(K_0^{-1}\mathbf{b}\right).$$

Since G is a linear combination of G_m for $m = 0, 1, 2, \ldots, M$, G has the same pattern as the stochastic Galerkin matrix A. Consequently, there are $2M + 1$ nonzero entries per row. Thus, linear systems with the matrix G can be solved in $O\left((2M+1)^2\right)$ operations by assuming that Cholesky factorisation of G is provided (see [125, Section 5.5]).

4.3 Truncation Preconditioners

Truncation preconditioners are a generalisation of the mean-based preconditioner. We start by finding a bilinear form $\tilde{B}$, which is equivalent to the bilinear form B in (2.12). Thus, a preconditioner P which is defined via the bilinear $\tilde{B}$ is optimal. That is the preconditioned Conjugate Gradient converges in a number of iterations independent of the size of the coefficient matrix A. To begin with, we will show that conditions (4.2) and (4.3) guarantee that a truncated coefficient of $a(\boldsymbol{x}, \boldsymbol{y})$ is positive and uniformly bounded away from zero. This allows us to design an optimal preconditioner via a bilinear form which is defined via the truncated coefficient of $a(\boldsymbol{x}, \boldsymbol{y})$.

As was done for the random field a, we define a quantity related to the truncated coefficient a by

$$\tau_0 \coloneqq 0, \qquad \tau_r \coloneqq \frac{1}{a_0^{\min}} \left\| \sum_{m=1}^{r} |a_m| \right\|_{L^\infty(D)}, \quad r \in \mathbb{N}. \tag{4.10}$$

Note that $(\tau_r)_{r \in \mathbb{N}_0}$ is an increasing sequence and bounded by 1 (see Figure 4.1). That is

$$0 = \tau_0 \leq \tau_1 \leq \cdots \leq \tau_r \leq \tau_{r+1} \leq \cdots \leq \tau < 1.$$

We will use this quantity to find a bound for the truncated random field a.

Lemma 4.5. *Let a be a parametric diffusion coefficient in the form (4.1) which satisfies conditions (4.2) and (4.3). Define a truncated expansion of the random field a by a_r :*

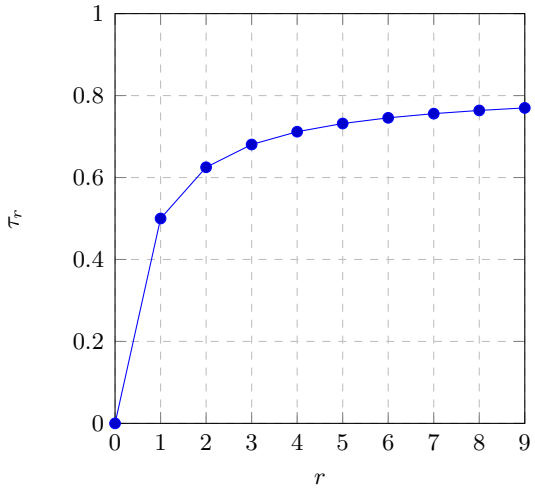


Figure 4.1: τ_r. is a non-decreasing sequence bounded by $\tau < 1$.

$D \times \mathbf{\Gamma} \to \mathbb{R}$

$$a_r(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{m=1}^{r} a_m(\boldsymbol{x}) y_m. \tag{4.11}$$

Then, for any $r \in \mathbb{N}_0$, a_r is positive and bounded in $D \times \mathbf{\Gamma}$.

Proof. It is obvious that a_0 is positive and bounded, by condition (4.2). Let $r \in \mathbb{N}$, since $|y_m| \leq 1$ for all $m \in \mathbb{N}$, we have

$$|a_r(\boldsymbol{x}, \boldsymbol{y}) - a_0(\boldsymbol{x})| = \left| \sum_{m=1}^{r} a_m(\boldsymbol{x}) y_m \right| \leq \sum_{m=1}^{r} |a_m(\boldsymbol{x})| \leq \left\| \sum_{m=1}^{r} |a_m| \right\|_{L^\infty(D)} = a_0^{\min} \tau_r.$$

Therefore, we get

$$a_0(\boldsymbol{x}) - a_0^{\min} \tau_r \leq a_r(\boldsymbol{x}, \boldsymbol{y}) \leq a_0(\boldsymbol{x}) + a_0^{\min} \tau_r.$$

By condition (4.2) on a_0,

$$a_r^{\min} := a_0^{\min} - a_0^{\min} \tau_r \leq a_r(\boldsymbol{x}, \boldsymbol{y}) \leq a_0^{\max} + a_0^{\min} \tau_r =: a_r^{\max}. \tag{4.12}$$

Since $\tau_r < 1$, then $a_r^{\min} = a_0^{\min}(1 - \tau_r) > 0$.

□

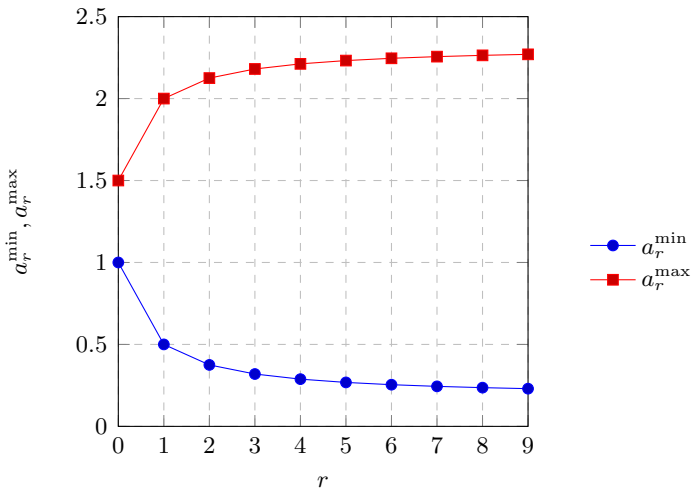


Figure 4.2: $a_r^{\min}$ is a non-increasing sequence whereas $a_r^{\max}$ is a non-decreasing sequence.

We now use Lemma 4.5 to define a family of bilinear forms equivalent to the bilinear form B.

Theorem 4.6. *Let a_r be a function in (4.11). Define the bilinear form $B_r : V \times V \to \mathbb{R}$ by*

$$B_r(u,v) = \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_r(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}. \tag{4.13}$$

Then, the bilinear form B_r is equivalent to the bilinear form B in (2.12) for all $r \in \mathbb{N}_0$. To be precise, there exist positive numbers θ_r and Θ_r such that

$$\theta_r B_r(v,v) \le B(v,v) \le \Theta_r B_r(v,v), \qquad \text{for all } v \in V,$$

where

$$\theta_r = \frac{1}{1+\varepsilon_r}, \quad \Theta_r = 1 + \varepsilon_r', \tag{4.14}$$

with

$$\varepsilon_r = \frac{1}{a_{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)} \text{ and } \varepsilon_r' = \frac{1}{a_r^{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)}.$$

Proof. Consider

$$\begin{aligned}
|B(v,v) - B_r(v,v)| &= \left| \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \left(\sum_{m=r+1}^{\infty} a_m y_m \right) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \right| \\
&\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \left| \sum_{m=r+1}^{\infty} a_m y_m \right| \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\
&\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \left(\sum_{m=r+1}^{\infty} |a_m| \right) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}.
\end{aligned}$$

For the lower bound, we have

$$\begin{aligned}
B_r(v,v) - B(v,v) &\leq |B(v,v) - B_r(v,v)| \\
&\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \frac{a(\boldsymbol{x},\boldsymbol{y})}{a_{\min}} \left(\sum_{m=r+1}^{\infty} |a_m| \right) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\
&\leq \frac{1}{a_{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)} \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a(\boldsymbol{x},\boldsymbol{y}) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\
&\leq \frac{1}{a_{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)} B(v,v).
\end{aligned}$$

Hence,

$$\frac{1}{1+\varepsilon_r} B_r(v,v) \leq B(v,v)$$

where

$$\varepsilon_r = \frac{1}{a_{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)}.$$

Next, we find an upper bound by considering

$$\begin{aligned}
B(v,v) - B_r(v,v) &\leq |B(v,v) - B_r(v,v)| \\
&\leq \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D \frac{a_r(\boldsymbol{x},\boldsymbol{y})}{a_r^{\min}} \left(\sum_{m=r+1}^{\infty} |a_m| \right) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\
&\leq \frac{1}{a_r^{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)} \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_r(\boldsymbol{x},\boldsymbol{y}) \nabla v(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\
&\leq \frac{1}{a_r^{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)} B_r(v,v).
\end{aligned}$$

Therefore, we get

$$B(v,v) \leq (1 + \varepsilon_r')\, B_r(v,v)$$

where

$$\varepsilon_r' = \frac{1}{a_r^{\min}} \left\| \sum_{m=r+1}^{\infty} |a_m| \right\|_{L^\infty(D)}.$$

□

Remark. Theorem 4.6 generalises Theorem 4.2 and the result in [21, eq. (2.5)]. That is, if we set $r = 0$, we get the eigenvalue bounds for the mean-based preconditioner.

We can see from the above theorem that ε_r and ε_r' tend to 0 as r increases. As a result, θ_r and Θ_r approach 1. Hence, the constants of equivalence are tighter around one as r increases. Most importantly, these constants do not depend on any discretisation parameters such as mesh size or the degree of the space of complete polynomials. We now know that the bilinear forms B and B_r are equivalent. Hence, we utilise the bilinear form B_r to generate a class of optimal preconditioners. Specifically, let P_r be induced by the bilinear form B_r. By Proposition 3.8, the eigenvalue bounds for the preconditioned system $P_r^{-1}A$ are

$$\Lambda\left(P_r^{-1}A\right) \subseteq [\theta_r, \Theta_r], \tag{4.15}$$

where θ_r and Θ_r are defined in (4.14).

Additionally, we obtain eigenvalue bounds for $P_0^{-1}P_r$ in the following corollary.

Corollary 4.7. *Let a_r be a truncated random diffusion coefficient in the form (4.11). Let P_r and P_0 be induced by the bilinear form B_r in (4.13) and B_0 in (4.7), respectively. Then*

$$\Lambda\left(P_0^{-1}P_r\right) \subseteq [1-\tau_r, 1+\tau_r],$$

where τ_r is defined in (4.10).

Proof. By Lemma 4.5, a_r is positive and bounded. If we set $a_m = 0$ for $m > r$, the result of Theorem 4.2 will hold with τ replaced by τ_r

$$(1-\tau_r)\, B_0(v,v) \leq B_r(v,v) \leq (1+\tau_r)\, B_0(v,v).$$

Since the bilinear forms B_r and B_0 induce the preconditioners P_r and P_0, respectively, by Proposition 3.8, we have

$$1-\tau_r \leq \lambda\left(P_0^{-1}P_r\right) \leq 1+\tau_r.$$

□

However, to construct a preconditioner P_r, there are a few issues to consider such as the speed of the convergence of the preconditioned system and the complexity of the action of the inverse of the preconditioner P_r on a vector.

4.4 Modified Truncation Preconditioners

To design an efficient preconditioner, there are some features that we have to take into account. The preconditioner should be optimal and also capture the main features of the coefficient matrix. Additionally, the complexity of the action of its inverse on a vector should not be expensive.

In the previous section, we have seen that the truncation preconditioners P_r induced by the bilinear B_r are optimal. Next, in order to keep the key features of the coefficient matrix A, we need to ensure that the parametric function a is properly approximated.

Because the coefficient function a in (4.1) is defined via a KL expansion, we sort a_m by their magnitudes. That is

$$\|a_1\|_{L^\infty(D)} \geq \|a_2\|_{L^\infty(D)} \geq \|a_3\|_{L^\infty(D)} \geq \cdots .$$

As a result, a is well represented by a_r and P_r captures the most significant $r+1$ terms of the matrix A.

Lastly, we replace the preconditioner P_r by its symmetric block Gauss-Seidel (SBGS) approximation:

$$\tilde{P}_r = \left(G_0 \otimes K_0 + \sum_{m=1}^{r} L_m \otimes K_m \right) (G_0 \otimes K_0)^{-1} \left(G_0 \otimes K_0 + \sum_{m=1}^{r} L_m^T \otimes K_m \right),$$

where L_m is the strictly lower triangular part of the matrix G_m. By the structure of the block-triangular and block-diagonal matrices in $\tilde{P}_r$, the multiplication of $\tilde{P}_r^{-1}$ with a vector is acceptable in terms of computational cost.

Remark. With a certain permutation, the truncation preconditioner P_r can be in the form of a block-diagonal matrix. However, the number of blocks reduces when r increases. In the case $r = 1$, we can employ a specific permutation so that P_1 is a block-diagonal matrix where each block is a block-tridiagonal matrix, as shown in Figure 4.3. This structure of P_1 could be beneficial if an efficient block-tridiagonal solver is provided.

4.4.1 Computational Costs

As we mentioned in the previous chapter, the stochastic Galerkin matrix A in (4.5) is symmetric and positive definite. Then PCG is a suitable linear solver. One iteration of PCG requires one matrix-vector multiplication and solving a linear system with a preconditioner as a coefficient matrix. Hence, to compare the efficiency with other preconditioners, we need to analyse its complexity for solving the linear system with a preconditioner.

Since the modified truncation preconditioners are decomposed into two block-triangular

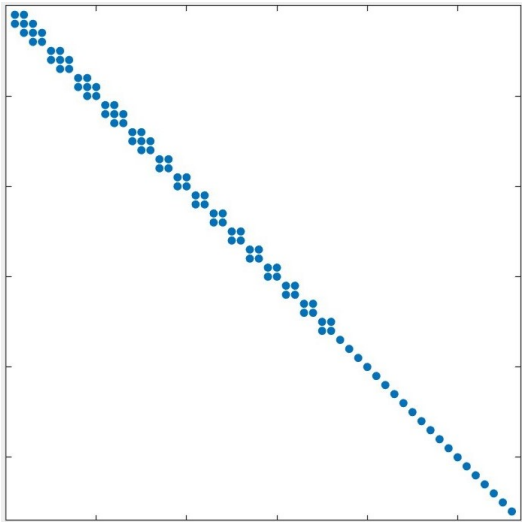


Figure 4.3: The sparsity pattern of the preconditioner P_1 when $M = 5$ and $k = 3$ based on [89, Theorem 9.62].

matrices and one block-diagonal matrix, in order to solve $\tilde{P}_r \mathbf{z} = \mathbf{r}$, we need to solve two block-triangular linear systems as follows.

1. Solve for $\mathbf{z}'$ by forward substitution from

$$\left(G_0 \otimes K_0 + \sum_{m=1}^{r} L_m \otimes K_m \right) \mathbf{z}' = \mathbf{r}.$$

2. Solve for $\mathbf{z}$ by backward substitution from

$$\left(G_0 \otimes K_0 + \sum_{m=1}^{r} L_m^T \otimes K_m \right) \mathbf{z} = \left(G_0 \otimes K_0 \right) \mathbf{z}'.$$

The computational cost for solving such a linear system involves two main operations, i.e., matrix-vector multiplication and solving a linear system with stiffness matrix K_0. For the former, we know that if A is a sparse matrix and $\mathbf{v}$ is a vector, then $\mathcal{F}\ell\left(A\mathbf{v}\right) = \mathrm{nnz}(A)$. In the latter case, we leave it to the reader to decide on a suitable approach such as multigrid or domain decomposition. Each method has a different key feature. For example, a multigrid method provides optimality or a domain decomposition method is capable of parallelism.

For the first step, forward substitution requires solving $N_{\boldsymbol{y}}$ linear systems with stiffness

matrix K_0 and $rN_{\boldsymbol{y}}$ matrix-vector multiplications because P_r have at most $2r+1$ non-zero block matrices per row. Therefore, the complexity is $rN_{\boldsymbol{y}}\text{nnz}(K_0) + N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right)$. Similarly, for backward substitution, it takes another $rN_{\boldsymbol{y}}\text{nnz}(K_0) + N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right)$ flops plus the complexity for multiplication on the right hand side of the equation which requires $N_{\boldsymbol{y}}\text{nnz}(K_0)$. As a result,

$$\mathcal{F}\ell\left(\tilde{P}_r^{-1}\mathbf{r}\right) \approx (2r+1)N_{\boldsymbol{y}}\text{nnz}(K_0) + 2N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right).$$

Generally, the structure of $\tilde{P}_r$ does not facilitate parallelism. However, using a certain permutation technique, the preconditioner P_r can be a block-diagonal matrix, although each block may have a different size. Furthermore, the parallel computing suitability of P_r deteriorates when r increases. Thus, we interested in cases when r is small.

Case $r = 0$: In this case, P_0 is a mean-based preconditioner which is a block-diagonal matrix. Recall that

$$\mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right) = N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right), \quad \mathcal{F}\ell p\left(P_0^{-1}\mathbf{r}\right) = \mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right).$$

Case $r = 1$: As mentioned earlier, by the permutation technique in [89, Theorem 9.62], we have

$$G_1 = \begin{bmatrix} T_{k+1}^1 & & & & & & \\ & T_k^1 & & & & & \\ & & \ddots & & & & \\ & & & T_k^1 & & & \\ & & & & \ddots & & \\ & & & & & T_1^1 & \\ & & & & & & \ddots \\ & & & & & & & T_1^1 \end{bmatrix}$$

where T_j^m is a $j \times j$ tridiagonal matrix defined in (2.17) for $j = 1, ..., k+1$ and $T_1^m = 0$. Hence, $P_1 = G_0 \otimes K_0 + G_1 \otimes K_1$ is a block-diagonal matrix whose block at

the main diagonal is a block-tridiagonal, and the complexity for $\tilde{P}_1^{-1}\mathbf{r}$ is

$$\mathcal{F}\ell\left(\tilde{P}_1^{-1}\mathbf{r}\right) \approx \left(2\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 3\text{nnz}(K_0)\right)\sum_{j=2}^{k+1} n_j j + n_1\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right),$$

where, if $M \geq 2$,

$$n_j = \begin{pmatrix} k+M-j-1 \\ M-2 \end{pmatrix}.$$

Note that if $M = 1$, then $n_j = 0$ for $j = 1, 2, ..., k$ but $n_{k+1} = 1$. Furthermore, the parallel complexity depends on the largest block which is T_{k+1}. Therefore,

$$\mathcal{F}\ell p\left(\tilde{P}_1^{-1}\mathbf{r}\right) \approx (k+1)\left(2\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 3\text{nnz}(K_0)\right).$$

Note that,

$$\sum_{j=1}^{k+1} n_j = \sum_{j=1}^{k+1} \begin{pmatrix} k+M-j-1 \\ M-2 \end{pmatrix} = \sum_{j=M-2}^{k+M-2} \begin{pmatrix} j \\ M-2 \end{pmatrix}.$$

By the Hockey-stick identity (see [75])

$$\sum_{i=k}^{n} \begin{pmatrix} i \\ k \end{pmatrix} = \begin{pmatrix} n+1 \\ k+1 \end{pmatrix},$$

we have that

$$\sum_{j=1}^{k+1} n_j = \begin{pmatrix} k+M-1 \\ M-1 \end{pmatrix} = \frac{k}{k+M} N_{\boldsymbol{y}}.$$

Therefore, we require at most $\frac{k}{k+M} N_{\boldsymbol{y}}$ processors on a parallel machine.

4.5 Analysis of Modified Truncation Preconditioners

In this section, we aim to derive spectral bounds for $\tilde{P}_r^{-1}A$ and show that $\tilde{P}_r$ is an optimal preconditioner. We know that P_r is optimal for the stochastic Galerkin matrix A. If P_r and $\tilde{P}_r$ are spectrally equivalent, so are $\tilde{P}_r$ and A by transitivity of the equivalence relation. For convenience, we let S_r and D_0 be the strictly lower block-triangular part and the block-diagonal part of the matrix A. That is

$$S_r = \sum_{m=1}^{r} L_m \otimes K_m, \quad D_0 = G_0 \otimes K_0,$$

so that

$$P_r = D_0 + S_r + S_r^T.$$

Additionally, we assume that the ordering of multi-indices in the index set $\mathbb{I}_k^M$ by lexicographic, anti-lexicographic, ascending or descending ordering. This ordering results in the matrices L_m have at most one nonzero entry per row and per column. Recall that $\tilde{P}_r$ is the SBGS approximation of P_r, in particular,

$$\tilde{P}_r = (D_0 + S_r)\, D_0^{-1} \left(D_0 + S_r^T\right) = P_r + S_r D_0^{-1} S_r^T.$$

Let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}} N_{\boldsymbol{y}}} \setminus \{\mathbf{0}\}$. Consider

$$\frac{\mathbf{v}^T \tilde{P}_r \mathbf{v}}{\mathbf{v}^T P_r \mathbf{v}} = 1 + \frac{\mathbf{v}^T S_r D_0^{-1} S_r^T \mathbf{v}}{\mathbf{v}^T \left(D_0 + S_r + S_r^T\right) \mathbf{v}}.$$

The minimum eigenvalue of $P_r^{-1}\tilde{P}_r$ is

$$\lambda_{\min}\left(P_r^{-1}\tilde{P}_r\right) = \min_{\mathbf{v}\neq\mathbf{0}} \frac{\mathbf{v}^T \tilde{P}_r \mathbf{v}}{\mathbf{v}^T P_r \mathbf{v}} = 1. \tag{4.16}$$

The last equality is obtained from the fact that $S_r D_0^{-1} S_r^T$ is a positive semi-definite matrix. Next, let

$$\tilde{S}_r = D_0^{-\frac{1}{2}} S_r D_0^{-\frac{1}{2}} \tag{4.17}$$

and apply the change of variable by setting $\mathbf{w} = D_0^{\frac{1}{2}}\mathbf{v}$ to get

$$\begin{aligned}\max_{\mathbf{v}\neq\mathbf{0}} \frac{\mathbf{v}^T S_r D_0^{-1} S_r^T \mathbf{v}}{\mathbf{v}^T \left(D_0 + S_r + S_r^T\right)\mathbf{v}} &= \max_{\mathbf{w}\neq\mathbf{0}} \frac{\mathbf{w}^T \tilde{S}_r \tilde{S}_r^T \mathbf{w}}{\mathbf{w}^T \left(I + \tilde{S}_r + \tilde{S}_r^T\right)\mathbf{w}} \\ &\leq \max_{\mathbf{w}\neq\mathbf{0}} \frac{\mathbf{w}^T \tilde{S}_r \tilde{S}_r^T \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \cdot \max_{\mathbf{w}\neq\mathbf{0}} \frac{\mathbf{w}^T \mathbf{w}}{\mathbf{w}^T \left(I + \tilde{S}_r + \tilde{S}_r^T\right)\mathbf{w}} \\ &= \frac{\sigma_{\max}^2\left(\tilde{S}_r\right)}{\lambda_{\min}\left(I + \tilde{S}_r + \tilde{S}_r^T\right)},\end{aligned} \tag{4.18}$$

where $\sigma_{\max}(A)$ is the largest singular value of A.

Thus, in order to obtain the upper bound for the eigenvalues of $P_r^{-1}\tilde{P}_r$, we need an upper bound for $\sigma_{\max}\left(\tilde{S}_r\right)$ and a lower bound for $\lambda_{\min}\left(I + \tilde{S}_r + \tilde{S}_r^T\right)$; these are derived in the following lemma.

Lemma 4.8. *Let $\tilde{S}_r$ be defined by (4.17). Then,*

$$\lambda_{\min}\left(I + \tilde{S}_r + \tilde{S}_r^T\right) \geq 1 - \tau_r,$$

and

$$\sigma_{\max}\left(\tilde{S}_r\right) \leq \frac{1}{a_0^{\min}} \sum_{m=1}^{r} \|a_m\|_{L^\infty(D)},$$

where $a_r^{\max}$ and $a_r^{\min}$ are defined by (4.12).

Proof. Since $I + \tilde{S}_r + \tilde{S}_r^T = D^{-\frac{1}{2}} P_r D^{-\frac{1}{2}}$, by Corollary 4.7, then we get

$$\lambda_{\min}\left(D^{-\frac{1}{2}} P_r D^{-\frac{1}{2}}\right) = \lambda_{\min}\left(P_0^{-1} P_r\right) \geq 1 - \tau_r.$$

Next, consider

$$\tilde{S}_r = \left(I_{N_{\boldsymbol{y}}} \otimes K_0^{-\frac{1}{2}}\right)\left(\sum_{m=1}^{r} L_m \otimes K_m\right)\left(I_{N_{\boldsymbol{y}}} \otimes K_0^{-\frac{1}{2}}\right) = \sum_{m=1}^{r} L_m \otimes \tilde{K}_m,$$

where $\tilde{K}_m = K_0^{-\frac{1}{2}} K_m K_0^{-\frac{1}{2}}$. Thus,

$$\sigma_{\max}\left(\tilde{S}_r\right) \leq \sum_{m=1}^{r} \sigma_{\max}\left(L_m\right) \sigma_{\max}\left(\tilde{K}_m\right). \tag{4.19}$$

Since $\sigma_{\max}^2\left(L_m\right) = \lambda_{\max}\left(L_m L_m^T\right)$ and $L_m L_m^T$ is a diagonal matrix because L_m has only one nonzero entry per row and per column by Theorem 2.11, we find

$$\sigma_{\max}\left(L_m\right) = \max_{i,j} \left[G_m\right]_{ij} = \max_{l} c_l^m.$$

By [55, Theorem 1.28], we have $c_l^m \leq 1$ for all $l, m \in \mathbb{N}_0$ because $\Gamma_m = [-1, 1]$ is bounded. Consequently,

$$\sigma_{\max}\left(L_m\right) \leq 1. \tag{4.20}$$

Now, $\sigma_{\max}\left(\tilde{K}_m\right) = \max_i \left|\lambda_i\left(K_0^{-1} K_m\right)\right|$. In order to find a spectral bound for $K_0^{-1} K_m$, we let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}} \backslash \{\mathbf{0}\}$ and set $v(\boldsymbol{x}) = \sum_{i=1}^{N_{\boldsymbol{x}}} \left[\mathbf{v}\right]_i \phi_i(\boldsymbol{x})$. Thus,

$$\left|\frac{\mathbf{v}^T K_m \mathbf{v}}{\mathbf{v}^T K_0 \mathbf{v}}\right| = \frac{\int_D a_m(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) d\boldsymbol{x}}{\int_D a_0(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) d\boldsymbol{x}} \leq \frac{\|a_m\|_{L^\infty(D)}}{a_0^{\min}}.$$

Therefore,

$$\sigma_{\max}\left(K_0^{-1} K_m\right) = \lambda_{\max}\left(K_0^{-1} K_m\right) \leq \frac{\|a_m\|_{L^\infty(D)}}{a_0^{\min}}. \tag{4.21}$$

Combine the results from (4.19), (4.20) and (4.21) to finish the proof.

□

We combine all the results from (4.16) and (4.18) with Lemma 4.8 to obtain spectral bounds for $P_r^{-1} \tilde{P}_r$ in the following theorem.

Theorem 4.9. *Let P_r be a truncation preconditioner and $\tilde{P}_r$ be the symmetric block Gauss-Seidel approximation of P_r. Then, the spectral bounds for the generalized eigenvalue problem $\tilde{P}_r \mathbf{v} = \lambda P_r \mathbf{v}$ are*

$$\Lambda(P_r^{-1} \tilde{P}_r) \subseteq [1, 1 + \delta_r],$$

where

$$\delta_r := \frac{1}{1-\tau_r}\left(\frac{1}{a_0^{\min}}\sum_{m=1}^{r}\|a_m\|_{L^\infty(D)}\right)^2. \tag{4.22}$$

Theorem 4.9 shows the efficiency of the modified truncation preconditioner $\tilde{P}_r$ compared to the original truncation preconditioner. Moreover, P_r and $\tilde{P}_r$ are spectral equivalent. Recall that A and P_r are also spectral equivalent by (4.15). Finally, we will merge these results and get the following theorem.

Theorem 4.10. *The condition number of $\tilde{P}_r^{-1}A$ is bounded independently of the degree of complete polynomial space, number of random variables and the mesh size. In other words, the preconditioner $\tilde{P}_r$ is spectrally equivalent to the stochastic Galerkin matrix A, i.e.,*

$$\Lambda\left(\tilde{P}_r^{-1}A\right) \subseteq \left[\frac{\theta_r}{1+\delta_r}, \Theta_r\right]$$

where θ_r and Θ_r are the constants defined in (4.14) and δ_r is defined in (4.22).

In conclusion, truncation preconditioners generalise the idea of the mean-based preconditioner. They are designed by the property of equivalent bilinear forms via the truncated coefficient function a. Therefore, they are optimal for the stochastic Galerkin matrix. However, the action of their inverse on a vector is expensive. This leads to modified truncation preconditioners. Thus, the complexity for solving a linear system with the preconditioner as a system matrix is acceptable.

4.6 Numerical Experiments

In this section, we investigate the behaviour of truncation preconditioners and modified truncation preconditioners, i.e., spectrum equivalence and performance, to support the theoretical results obtained earlier. Furthermore, we compare the efficiency of our proposed preconditioners with other preconditioners such as the mean-based preconditioner and the Kronecker product preconditioner. All the experiments in this thesis were imple-

mented by S-IFISS [112], which is a MATLAB package to construct a stochastic Galerkin approximation for PDE with uncertainty.

All test problems in this section solve the model problem in (2.7) where the diffusion coefficient $a(\boldsymbol{x}, \omega)$ is assumed to be

$$a(\boldsymbol{x}, \omega) = a_0(\boldsymbol{x}) + \sum_{m=1}^{\infty} a_m(\boldsymbol{x}) Y_m(\omega)$$

and to satisfy the assumptions (4.2) and (4.3) with $Y_m : \Omega \to [-1, 1]$ independent and uniformly distributed. Thus, the corresponding parametric representation of $a(\boldsymbol{x}, \boldsymbol{y})$ is affine-parametric in the form (4.1) with $y_m \in [-1, 1]$. Moreover, the forcing function f is set to be $f(\boldsymbol{x}) = 1$.

In these experiments, we vary the discretisation parameters in SGFEM. $L^2(\boldsymbol{\Gamma})$ is represented by the space of complete polynomials S_k^M from 1 to 8 random variables ($M = 1, ..., 8$) with the degree 1 to 6 ($k = 1, ..., 6$). We used square elements in D with size h from 2^{-4} to as fine as 2^{-7}. These parameters lead to the dimension of V_{hk}^M, where $\dim V_{hk}^M = N_{\boldsymbol{x}} N_{\boldsymbol{y}}$, as shown in Figure 4.4. Moreover, the dimension of the finite dimensional subspace V_{hk}^M reflects the sizes of linear system arising from SGFEM. Next, PCG is applied to the linear system with initial guess $\mathbf{x}_0 = \mathbf{0}$ and terminates within $tol = 10^{-6}$.

Note that Figure 4.4 is plotted on a semilog scale. Thus, it can be seen that the size of linear system grows rapidly with discretisation parameters.

Example 4.1. In this test problem, we use the problem in [41, Section 11] which represents planar Fourier sine modes in increasing total order. That is $a_0 = 1$ and

$$a_m(\boldsymbol{x}) = \bar{\alpha} m^{-\tilde{\sigma}} \cos(2\pi\beta_1(m)x_1) \cos(2\pi\beta_2(m)x_2), \quad \boldsymbol{x} = (x_1, x_2) \in D,$$

where $\bar{\alpha}$ and $\tilde{\sigma}$ are constants with $\tilde{\sigma} > 1$ and $0 < \bar{\alpha}\zeta(\tilde{\sigma}) < 1$. Here, ζ is the Riemann zeta function. Additionally, β_1 and β_2 are defined by

$$\beta_1(m) = m - \frac{1}{2} k(m) (k(m) + 1), \quad \beta_2(m) = k(m) - \beta_1(m)$$

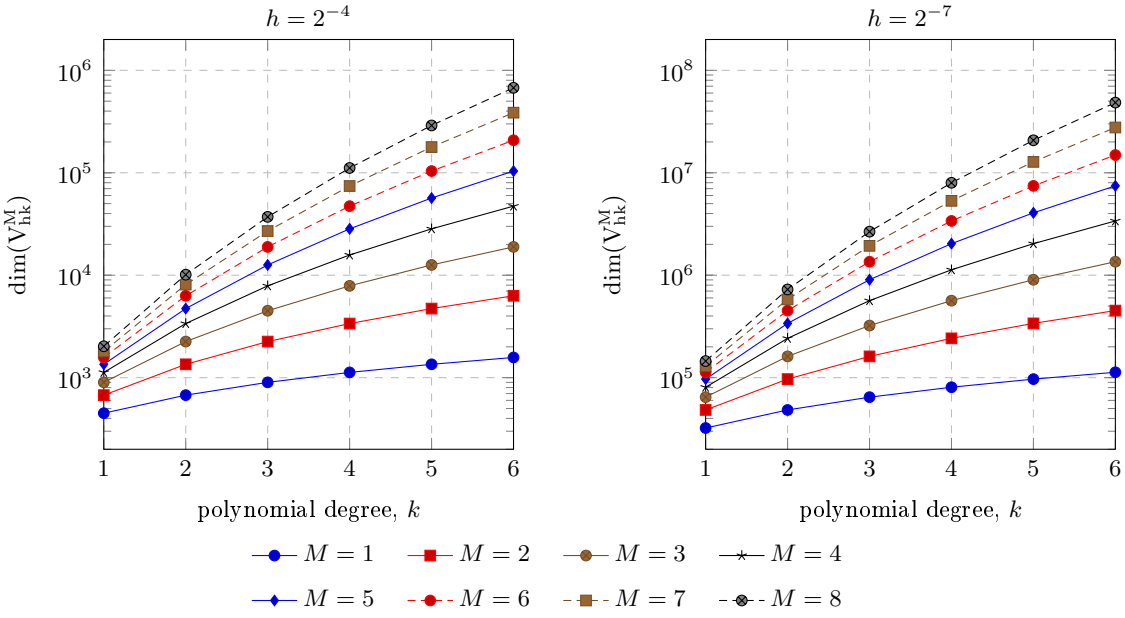


Figure 4.4: The dimension of finite dimensional subspace V_{hk}^M with different discretisation parameters.

m	0	1	2	3	4	5	6
fast decay $\tilde{\sigma} = 4$	1.0000	0.9239	0.0577	0.0114	0.0036	0.0015	0.0007
slow decay $\tilde{\sigma} = 2$	1.0000	0.6079	0.1520	0.0675	0.0380	0.0243	0.0169

Table 4.1: Amplitudes of a_m in fast and slow decay cases.

with $k(m) = \left\lfloor -1/2 + \sqrt{1/4 + 2m} \right\rfloor$. Note that $\sum_{m=1}^{\infty} |a_m| \leq \bar{\alpha} \sum_{m=1}^{\infty} m^{-\tilde{\sigma}} = \bar{\alpha}\zeta(\tilde{\sigma}) < 1$. Thus, the coefficient a satisfies conditions (4.2) and (4.3) with $a_0^{\min} = a_0^{\max} = 1$.

In this experiment, we tested two cases: fast decay and slow decay of amplitude of a_m with $\tilde{\sigma}$ is set to be 4 and 2, respectively. In both cases, we choose $\bar{\alpha}$ such that $\bar{\alpha}\zeta(\tilde{\sigma}) = 0.9999$. Therefore, $\tilde{\sigma} = 4$ and $\bar{\alpha} = 0.9239$ for fast decay whereas $\tilde{\sigma} = 2$ and $\bar{\alpha} = 0.6079$ for slow decay. The magnitude of the coefficient a_m in the case of fast decay drops sharply, whereas the magnitude of coefficient a_m drops slower compared to the case of fast decay.

Table 4.1 illustrates the magnitudes of $\|a_m\|_{L^\infty(D)}$ in both cases. Although the magnitude of a_1 for fast decay is larger than the one for slow decay, the magnitudes of the expansion coefficients a_m in fast decay drop very sharply. For example, $\|a_1\|_{L^\infty(D)} \approx$

$16\,\|a_2\|_{L^\infty(D)}$ in fast decay whereas $\|a_1\|_{L^\infty(D)} \approx 4\,\|a_2\|_{L^\infty(D)}$ in slow decay. Consequently, the magnitude of $\|a_2\|_{L^\infty(D)}$ in fast decay is only one third of the magnitude of $\|a_2\|_{L^\infty(D)}$ in slow decay. This implies that $a_1(\boldsymbol{x}, \boldsymbol{y})$ in the case of fast decay gives a better approximation of a. Since the truncation preconditioner is induced by the bilinear form via the truncated expansion of a, this indicates that the performance of P_1 in fast decay should outweigh P_1 in the case of slow decay. This is confirmed by the results in Table 4.2.

For this experiment, we set the discretisation parameters $h = 2^{-4}$, $M = 8$ and vary k. According to Table 4.2, for a fixed k, we can see that the iteration counts for the fast decay coefficient sharply decease from P_0, which represents the mean-based preconditioner, and then remain stable when r increases. These iteration counts behave in the same way as the magnitudes in Table 4.1. Furthermore, it is obvious that the numbers of iterations in Table 4.2 do not depend on the degree k. Thus, P_r is optimal with respect to k.

Next, we investigate the performance of the modified truncation preconditioners which are the symmetric block Gauss-Seidel approximations of P_r and compare their iteration counts with the mean-based preconditioner and also the Kronecker product preconditioner. The results are shown in Table 4.3. Overall, the modified truncation preconditioners outperform the other preconditioners in term of iteration counts, especially in the fast decay case. The iteration counts by P_1 are only 1/3 to 1/4 of those for the mean-based preconditioner in the fast decay case while the iteration counts for P_1 are about half of those for the mean-based preconditioner in the other case. Recall that the complexity of the mean-based preconditioner is about half of the complexity of the modified truncation preconditioner. Therefore, in both cases, the modified truncation preconditioners

		fast decay							slow decay						
		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_0	P_1	P_2	P_3	P_4	P_5	P_6
$k =$	1	13	4	3	3	2	2	2	10	6	4	4	4	3	3
	2	16	5	4	3	3	2	2	12	7	5	5	4	4	3
	3	21	6	4	3	3	2	2	14	7	6	5	4	4	4
	4	24	6	4	3	3	3	2	15	8	6	5	4	4	4

Table 4.2: The numbers of PCG iterations with the mean-based preconditioner and the truncation preconditioners P_r in the cases of fast decay and slow decay.

		fast decay								slow decay							
		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$	$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$
$k=$	1	12	13	7	6	6	6	6	6	9	10	6	5	5	5	5	5
	2	16	16	8	7	7	7	7	7	12	12	7	6	6	6	5	5
	3	20	21	9	9	8	8	8	8	14	14	8	7	6	6	6	6
	4	24	24	10	9	9	9	9	9	15	15	9	7	7	6	6	6
	5	26	27	11	10	10	10	10	10	16	16	9	7	7	7	6	6
	6	29	29	12	11	11	11	11	11	17	17	10	8	7	7	7	7

Table 4.3: The numbers of PCG iterations with the mean-based preconditioner, the Kronecker product preconditioner and the modified truncation preconditioners in fast decay and slow decay.

are more efficient in general.

However, according to Figure 4.5, PCG with a mean-based preconditioner consumes less time than other preconditioners in general. Moreover, modified truncation preconditioners spend more time before convergence. This probably due to the stiffness matrices are not sufficiently large. Note also that improving the function for solving a linear system with symmetric block Gauss-Seidel by utilising the sparsity pattern of the matrix can significantly speed up the solver's performance.

Additionally, the numbers of iterations by $\tilde{P}_r$ increase as compared to the numbers in Table 4.2. This can be explained by Theorem 4.10 as the eigenvalue bounds of $\tilde{P}_r^{-1}A$ are not as tight as those for $P_r^{-1}A$. The spectral bounds in the case of fast decay by truncation preconditioners and modified truncation preconditioners are shown in Figure 4.4.

Although the eigenvalue bounds in Theorem 4.6 are not sharp, we can see that increasing the number of terms in the truncation preconditioner results in the tighter eigenvalue bounds of $P_r^{-1}A$ around one as described in Theorem 4.6. Moreover, the maximum eigenvalues of the preconditioned system $\tilde{P}_r^{-1}A$ are less than the ones for the system $P_r^{-1}A$ in all tests. These give us the tighter upper eigenvalue bounds, but the lower bounds deteriorate significantly due to the symmetric block Gauss-Seidel approximation of P_r. In addition, for a fixed k, the lower eigenvalue bounds of $\tilde{P}_r^{-1}A$ are improved sharply as compared to those for the mean-based preconditioner but stay the same when increasing

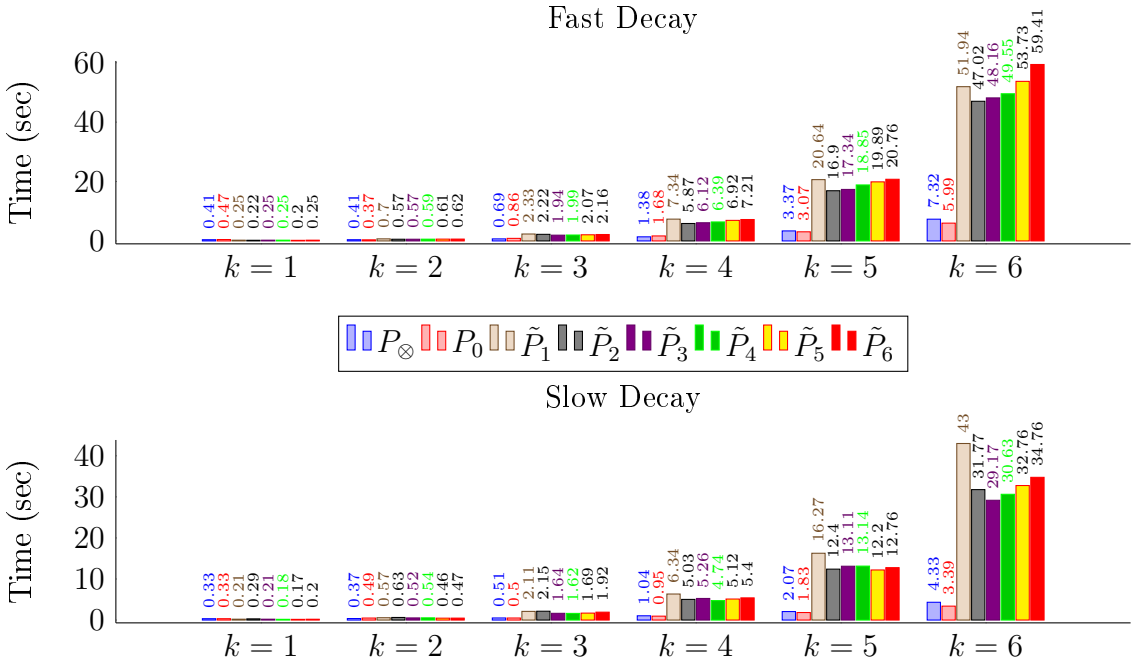


Figure 4.5: The runtimes of PCG iterations (in seconds) with the mean-based preconditioner, the Kronecker product preconditioner and the modified truncation preconditioners in fast decay and slow decay.

r. These behaviours can be explained by Theorem 4.9. That is, the symmetric block Gauss-Seidel approximation of P_r gives an accurate approximation with a small r such as $r = 1, 2$ because δ_r in (4.22) is small. What is more, if r is fixed, τ_r in the fast decay problem is larger than that in the slow decay case. By Theorem 4.9, this implies that $\tilde{P}_r$ in the slow decay case is more accurate with respect to P_r than that in the other case. This is reflected in the iteration counts by $\tilde{P}_r$ in Table 4.3, which moderately increase in the fast decay problem from those in Table 4.2. Based on these facts, we suggest choosing $r = 1$ or $r = 2$ in most cases.

Before ending the discussion of this experiment, we observe the optimality of the modified truncation preconditioners with respect to the mesh size h and the number of random variables M. We range the mesh size h from 2^{-3} to 2^{-7} and $M = 4, 8$ with P_0, $\tilde{P}_1$ and $\tilde{P}_2$. Note that, as discussed above, there is no significant change in the iteration counts for $\tilde{P}_r$ when $r \geq 2$. We have seen that truncation preconditioners are optimal with respect to the degree k, so the degree of the space of complete polynomials is fixed

$P_r^{-1}A$		$k=1$		$k=2$		$k=3$	
		$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$
$r=$	0	0.4718	1.5281	0.2886	1.7113	0.2036	1.7963
	1	0.9607	1.0392	0.9124	1.0875	0.8530	1.1469
	2	0.9922	1.0077	0.9822	1.0177	0.9693	1.0306
	3	0.9974	1.0025	0.9940	1.0059	0.9896	1.0103
	4	0.9989	1.0010	0.9975	1.0024	0.9957	1.0042
	5	0.9994	1.0005	0.9988	1.0011	0.9979	1.0020
	6	0.9997	1.0002	0.9993	1.0006	0.9988	1.0011

$\tilde{P}_r^{-1}A$		$k=1$		$k=2$		$k=3$	
		$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$
$r=$	0	0.4718	1.5281	0.2886	1.7113	0.2036	1.7963
	1	0.7210	1.0339	0.5750	1.0481	0.4675	1.0561
	2	0.7210	1.0066	0.5810	1.0096	0.4803	1.0114
	3	0.7210	1.0022	0.5812	1.0032	0.4810	1.0039
	4	0.7210	1.0009	0.5812	1.0013	0.4810	1.0016
	5	0.7210	1.0004	0.5812	1.0006	0.4810	1.0008
	6	0.7210	1.0002	0.5812	1.0003	0.4810	1.0004

Table 4.4: The extreme eigenvalues of the preconditioned system $P_r^{-1}A$ (the first table) and $\tilde{P}_r^{-1}A$ (the second table) for the fast decay case.

at $k = 3$. The results are shown in Table 4.5. It is obvious that the iteration counts for the modified truncation preconditioners do not depend on M as the iteration counts for $M = 4$ are the same as the results for $M = 8$ in both cases. Moreover, as the mesh is refined, the iteration counts increase very slowly and remain stable when the mesh is sufficiently refined.

As we have seen in the previous experiment, the modified truncation preconditioners are very efficient in the case of the fast decay affine-parametric coefficient. On the other hand, they may struggle with the coefficient whose magnitude of expansion coefficients gradually reduce very slowly.

Example 4.2. Let $D = (-1, 1)^2$. Define a covariance model in geostatistics, for $\boldsymbol{x}, \boldsymbol{x}' \in D$,

$$\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{|x_1 - x_1'|}{l_1} - \frac{|x_2 - x_2'|}{l_2}\right),$$

where l_1 and l_2 are correlation lengths and σ denotes the standard deviation. The eigen-

h	fast decay						slow decay					
	$M=4$			$M=8$			$M=4$			$M=8$		
	P_0	$\tilde{P}_1$	$\tilde{P}_2$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	P_0	$\tilde{P}_1$	$\tilde{P}_2$
2^{-3}	18	8	8	18	8	8	13	7	6	13	7	6
2^{-4}	21	9	9	21	9	9	14	8	7	14	8	7
2^{-5}	23	10	9	23	10	9	14	8	7	14	8	7
2^{-6}	24	10	10	24	10	10	15	8	7	15	8	7
2^{-7}	24	10	10	24	10	10	15	8	7	15	8	7

Table 4.5: The iteration counts for the mean-based preconditioner and the modified truncation preconditioners when h and M are varied.

m	0	1	2	3	4	5	6
$\|a_m\|_{L^\infty(D)}$	1	0.2943	0.1572	0.1572	0.0938	0.0938	0.0840

Table 4.6: The magnitudes of the expansion coefficients a_m in Example 4.2.

pairs $\{(\lambda_m, \varphi_m)\}_{m=1}^{\infty}$ of the Fredholm integral operator in (2.1) are defined in [64, pp 28 - 29]. We set $l_1 = l_2 = 2$ and $\sigma = 0.2$. The mean of the coefficient a is set to be one, i.e., $a_0 = 1$, and the expansion coefficients are defined by

$$a_m(\boldsymbol{x}) = \sigma\sqrt{3\lambda_m}\varphi_m(\boldsymbol{x})y_m, \quad \boldsymbol{x} \in D.$$

In this experiment, we set $h = 2^{-4}$, $M = 8$ and note that $\tau_8 = 0.8054$. The magnitudes of a_m are displayed in Figure 4.6.

We can see that there is a significant difference between a_0 and a_1 ($\|a_0\|_{L^\infty(D)} \approx 3\|a_1\|_{L^\infty(D)}$). In general, the magnitudes of the expansion coefficients gradually decrease except for a_2 and a_4, i.e., $\|a_2\|_{L^\infty(D)} = \|a_3\|_{L^\infty(D)}$ and $\|a_4\|_{L^\infty(D)} = \|a_5\|_{L^\infty(D)}$. This indicates that the importance of each term is not very different. Thus, we expect that the iteration counts for the truncation preconditioners will also reduce very slowly. The results are shown in Table 4.7 and Figure 4.6. The truncation preconditioners and the modified truncation preconditioners perform well in terms of iteration counts. Moreover, as discussed in the previous experiment, there are only a few cases that the iteration counts by modified truncation preconditioners increase from those by truncation preconditioners due to small τ_r. However, the overall performance by the modified truncation precon-

		P_0	P_1	P_2	P_3	P_4	P_5	P_6
$k =$	1	5	6	6	5	5	5	5
	2	8	7	7	6	6	5	5
	3	9	8	8	7	6	6	5
	4	10	9	8	7	7	6	6

		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$
$k =$	1	6	5	6	6	5	5	5	5
	2	7	8	7	7	6	6	5	5
	3	8	9	8	8	7	6	6	6
	4	9	10	9	8	7	7	6	6
	5	10	11	9	9	8	7	6	6
	6	10	11	10	9	8	7	7	6

Table 4.7: The PCG iteration counts for the mean-based preconditioner, and the truncation preconditioners in the first table and the PCG iteration counts for the Kronecker product preconditioner, the mean-based preconditioner and the modified truncation preconditioners in the second table.

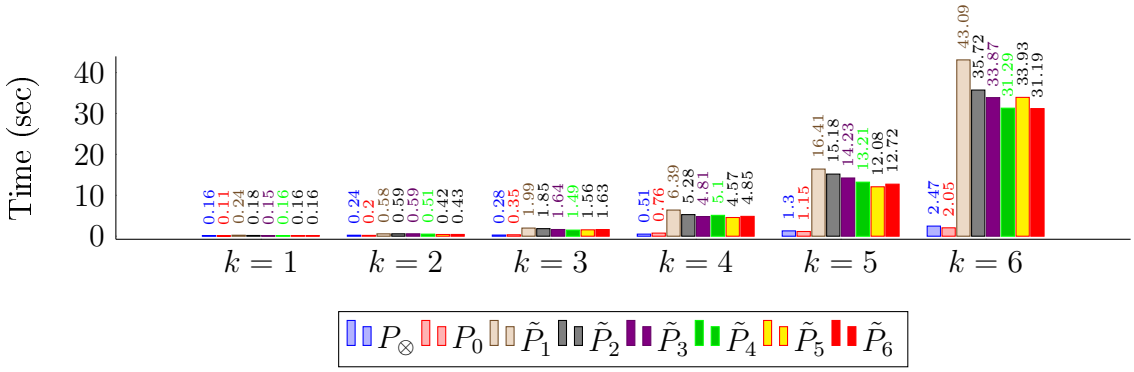


Figure 4.6: The runtimes of PCG iterations (in seconds) with the mean-based preconditioner, the Kronecker product preconditioner and the modified truncation preconditioners.

ditioners is less efficient than that of the mean-based preconditioner or the Kronecker product preconditioner. This is due to the cost per iteration by the modified truncation preconditioner is more expensive than that for the other preconditioners.

Overall, truncation preconditioners and modified truncation preconditioners are efficient preconditioners. They perform very well in term of PCG iteration counts. For the case of fast decay, the experiment showed that modified truncation preconditioners can improve the efficiency of the solver. However, they are still less efficient than existing preconditioners in the case of slow decay. In the next chapter, we will introduce another preconditioning technique to deal with both cases.

CHAPTER 5

DOMAIN DECOMPOSITION METHODS ON PARAMETRIC DOMAIN

Domain decomposition is a preconditioning technique designed to improve the performance of an iterative solver for the case where the matrix A arises from a FEM discretisation. Its goal is to manipulate the pattern of the coefficient matrix A to a 2×2 block matrix such that

$$A = \begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix}, \tag{5.1}$$

where the indices I and F correspond to interior nodes and interface nodes of all subdomains, respectively, and A_{II} is a block diagonal matrix. By the structure of A_{II}, we can gain the benefit of parallelism. However, the main challenge in domain decomposition is to approximate the Schur complement, denoted by

$$S \coloneqq A_{FF} - A_{FI} A_{II}^{-1} A_{IF},$$

so that the action of S^{-1} on a vector is not expensive.

Given a problem posed on a physical domain D, in order to get 2×2 block matrix structure, we partition the spatial domain into subdomains. Reordering the basis functions by subdomain leads to the block diagonal matrix structure of A_{II}, while the basis functions along the boundary of each subdomain are assigned to a so-called interface set.

In this chapter, we introduce a domain decomposition method for the parametric

domain $\mathbf{\Gamma}$ for the case of affine parametric diffusion coefficients. Thus, we want to group the orthogonal basis functions in the space of complete polynomials, so that A is a 2×2 block matrix. To achieve this, since the pattern of the stochastic matrix A is induced by the matrices G_m, the pattern of the matrix G_m is studied in section 5.1. In the following, we will introduce the concept of *parametric mesh* and explain how to create it for the space S_k^M from the multi-indices in $\mathbb{I}_k^M$. Moreover, the connection between parametric mesh and the sparsity pattern of the matrix A is discussed. Next, the concepts of *submesh* and *interface for parametric mesh* are defined. We indicate how the parametric mesh can be partitioned into many non-overlapping submeshes which are separated by the interface. In section 5.2, grouping the basis functions for the space of complete polynomials by submesh leads to the block-diagonal structure of A_{II}. This procedure results in the 2×2 block matrix structure (5.1) of the coefficient matrix A, which motivates the preconditioner matrix structures presented in section 5.3. Our partitioning strategy, namely, the *even-odd partition*, is introduced in section 5.4. We discuss how the actions of the inverse of the Schur complement on a vector are approximated. We combine all the results and present several versions of the preconditioner based on our partitioning strategy, together with the corresponding complexities and the spectral analyses. Finally, numerical experiments comparing the efficiency of all the versions of the even-odd preconditioners with the other preconditioners are presented in section 5.5.

5.1 Parametric Mesh

Recall that $\Gamma_m = [-1, 1]$ and the affine parametric coefficient diffusion a is defined in (4.1) with the assumptions (4.2) and (4.3). The Galerkin projection on the finite dimensional space $S_k^M \otimes X_h$ yields a linear system with coefficient matrix

$$A = \sum_{m=0}^{M} G_m \otimes K_m,$$

where the matrices G_m and K_m are defined in (4.6). Consequently, the stochastic Galerkin matrix A for the affine diffusion coefficient case can be represented in block form with blocks

$$A_{js} \quad = [G_0]_{js} K_0 + \sum_{m=1}^{M} [G_m]_{js} K_m, \qquad j, s = 1, 2, \ldots, N_{\boldsymbol{y}} \ .$$

Define the matrices G and $\bar{G}$ to be

$$G = \sum_{m=1}^{M} G_m, \tag{5.2}$$

and

$$\bar{G} = G_0 + G. \tag{5.3}$$

We would like to study the pattern of the matrix A via the following theorem.

Theorem 5.1. *Let $\bar{G}$ be a $N_{\boldsymbol{y}} \times N_{\boldsymbol{y}}$ matrix as defined in (5.3) and q be a bijection from $\{1, 2, \ldots, N_{\boldsymbol{y}}\}$ to $\mathbb{I}_k^M$. Let $j, s \in \{1, 2, \ldots, N_{\boldsymbol{y}}\}$ and $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_k^M$ such that $\boldsymbol{\beta} = q(j)$ and $\boldsymbol{\beta}' = q(s)$. Then, $\bar{G}_{js} \neq 0$ if and only if one of the following holds*

1. *$\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ are identical, i.e.,*

$$\boldsymbol{\beta} = \boldsymbol{\beta}', \tag{5.4}$$

2. *There exists $m' \in \{1, \ldots, M\}$ such that*

$$|\beta_{m'} - \beta'_{m'}| = 1 \text{ and } \beta_m = \beta'_m \text{ for } m \in \{1, \ldots, M\} \setminus \{m'\}. \tag{5.5}$$

Proof. Suppose $\bar{G}_{js} \neq 0$. Then, $[G_0]_{js} \neq 0$ or $[G]_{js} \neq 0$. If $[G_0]_{js} \neq 0$, this implies that $j = s$ and $\boldsymbol{\beta} = \boldsymbol{\beta}'$ because G_0 is the identity matrix. If $[G]_{js} \neq 0$, we have that there exists $m' \in \{1, \ldots, M\}$ such that $[G_{m'}]_{js} \neq 0$. By Theorem 2.11, we have $|\beta_{m'} - \beta'_{m'}| = 1$ and $\beta_m = \beta'_m$ for all $m \in \{1, \ldots, M\} \setminus \{m'\}$.

Conversely, if $\boldsymbol{\beta} = \boldsymbol{\beta}'$, then

$$[G_0]_{js} = 1 \text{ but } \ [G_m]_{js} = 0 \text{ for all } m \in \{1, 2, \ldots, M\} .$$

Then, $\bar{G}_{js} \neq 0$. Next, if condition (5.5) holds, then $[G_0]_{js} = 0$ and, by Theorem 2.11, there exists $m' \in \{1, 2, \ldots, M\}$ such that $[G_{m'}]_{js} \neq 0$. Moreover, if $[G_{m'}]_{js} \neq 0$, then we have $[G_m]_{js} = 0$ for $m \in \{1, \ldots, M\} \setminus \{m'\}$. We have $\bar{G}_{js} \neq 0$.

□

According to the proof in Theorem 5.1, the coefficient matrix A has K_0 along the main diagonal and $[G_m]_{js} K_m$ for some $m \in \{1, 2, \ldots, M\}$ off the main diagonal. Recall that the sparsity pattern of the matrix A can be investigated via the matrix $\bar{G}$. In order to obtain a 2×2 block matrix structure as in (5.1), we need a rule to find if the block A_{js} is a zero matrix.

Corollary 5.2. *Let A be a $N_{\boldsymbol{y}} \times N_{\boldsymbol{y}}$ block matrix as defined in (4.5) and q be a bijection from $\{1, 2, \ldots, N_{\boldsymbol{y}}\}$ to $\mathbb{I}_k^M$. Let $j, s \in \{1, 2, \ldots, N_{\boldsymbol{y}}\}$ and $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_k^M$ such that $\boldsymbol{\beta} = q(j)$ and $\boldsymbol{\beta}' = q(s)$. Then, $A_{js} \neq 0$ if and only if conditions (5.4) or (5.5) hold.*

Changing the bijection map q causes the change in the pattern of the matrix A except K_0 is always on the main diagonal of A (see Figure 2.3). To have the 2-by-2 block matrix structure in (5.1), a suitable map q is required and the corollary shows the link between the map q and the multi-indices in $\mathbb{I}_k^M$. Hence, any pairs of multi-indices in $\mathbb{I}_k^M$ which yield non-zero blocks in A need to satisfy conditions (5.4) or (5.5). To achieve this, the concept of a parametric mesh for a parametric elliptic PDE problem is introduced via graph theory. To this aim, let us review some necessary definitions.

Definition 5.3. A *graph* G is a pair (V, E) where V is a set and E is a set of subsets with two elements in V. The members in the set V are called *vertices* or *nodes* of the graph G and the members in the set E are called *edges* of the graph G. We call a graph with $V = E = \emptyset$ an *empty graph*.

For simplicity, we write an edge $\{v_1, v_2\} \in E$ as $v_1 v_2$ or $v_2 v_1$.

Definition 5.4. Let $G = (V, E)$ be a graph. For $e \in E$, if $v \in V$ and $v \in e$, we say e is an edge *at* v and v is an *end-vertex* or *end* of the edge e. Moreover, if $v_1, v_2 \in V$ and $v_1 v_2 \in E$, we say v_1 and v_2 are *adjacent* or *neighbours*. The degree $d_G(v)$ of a node

$v \in V$ in the graph G is the number of adjacent nodes of v or the number of edges at v. Additionally, $\delta(G) \coloneqq \min\{d_G(v) \mid v \in V\}$ and $\Delta(G) \coloneqq \max\{d_G(v) \mid v \in V\}$ denote the *minimum degree* and *maximum degree* of the graph G.

Definition 5.5. Let $G = (V, E)$ be a graph where $V = \{v_1, v_2, ..., v_n\}$. The adjacency matrix $A = (a_{ij})_{n\times n}$ of the graph G is defined by

$$a_{ij} = \begin{cases} 1 & , v_i v_j \in E \\ 0 & \text{otherwise.} \end{cases}$$

Definition 5.6. Let $G = (V, E)$ and $G' = (V', E')$ be graphs.

- $G' = (V', E')$ is called a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$.
- The union and intersection of two graphs are defined by $G \cup G' \coloneqq (V \cup V', E \cup E')$ and $G \cap G' \coloneqq (V \cap V', E \cap E')$.
- G and G' are *disjoint* if $G \cap G'$ is an empty graph.

Next, a parametric mesh for a parametric elliptic PDE problem is defined as follows.

Definition 5.7. A parametric mesh $\mathcal{M}$ associated with the set of multi-indices $\mathbb{I}_k^M$ for the affine parametric diffusion coefficient is a graph whose nodes are multi-indices in $\mathbb{I}_k^M$ with edges between any two nodes which satisfy condition (5.5).

Precisely, $\mathcal{M} = (V, E)$ where

$$V = \mathbb{I}_k^M \text{ and } E = \{\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2\} \subseteq V \mid \boldsymbol{\beta}_1 \text{ and } \boldsymbol{\beta}_2 \text{ satisfy condition (5.5).}\}.$$

By the definition of the parametric mesh, for $q(j) = \boldsymbol{\beta}$ and $q(s) = \boldsymbol{\beta}'$ and $j \neq s$, the nodes $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ in the parametric mesh are adjacent if A_{js} is a non-zero block matrix. Consequently, the degree of the node $\boldsymbol{\beta}$ is the number of non-zero block matrices outside the main diagonal at block row or column j of the matrix A. The maximum and minimum

degrees mean the maximum and minimum numbers of the non-zero block matrices per row or column outside the main diagonal of the matrix A, respectively.

Moreover, the matrix G in (5.2) can be viewed as an adjacency matrix of the mesh $\mathcal{M}$. For example, $[G]_{js} = 0$ means the multi-indices $q(j)$ and $q(s)$ are not linked whereas $[G]_{js} \neq 0$ means there is an edge between the nodes $q(j)$ and $q(s)$.

Remark. The maximum degree of the parametric mesh $\mathcal{M}$ is $2M$, since each node in $\mathbb{I}_k^M$ has M entries and each entry may be increased or decreased by one. This is consistent with the fact that the coefficient matrix A has at most $2M+1$ non-zero blocks per block row. Since G has at most $2M$ non-zero entries per row with zeros along the main diagonal. $\bar{G}$ in (5.3) which represents the pattern of the matrix A has at most $2M+1$ non-zero entries per row.

Moreover, the minimum degree of the parametric mesh $\mathcal{M}$ is 1, for example, the node $\boldsymbol{\beta}$ with $|\boldsymbol{\beta}| = k$. Thus, the matrix A has at least two non-zero matrices per row (include the main diagonal).

Example 5.1. The set of all nodes of the parametric mesh for the space S_2^3, i.e., $M = 3$ and $k = 2$, is

$$\begin{aligned}\mathbb{I}_2^3 = \{&(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0),\\ &(1,0,1), (0,1,1), (2,0,0), (0,2,0), (0,0,2)\}.\end{aligned}$$

In this example, we modify the notation slightly and write the nodes as $\alpha_1\alpha_2\alpha_3$ instead of $(\alpha_1, \alpha_2, \alpha_3)$, for convenience. Recall that any two nodes are connected if they satisfy condition (5.5). The following multi-index pairs satisfy this condition:

$$\begin{array}{lll}(000,100) & (100,101) & (010,020)\\ (000,010) & (100,200) & (001,101)\\ (000,001) & (010,110) & (001,011)\\ (100,110) & (010,011) & (001,002)\end{array}$$

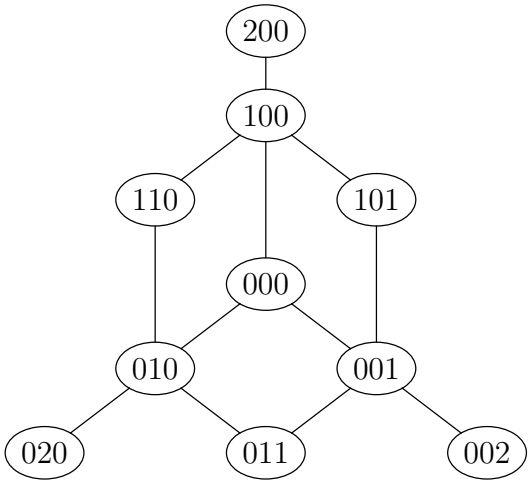


Figure 5.1: The parametric mesh associated with the set $\mathbb{I}_2^3$.

The resulting parametric mesh associated with the set $\mathbb{I}_2^3$ is shown in Figure 5.1.

Submeshes and Interface in a Parametric Mesh

Before introducing the definitions of submeshes and interface of the parametric mesh, we need additional definitions from graph theory.

Definition 5.8. A non-empty graph $P = (V, E)$ is called a path if $V = \{v_0, v_1, ..., v_i\}$ and $E = \{v_0v_1, v_1v_2, ..., v_{i-1}v_i\}$. We denote the path P by $P = v_0v_1...v_i$.

Definition 5.9. A non-empty graph G is called connected if there exists a path in G linking any two vertices in G. Additionally, G is called *disconnected* if G is not connected.

Definition 5.10. Let $G = (V, E)$ be a graph where V and E are sets of nodes and edges, respectively. Let $G' = (V', E')$ be a subgraph of G. G' is called an *induced subgraph* of G if all edges in E whose both ends are in V' are in E'.

The following definitions of submeshes and interface are crucial in order to define a partition for the parametric mesh.

Definition 5.11. A *submesh of the parametric mesh* $\mathcal{M}$ is a connected subgraph of $\mathcal{M}$ and is denoted by $\mathcal{P}_i$. Additionally, $\mathbb{I}_{\mathcal{P}_i}$ denotes the set of all nodes of the submesh $\mathcal{P}_i$ and n_i denotes the cardinal number of the set $\mathbb{I}_{\mathcal{P}_i}$.

Definition 5.12. Let $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_{N_\mathcal{P}}\}$ denote a set of submeshes of a parametric mesh $\mathcal{M}$. The *interface of the parametric mesh* $\mathcal{M}$, denoted by $\mathcal{I}$, is an induced subgraph of the parametric mesh $\mathcal{M}$. The set of nodes of $\mathcal{I}$ is a set of multi-indices which are adjacent to any submesh $\mathcal{P}_i$ and may include nodes that are not connected to any submeshes. $\mathbb{I}_\mathcal{I}$ denotes the set of nodes in the interface $\mathcal{I}$. That is

$$\mathbb{I}_\mathcal{I} = \mathbb{I}_k^M - \bigcup_{i=1}^{N_\mathcal{P}} \mathbb{I}_{\mathcal{P}_i}.$$

Remark. It is possible to partition a parametric mesh and obtain the interface $\mathcal{I}$ of the parametric mesh with no edge.

Note that a submesh of the mesh for the spatial domain D can be associated with a subdomain in D because all the nodes in a spatial mesh are represented in a Cartesian coordinate system. However, in the case of a submesh of the parametric mesh, nodes in the parametric mesh are represented by multi-indices which are not associated with points in parametric domain $\mathbf{\Gamma}$. As a result, a submesh in the parametric mesh $\mathcal{M}$ is not associated with a subdomain in $\mathbf{\Gamma}$.

Suppose the parametric mesh is partitioned into $N_\mathcal{P}$ submeshes. Then, $\mathcal{P}_i \cup \mathcal{P}_j$ is disconnected for $i \neq j$, as the following example illustrates.

Example 5.2. The parametric mesh in Example 5.1 can be partitioned in several ways. On the one hand, it can be partitioned into two submeshes, namely, $\mathcal{P}_1$ and $\mathcal{P}_2$. That is,

$$\mathbb{I}_{\mathcal{P}_1} = \{200, 100, 101, 001\}, \qquad \mathbb{I}_{\mathcal{P}_2} = \{010, 020\}$$

and the set of nodes on the interface is

$$\mathbb{I}_\mathcal{I} = \{110, 000, 011, 002\},$$

as shown in Figure 5.2. In the figure, double circle nodes denote interface nodes and the edges which link between submesh and interface nodes are removed to show how the

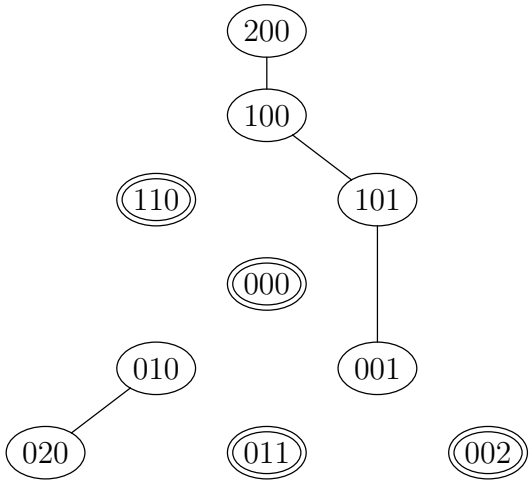


Figure 5.2: Partition of the parametric mesh associated with $\mathbb{I}_2^3$: all the interface nodes are disconnected.

parametric mesh is partitioned.

In addition, this figure shows a partition for the case where interface nodes do not link to each other and all interface nodes are adjacent to a submesh in the original parametric mesh. In this case, we can see that all the nodes in the set $\mathcal{I}$ are connected to at least one submesh before partitioning.

It is also feasible to partition the parametric mesh such that there is a node on the interface not adjacent to any submeshes. Figure 5.3 shows how the parametric mesh is partitioned. That is

$$\mathbb{I}_{\mathcal{P}_1} = \{200, 100, 000, 101\}\,,\ \mathbb{I}_{\mathcal{P}_2} = \{020\} \text{ and } \mathbb{I}_{\mathcal{P}_3} = \{002\}\,,$$

and the set of nodes on the interface is

$$\mathbb{I}_{\mathcal{I}} = \{110, 010, 011, 001\}\,.$$

We can see that the interface node 011 is connected to the nodes 010 and 001 but the node 011 does not connect to any submeshes.

Remark. The original coefficient matrix A in (4.5) corresponds to the case $\mathbb{I}_{\mathcal{I}} = \mathbb{I}_k^M$.

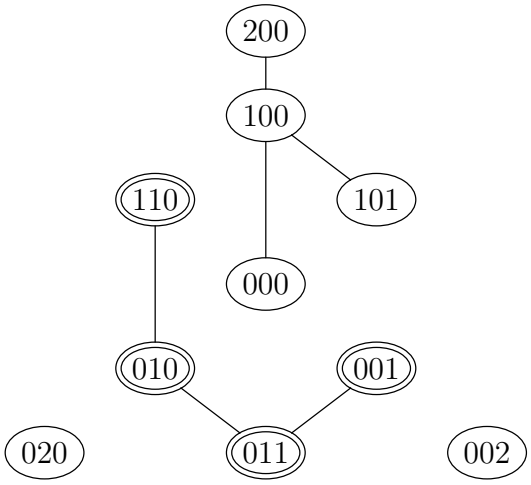


Figure 5.3: Partition of the parametric mesh associated with $\mathbb{I}_2^3$: all the interface nodes are connected.

5.2 Matrix Structure

Recall that the domain decomposition technique permutes rows and columns of the coefficient matrix A to obtain the 2-by-2 block matrix in (5.1) where A_{II} is a block-diagonal matrix. Suppose the parametric mesh $\mathcal{M}$ associated with $\mathbb{I}_k^M$ is partitioned into $N_{\mathcal{P}}$ submeshes, namely $\mathcal{P}_1$, $\mathcal{P}_2$, ..., $\mathcal{P}_{N_{\mathcal{P}}}$. Since each submesh does not overlap with other submeshes, i.e., their respective sets of nodes are disjoint, we obtain a particular structure of the matrix.

Let $S^{(i)}$ and $S_{\mathcal{I}}$ be the subspaces of S_k^M associated with the nodes in $\mathcal{P}_i$ and $\mathcal{I}$, respectively, defined by

$$S^{(i)} = \operatorname{span}\left\{\psi_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{I}_{\mathcal{P}_i}\right\},$$

$$S_{\mathcal{I}} = \operatorname{span}\left\{\psi_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{I}_{\mathcal{I}}\right\}.$$

Then,

$$S_k^M = \bigoplus_{i=1}^{N_{\mathcal{P}}} S^{(i)} \oplus S_{\mathcal{I}}.$$

Let $V^{(i)} \coloneqq S^{(i)} \otimes X_h$. Consider now the following discrete formulation

$$\begin{cases} \text{Find } u^{(i)} \in V^{(i)} \text{ such that,} \\ B(u^{(i)}, v) = F(v) \qquad \text{for all } v \in V^{(i)}, \end{cases} \tag{5.6}$$

where the bilinear form B and the functional F are defined in (2.12) and (2.13), respectively. Since the diffusion coefficient a is strictly positive, bounded and has a positive lower bound, the bilinear form $B : V^{(i)} \times V^{(i)} \to \mathbb{R}$ is bounded and coercive. By the Lax-Milgram theorem, the discrete formulation (5.6) is well-posed. That is there exists a unique $u^{(i)} \in V^{(i)}$ satisfying the discrete formulation (5.6), which results in the linear system

$$A^{(i)}\mathbf{u}^{(i)} = \mathbf{b}^{(i)}. \tag{5.7}$$

The coefficient matrix $A^{(i)}$ is a n_i-by-n_i block matrix with

$$A^{(i)}_{js} = \left\langle \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho K_0 + \sum_{m=1}^{M} \left\langle y_m \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho K_m, \qquad j, s = 1, 2, \ldots, n_i,$$

where $q^{(i)}$ is a bijection between $\{1, ..., n_i\}$ and $\mathbb{I}_{\mathcal{P}_i}$ and K_m, $m = 0, 1, ..., M$ are defined in (4.6). Note that $A^{(i)}$ is symmetric and positive definite.

The vectors $\mathbf{u}^{(i)}$ and $\mathbf{b}^{(i)}$ are vectors in $\mathbb{R}^{N_{\boldsymbol{x}} n_i}$, in particular,

$$\begin{aligned} \mathbf{u}^{(i)}_j &= [\ u_{1j} \quad u_{2j} \quad \cdots \quad u_{N_{\boldsymbol{x}} j}\ ]^T && j = 1, 2, \ldots, n_i, \\ \left[\mathbf{b}^{(i)}_s\right]_r &= \int_\Gamma \rho(\boldsymbol{y}) \psi_{q^{(i)}(s)} \mathrm{d}\boldsymbol{y} \cdot \int_D f \phi_r \mathrm{d}\boldsymbol{x} && r = 1, 2, \ldots, N_{\boldsymbol{x}}, s = 1, 2, \ldots, n_i. \end{aligned}$$

The following result indicates the pattern of the coefficient matrix $A^{(i)}$.

Proposition 5.13. *Let $\mathcal{P}_i$ be a submesh in the parametric mesh associated with $\mathbb{I}^M_k$. Assume that $\mathcal{P}_i$ is a path which is the sequence of nodes in $\mathcal{P}_i$, denoted by $\mathcal{P}_i = \boldsymbol{\alpha}^{(1)}\boldsymbol{\alpha}^{(2)} \cdots \boldsymbol{\alpha}^{(n_i)}$. Define $q^{(i)} : \{1, 2, ..., n_i\} \to \mathbb{I}_{\mathcal{P}_i}$ by $q^{(i)}(j) = \boldsymbol{\alpha}^{(j)}$. Then, $A^{(i)}$ in (5.7) is a block-tridiagonal matrix.*

Proof. Let $j \in \{1, 2, ..., n_i - 1\}$ and $s \in \{j + 2, ..., n_i\}$.

It is obvious that $\left\langle \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho = 0$. Since the multi-indices $q^{(i)}(j)$ and $q^{(i)}(s)$ are not adjacent in the parametric mesh, they do not satisfy condition (5.5). That is $\left\langle y_m \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho = 0$. Consequently, $A^{(i)}_{js} = \mathbf{0}$. By symmetry of the matrix $A^{(i)}$, we have $A^{(i)}_{sj} = A^{(i)}_{js} = \mathbf{0}$. Therefore, $A^{(i)}$ is a block-tridiagonal matrix.

□

Proposition 5.13 is very useful and could improve the performance of the solver if an efficient block-tridiagonal solver is provided.

Example 5.3. Suppose the parametric mesh in Example 5.1 is partitioned as in Figure 5.2. That is,

$$\mathbb{I}_{\mathcal{P}_1} = \{200, 100, 101, 001\}, \qquad \mathbb{I}_{\mathcal{P}_2} = \{010, 020\}$$

and the set of nodes on the interface is

$$\mathbb{I}_{\mathcal{I}} = \{110, 000, 011, 002\}.$$

Then, we have the coefficient matrices $A^{(1)}$ and $A^{(2)}$ as follows

$$A^{(1)} = \begin{bmatrix} K_0 & c_2^1 K_1 & & \\ c_2^1 K_1 & K_0 & c_1^3 K_3 & \\ & c_1^3 K_3 & K_0 & c_1^1 K_1 \\ & & c_1^1 K_1 & K_0 \end{bmatrix}, A^{(2)} = \begin{bmatrix} K_0 & c_2^2 K_2 \\ c_2^2 K_2 & K_0 \end{bmatrix},$$

where c_j^m is defined in (2.10).

Note that if $\left\langle y_m \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho \neq 0$ where $q^{(i)}(j) = \boldsymbol{\beta}$ and $q^{(i)}(s) = \boldsymbol{\beta}'$, by Theorem 2.11, we have $\beta_{m'} = \beta'_{m'}$ for all $m' \in \{1, \ldots, M\} \setminus \{m\}$ and $|\beta_m - \beta'_m| = 1$. Thus, $\left\langle y_m \psi_{q^{(i)}(j)}, \psi_{q^{(i)}(s)} \right\rangle_\rho = c^m_{\max\{\beta_m, \beta'_m\}}$.

In general, a submesh $\mathcal{P}_i$ in a parametric mesh may not be associated with a block-tridiagonal matrix $A^{(i)}$ as the follow example shows.

Example 5.4. Suppose the parametric mesh in Example 5.1 is partitioned as in Figure 5.3. That is,

$$\mathbb{I}_{\mathcal{P}_1} = \{200, 100, 000, 101\}\,,\ \mathbb{I}_{\mathcal{P}_2} = \{020\}\ \text{ and } \mathbb{I}_{\mathcal{P}_3} = \{002\}\,,$$

and the set of nodes on the interface is

$$\mathbb{I}_{\mathcal{I}} = \{110, 010, 011, 001\}\,.$$

We choose the map $q^{(1)} : \{1, 2, 3, 4\} \to \mathbb{I}_{\mathcal{P}_1}$ to be

$$q^{(1)}(i) = \begin{cases} 200 & i = 1, \\ 100 & i = 2, \\ 000 & i = 3, \\ 101 & i = 4. \end{cases}$$

Then, we have the coefficient matrices $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$ as follows

$$A^{(1)} = \begin{bmatrix} K_0 & c_2^1 K_1 & & \\ c_2^1 K_1 & K_0 & c_1^1 K_1 & c_1^3 K_3 \\ & c_1^1 K_1 & K_0 & \\ & c_1^3 K_3 & & K_0 \end{bmatrix} \text{ and } A^{(2)} = A^{(3)} = K_0.$$

where c_j^m is defined in (2.10).

We would like to characterise the partitions that yield a block diagonal matrix A_{II}. We permute the orthonormal basis of S_k^M corresponding to the partition. Since one submesh $\mathcal{P}_i$ in a non-overlapping partition yields a block matrix $A^{(i)}$, we have the block diagonal matrix A_{II} in (5.1) to be

$$A_{II} = \bigoplus_{i=1}^{N_{\mathcal{P}}} A^{(i)}.$$

Let $N_{\boldsymbol{y}_\mathcal{P}}$ be the total number of nodes in all submeshes, i.e., $N_{\boldsymbol{y}_\mathcal{P}} \coloneqq \sum_{i=1}^{N_\mathcal{P}} n_i$, and $\mathbb{I}_\mathcal{P} \coloneqq \bigcup_{i=1}^{N_\mathcal{P}} \mathbb{I}_{\mathcal{P}_i}$ be a set of all nodes in all submeshes. So, $N_{\boldsymbol{y}_\mathcal{P}} = \#\mathbb{I}_\mathcal{P}$. The block-diagonal matrix A_{II} is induced by bijection $q_\mathcal{P}$ which is a map to group the basis elements of S_k^M in the same order as $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{N_\mathcal{P}}$, respectively. That is, we let $q_\mathcal{P} : \left\{1, 2, \ldots, N_{\boldsymbol{y}_\mathcal{P}}\right\} \to \mathbb{I}_\mathcal{P}$ be defined by

$$q_\mathcal{P}(j) = \begin{cases} q^{(1)}(j) & \text{for } 1 \le j \le n_1, \\ q^{(2)}(j - n_1) & \text{for } n_1 + 1 \le j \le \sum_{i=1}^{2} n_i, \\ \vdots & \vdots \\ q^{(N_\mathcal{P})}(j - \sum_{i=1}^{N_\mathcal{P}-1} n_i) & \text{for } \sum_{i=1}^{N_\mathcal{P}-1} n_i + 1 \le j \le N_{\boldsymbol{y}_\mathcal{P}}, \end{cases}$$

where $q^{(i)}$ is a bijection map from $\{1, ..., n_i\}$ to $\mathbb{I}_{\mathcal{P}_i}$.

The stochastic Galerkin matrix A in (5.1) can be constructed by permuting the nodes in $\mathbb{I}_k^M$. In other words, we need to choose a suitable bijection $q : \{1, 2, ..., N_{\boldsymbol{y}}\} \to \mathbb{I}_k^M$. Let $N_{\boldsymbol{y}_\mathcal{I}}$ be the total number of nodes on the interface $\mathcal{I}$, i.e., $N_{\boldsymbol{y}_\mathcal{I}} \coloneqq \#\mathbb{I}_\mathcal{I}$ and $q_\mathcal{I}$ be any bijection from $\left\{1, 2, ..., N_{\boldsymbol{y}_\mathcal{I}}\right\}$ to $\mathbb{I}_\mathcal{I}$. We choose the bijection q to be

$$q(j) = \begin{cases} q_\mathcal{P}(j) & \text{for } 1 \le j \le N_{\boldsymbol{y}_\mathcal{P}}, \\ q_\mathcal{I}(j - N_{\boldsymbol{y}_\mathcal{P}}) & \text{for } N_{\boldsymbol{y}_\mathcal{P}} + 1 \le j \le N_{\boldsymbol{y}}. \end{cases}$$

By this choice of the map q, the stochastic Galerkin matrix A can be permuted to the 2×2 block matrix (5.1) with A_{II} a block-diagonal matrix, as required.

5.3 Domain Decomposition Preconditioners

We have seen how the coefficient matrix A can be permuted to a 2-by-2 block matrix. In this section, we will discuss how to design a preconditioner, which is required to be symmetric and positive definite, based on the structure of the matrix A. Moreover, we

discuss its complexity and analyse its spectrum.

5.3.1 Block Preconditioners

In order to design a preconditioner, the matrix A is factorised as follows

$$\begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix} = \begin{bmatrix} I & \\ A_{FI}A_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{II} & \\ & S \end{bmatrix} \begin{bmatrix} I & A_{II}^{-1}A_{IF} \\ & I \end{bmatrix}$$

where $S = A_{FF} - A_{FI}A_{II}^{-1}A_{IF}$ is the Schur complement.

Note that since the matrix A is symmetric and positive definite, so is S.

In order to obtain an efficient preconditioner, A_{II} and S need to be approximated so that the actions of A_{II}^{-1} or S^{-1} on a vector are not expensive.

Suppose A_{II} and S are approximated by symmetric and positive definite matrices $\tilde{A}_{II}$ and $\tilde{S}$, respectively. Thus, a 2-by-2 block preconditioner for the matrix A can be defined by

$$P_B = \begin{bmatrix} I & \\ A_{FI}\tilde{A}_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix} \begin{bmatrix} I & \tilde{A}_{II}^{-1}A_{IF} \\ & I \end{bmatrix}.$$

Under the assumption of positivity of $\tilde{A}_{II}$ and $\tilde{S}$, P_B is symmetric and positive definite.

Remark. Let $\mathbb{I}_{\mathcal{I}_i} \subseteq \mathbb{I}_{\mathcal{I}}$ be the set of nodes on the interface that are adjacent to the submesh $\mathcal{P}_i$ for $i = 1, 2, \ldots, N_{\mathcal{P}}$. Denote $N_{\mathcal{I}_i}$ to be the number of nodes in $\mathbb{I}_{\mathcal{I}_i}$. Next, define a stochastic restriction matrix $R_{\mathcal{I}_i} \in \mathbb{R}^{N_{\mathcal{I}_i} \times N_{\mathcal{I}}}$ which maps a vector in $\mathbb{R}^{N_{\mathcal{I}}}$ to a vector corresponding to the multi-indices in $\mathbb{I}_{\mathcal{I}_i}$. Additionally, we define a global restriction matrix $\mathcal{R}_{\mathcal{I}_i}$ to be

$$\mathcal{R}_{\mathcal{I}_i} \coloneqq R_{\mathcal{I}_i} \otimes I_{N_{\boldsymbol{x}}}.$$

Because A_{FF} arises from the interface nodes, we may assume that

$$A_{FF} = \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{I}_i}^T A_{FF}^{(i)} \mathcal{R}_{\mathcal{I}_i},$$

where $A_{FF}^{(i)}$ are matrices of size $N_{\mathcal{I}_i}N_{\boldsymbol{x}} \times N_{\mathcal{I}_i}N_{\boldsymbol{x}}$. They represent actions between interface nodes adjacent to the submesh $\mathcal{P}_i$. As a result, the Schur complement S can be viewed as a summation of the Schur complement $S^{(i)}$ from each submesh, i.e.,

$$S = \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{I}_i}^T S^{(i)} \mathcal{R}_{\mathcal{I}_i}.$$

Thus, S^{-1} can be approximated by

$$\tilde{S}^{-1} \coloneqq \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{I}_i}^T \left[\tilde{S}^{(i)}\right]^{-1} \mathcal{R}_{\mathcal{I}_i}, \tag{5.8}$$

where $\tilde{S}^{(i)}$ is an approximation of $S^{(i)}$.

This preconditioning technique is similar to some classical domain decomposition preconditioners such as FETI methods or balancing domain decomposition if $\tilde{S}$ can be represented in the form in (5.8).

5.3.1.1 Computational Costs

Consider $\mathbf{r} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}}$ under the same permutation used for A:

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_F \end{bmatrix},$$

where $\mathbf{r}_I \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{P}}}$ and $\mathbf{r}_F \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}}$. Solving the linear system

$$P_B \mathbf{z} = \mathbf{r} \tag{5.9}$$

requires the solution of three linear systems in the following steps.

1. Solve for $\mathbf{z}^{(1)} = \begin{bmatrix} \mathbf{z}_I^{(1)} & \mathbf{z}_F^{(1)} \end{bmatrix}^T$

$$\begin{bmatrix} I & \\ A_{FI}\tilde{A}_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{z}_I^{(1)} \\ \mathbf{z}_F^{(1)} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_F \end{bmatrix}.$$

2. Solve for $\mathbf{z}^{(2)} = \begin{bmatrix} \mathbf{z}_I^{(2)} & \mathbf{z}_F^{(2)} \end{bmatrix}^T$

$$\begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix} \begin{bmatrix} \mathbf{z}_I^{(2)} \\ \mathbf{z}_F^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_I^{(1)} \\ \mathbf{z}_F^{(1)} \end{bmatrix}.$$

3. Solve for $\mathbf{z} = \begin{bmatrix} \mathbf{z}_I & \mathbf{z}_F \end{bmatrix}^T$

$$\begin{bmatrix} I & \tilde{A}_{II}^{-1}A_{IF} \\ & I \end{bmatrix} \begin{bmatrix} \mathbf{z}_I \\ \mathbf{z}_F \end{bmatrix} = \begin{bmatrix} \mathbf{z}_I^{(2)} \\ \mathbf{z}_F^{(2)} \end{bmatrix}.$$

We summarise these steps in Algorithm 3.

Algorithm 3 The algorithm to solve the linear system $P_B\mathbf{z} = \mathbf{r}$.

$\mathbf{z}_I^{(2)} = \tilde{A}_{II}^{-1}\mathbf{r}_I$
$\mathbf{z}_F^{(1)} = \mathbf{r}_F - A_{FI}\mathbf{z}_I^{(2)}$
$\mathbf{z}_F = \tilde{S}^{-1}\mathbf{z}_F^{(1)}$
$\mathbf{z}_I = \mathbf{z}_I^{(2)} - \tilde{A}_{II}^{-1}A_{IF}\mathbf{z}_F$

According to Algorithm 3, we need to solve three linear systems: two systems with $\tilde{A}_{II}$ as a system matrix and one with the matrix $\tilde{S}$. In addition, we require two matrix-vector multiplications involving A_{FI} and A_{IF}. By Theorem 2.11, the coefficient matrix A has at most $2M+1$ non-zero block matrices per row. Thus, the matrix A_{IF} has at most $2M$ non-zero block matrices per row because it is a certain that K_0 is in A_{II} and A_{FF}. As a

result, the sequential complexity for the action of P_B^{-1} on a vector is

$$\mathcal{F}\ell\left(P_B^{-1}\mathbf{r}\right) \approx 2\mathcal{F}\ell\left(\tilde{A}_{II}^{-1}\mathbf{v}\right) + \mathcal{F}\ell\left(\tilde{S}^{-1}\mathbf{v}\right) + 4MN_{\boldsymbol{y}_{\mathcal{P}}}\mathrm{nnz}(K_0).$$

Furthermore, due to the structure of A_{II}, we can solve the linear system with A_{II} in parallel which is the main advantage of our domain decomposition technique. On the other hand, its parallel complexity depends on how the parametric mesh is partitioned, i.e., the size of the largest submeshes. Moreover, the parallel complexity for $\tilde{S}^{-1}\mathbf{v}$ depends on the choice of $\tilde{S}$. Thus, we have

$$\mathcal{F}\ell p\left(P_B^{-1}\mathbf{r}\right) \approx 2\mathcal{F}\ell p\left(\tilde{A}_{II}^{-1}\mathbf{v}\right) + \mathcal{F}\ell p\left(\tilde{S}^{-1}\mathbf{v}\right) + 2\mathrm{nnz}(K_0).$$

5.3.1.2 Spectral Analysis

To derive bounds for the generalised eigenvalue problem $A\mathbf{v} = \lambda P_B\mathbf{v}$, i.e.,

$$\begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix}\mathbf{v} = \lambda \begin{bmatrix} I & \\ A_{FI}\tilde{A}_{II}^{-1} & I \end{bmatrix}\begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix}\begin{bmatrix} I & \tilde{A}_{II}^{-1}A_{IF} \\ & I \end{bmatrix}\mathbf{v}, \tag{5.10}$$

the following proposition is essential.

Proposition 5.14 ([5, Proposition 2.1])**.** *The eigenvalues of the generalised eigenvalue problem in (5.10) are the same as the eigenvalues of the matrix*

$$\begin{bmatrix} \tilde{A}_{II}^{-1}A_{II} & \\ & \tilde{S}^{-1}S \end{bmatrix}\begin{bmatrix} I & \\ E_{FI} & I \end{bmatrix}\begin{bmatrix} I & E_{IF} \\ & I \end{bmatrix},$$

where $E_{FI} = \left(A_{II}^{-1} - \tilde{A}_{II}^{-1}\right)A_{IF}$ *and* $E_{FI} = S^{-1}A_{FI}\left(I - \tilde{A}_{II}^{-1}A_{II}\right)$.

Proposition 5.14 is a very useful tool to analyse the eigenvalues bounds for problem

(5.10) in the next section.

5.3.2 Block-diagonal Preconditioners

One of the factors of the matrix A is a block-diagonal matrix which is symmetric and positive definite, i.e.,

$$\begin{bmatrix} A_{II} & \\ & S \end{bmatrix}.$$

As a result, it is feasible to design a block-diagonal preconditioner P_D for the matrix A from such a factor of A. Again, assume that A_{II} and S are approximated by symmetric and positive definite matrices $\tilde{A}_{II}$ and $\tilde{S}$, respectively. Thus, the block-diagonal preconditioner for the matrix A can be defined by

$$P_D = \begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix}, \tag{5.11}$$

which is symmetric and positive definite.

Remark. The preconditioner P_D can be viewed as a Schwarz preconditioner. That is let $R_{\mathcal{P}_i} \in \mathbb{R}^{n_i \times N_{\boldsymbol{y}}}$ and $R_{\mathcal{I}} \in \mathbb{R}^{N_{\mathcal{I}} \times N_{\boldsymbol{y}}}$ be stochastic restriction matrices which map a vector in $\mathbb{R}^{N_{\boldsymbol{y}}}$ to a vector corresponding to the multi-indices in $\mathbb{I}_{\mathcal{P}_i}$ and $\mathbb{I}_{\mathcal{I}}$, respectively. Next, define a global restriction matrices $\mathcal{R}_{\mathcal{P}_i}$ and $\mathcal{R}_{\mathcal{I}}$ to be

$$\mathcal{R}_{\mathcal{P}_i} \coloneqq R_{\mathcal{P}_i} \otimes I_{N_{\boldsymbol{x}}} \text{ and } \mathcal{R}_{\mathcal{I}} \coloneqq R_{\mathcal{I}} \otimes I_{N_{\boldsymbol{x}}}.$$

Suppose $\tilde{A}^{(i)}$ is an approximation to the matrix $A^{(i)}$ in (5.7) for $i = 1, 2, \ldots, N_{\mathcal{P}}$ and $\tilde{S}$ is an approximation for the Schur complement S. Then, the preconditioner P_D can be rewritten as

$$P_D = \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{P}_i}^T \tilde{A}^{(i)} \mathcal{R}_{\mathcal{P}_i} + \mathcal{R}_{\mathcal{I}}^T \tilde{S} \mathcal{R}_{\mathcal{I}}.$$

5.3.2.1 Computational Costs

Block-diagonal preconditioners are very convenient to use due to their structure. Let $\mathbf{r} \in \mathbb{R}^{N_{\boldsymbol{x}} N_{\boldsymbol{y}}}$ have the block structure

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_I \\ \\ \mathbf{r}_F \end{bmatrix},$$

where $\mathbf{r}_I \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{P}}}$ and $\mathbf{r}_F \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}}$. To solve the linear system

$$P_D \mathbf{z} = \mathbf{r}, \tag{5.12}$$

requires the solution of two linear subsystems with the system matrices $\tilde{A}_{II}$ and $\tilde{S}$, as

$$\mathbf{z} \coloneqq \begin{bmatrix} \mathbf{z}_I \\ \\ \mathbf{z}_F \end{bmatrix} = \begin{bmatrix} \tilde{A}_{II}^{-1} \mathbf{r}_I \\ \\ \tilde{S}^{-1} \mathbf{r}_F \end{bmatrix}.$$

The sequential complexity for P_D is

$$\mathcal{F}\ell \left(P_D^{-1} \mathbf{r} \right) = \mathcal{F}\ell \left(\tilde{A}_{II}^{-1} \mathbf{v} \right) + \mathcal{F}\ell \left(\tilde{S}^{-1} \mathbf{v} \right).$$

Given the structure of P_D, we can solve for $\mathbf{z}_I$ and $\mathbf{z}_F$ in parallel. Thus, we have

$$\mathcal{F}\ell p \left(P_D^{-1} \mathbf{r} \right) = \max \left\{ \mathcal{F}\ell p \left(\tilde{A}_{II}^{-1} \mathbf{v} \right), \mathcal{F}\ell p \left(\tilde{S}^{-1} \mathbf{v} \right) \right\}.$$

Note that the computational cost in parallel for $\tilde{S}^{-1}\mathbf{v}$ tends to increase the closer $\tilde{S}$ is. To avoid this bottleneck, we should balance the parallel computational cost for $\tilde{A}_{II}^{-1}\mathbf{v}$ and $\tilde{S}^{-1}\mathbf{v}$ by adjusting the size of submeshes or the interface.

5.3.2.2 Spectral Analysis

Bounds for the eigenvalues of the preconditioned system $P_D^{-1}A$ are derived in the following proposition.

Proposition 5.15. *Let A be the 2-by-2 block matrix*

$$A = \begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix}.$$

Let P_D be defined in (5.11) with $\tilde{A}_{II} = A_{II}$. Assume that

$$\Lambda\left(\tilde{S}^{-1}S\right) \subseteq [\theta_1, \Theta_1] \text{ and } \Lambda\left(\tilde{S}^{-1}A_{FF}\right) \subseteq [\theta_2, \Theta_2],$$

where $(1+\theta_2)^2 \geq 4\Theta_1$. Then, the eigenvalue bounds of the generalised eigenvalue problem $A\mathbf{v} = \lambda P_D \mathbf{v}$ are

$$\frac{2\theta_1}{\gamma} \leq \lambda \leq \frac{\gamma}{2},$$

where $\gamma = 1 + \Theta_2 + \sqrt{(1+\Theta_2)^2 - 4\theta_1}$.

Proof. Let λ and $\mathbf{v}$ be an eigenvalue and an eigenvector of the generalised eigenvalue problem $A\mathbf{v} = \lambda P_D \mathbf{v}$. We write $\mathbf{v}$ as

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_I \\ \mathbf{v}_F \end{bmatrix}.$$

So, we get

$$A_{II}\mathbf{v}_I + A_{IF}\mathbf{v}_F = \lambda A_{II}\mathbf{v}_I, \tag{5.13}$$

$$A_{FI}\mathbf{v}_I + A_{FF}\mathbf{v}_F = \lambda \tilde{S}\mathbf{v}_F. \tag{5.14}$$

It is clear that the vector $\mathbf{v}$ with $\mathbf{v}_I \notin \ker(A_{FI})$ and $\mathbf{v}_F = \mathbf{0}$ is not an eigenvector of the generalised eigenvalue problem $A\mathbf{v} = \lambda P_D \mathbf{v}$ because the equation (5.14) is not satisfied.

Suppose $\mathbf{v}_F = \mathbf{0}$ and $\mathbf{v}_I \in \ker(A_{FI})$. We have that $\lambda = 1$.

Suppose $\mathbf{v}_F \neq \mathbf{0}$. From (5.13), we obtain

$$(1-\lambda)A_{FI}\mathbf{v}_I + A_{FI}A_{II}^{-1}A_{IF}\mathbf{v}_F = \mathbf{0}. \tag{5.15}$$

Substitute (5.14) in (5.15) and get

$$(1-\lambda)\left(\lambda\tilde{S}\mathbf{v}_F - A_{FF}\mathbf{v}_F\right) + A_{FI}A_{II}^{-1}A_{IF}\mathbf{v}_F = \mathbf{0}.$$

This leads to

$$\frac{\mathbf{v}_F^T S\mathbf{v}_F}{\mathbf{v}_F^T\tilde{S}\mathbf{v}_F} = \lambda(1-\lambda) + \lambda\frac{\mathbf{v}_F^T A_{FF}\mathbf{v}_F}{\mathbf{v}_F^T\tilde{S}\mathbf{v}_F}.$$

Hence, we solve for λ and get

$$\lambda = \frac{1}{2}\left(1 + \frac{\mathbf{v}_F^T A_{FF}\mathbf{v}_F}{\mathbf{v}_F^T\tilde{S}\mathbf{v}_F} \pm \sqrt{\left(1 + \frac{\mathbf{v}_F^T A_{FF}\mathbf{v}_F}{\mathbf{v}_F^T\tilde{S}\mathbf{v}_F}\right)^2 - 4\frac{\mathbf{v}_F^T S\mathbf{v}_F}{\mathbf{v}_F^T\tilde{S}\mathbf{v}_F}}\right).$$

Note that, since $(1+\theta_2)^2 \geq 4\Theta_1$, then λ is real.

By the eigenvalue bounds of $\tilde{S}^{-1}S$ and $\tilde{S}^{-1}A_{FF}$, we obtain bounds as follows,

$$\frac{2\theta_1}{1+\Theta_2+\sqrt{(1+\Theta_2)^2-4\theta_1}} \leq \lambda \leq \frac{1}{2}\left(1+\Theta_2+\sqrt{(1+\Theta_2)^2-4\theta_1}\right).$$

□

If the assumption in the above theorem holds, the theorem provides tight eigenvalue bounds of $P_D^{-1}A$. We will use this result later.

5.4 Even-odd Partition and Its Preconditioners

We introduce in this section a certain partitioning strategy and the preconditioner associated with it. Generally, for domain decomposition on spatial domain, the number of

subdomains is chosen depending on the resources we have such as the number of processors or memory. Each subdomain should have roughly the same size and be sufficiently small to be solved by a direct solver. This is because when solving a linear system with A_{II} in parallel, its parallel complexity is dominated by the complexity on the largest subdomain. In addition, if we consider the parallel complexity of the block-diagonal preconditioner P_D, the size of the interface should not be very different from the size of the largest subdomain due to the number of nodes on the interface inducing the size of the Schur complement. However, our partitioning strategy is one of the extreme cases with each submesh having only one node. It aims to maximise the number of submeshes to maximise the capability of parallelism. Typically, the number of subdomains in a spatial mesh is related to the size of the interface. Although this also occur in parametric mesh, some characteristics of parametric mesh and spatial mesh are different. Before moving to the main result, the following definitions are required.

Definition 5.16. Let S be a set of vertices of a graph G. The set S is called an independent set if no two vertices in S are adjacent.

Definition 5.17. Let G be a graph. G is a bipartite graph if the set of vertices can be divided into two disjoint and independent sets such that every edge in graph G connects a vertex from one set to the other set.

The following result is a property of the parametric mesh associated with $\mathbb{I}_k^M$.

Theorem 5.18. *Let $\mathcal{M}$ be the parametric mesh associated with $\mathbb{I}_k^M$ for the discrete formulation (2.14) with affine-parametric diffusion coefficient. Then, $\mathcal{M}$ is a bipartite graph.*

Proof. Let $\boldsymbol{\alpha} \in \mathbb{I}_k^M$ and let $\boldsymbol{\beta}$ be an adjacent node to $\boldsymbol{\alpha}$. This means that there exists $m \in \{1, ..., M\}$ such that $|\alpha_m - \beta_m| = 1$ and $\alpha_{m'} = \beta_{m'}$ for all $m' \in \{1, ..., M\} \setminus \{m\}$. Suppose that $|\boldsymbol{\alpha}|$ is even, then $|\boldsymbol{\beta}|$ is odd and vice versa. That is every edge in parametric mesh connects an even and an odd node together. Thus, the parametric mesh is bipartite. □

Theorem 5.18 indicates that the set of multi-indices $\mathbb{I}_k^M$ can be divided into two disjoint and independent sets. That is every edge links a node in one set to another node in the other set and no any pair of nodes are adjacent in each set.

Example 5.5. Consider the even-odd partition described in Theorem 5.18 for the case $M = 3$ and $k = 2$.

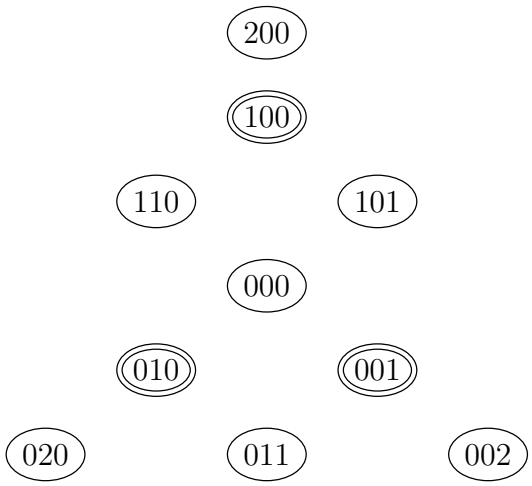


Figure 5.4: The parametric mesh associated with $\mathbb{I}_2^3$ is partitioned based on the property of bipartite graph. Each submesh has only one node. Additionally, no two nodes on the interface are adjacent.

Note that this partitioning strategy does not minimise the number of nodes on the interface. For instance, in the case $M = 1$, the parametric mesh is a path from index with the degree 0 to the degree k. Suppose we want to partition the parametric mesh into 2 submeshes for 2 processors. We can choose the node with the index $\lfloor k/2 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function, to be the interface. Thus, this partition has only one node on the interface. On the other hand, even-odd partitioning strategy gives $k + 1 - \lfloor k/2 \rfloor$ submeshes. We may assign half of the number of submeshes to one processor and the rest to another processor. However, this partition produces $\lfloor k/2 \rfloor$ nodes on the interface.

Note also that, in practice, $\mathbb{I}_{\mathcal{P}}$ can be chosen to be the smaller set. For instance, we may choose $\mathbb{I}_{\mathcal{P}}$ and $\mathbb{I}_{\mathcal{I}}$ to be

$$\mathbb{I}_{\mathcal{P}} = \{200, 110, 101, 000, 020, 011, 002\} \text{ and } \mathbb{I}_{\mathcal{I}} = \{100, 010, 001\}$$

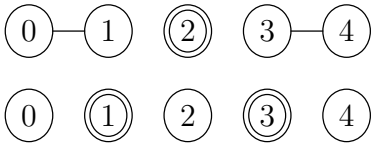


Figure 5.5: The parametric mesh when $M = 1$ and $k = 4$ is partitioned in two different ways. The first mesh is partitioned in two submeshes with the same size and has only one node on the interface. The other is the even-odd partition. It partitions the parametric mesh by choosing odd nodes to be the interface.

or

$$\mathbb{I}_{\mathcal{P}} = \{100, 010, 001\} \text{ and } \mathbb{I}_{\mathcal{I}} = \{200, 110, 101, 000, 020, 011, 002\}.$$

This type of partitioning strategy provides the desired structure of the coefficient matrix A. Due to no two nodes being adjacent on both sets $\mathbb{I}_{\mathcal{P}}$ and $\mathbb{I}_{\mathcal{I}}$, it results in the diagonal block structure of A_{II} and A_{FF} with K_0 along the main diagonal. This implies that all non-zeros block matrices outside the main diagonal are put in A_{IF} and A_{FI} as shown in Figure 5.6. That is

$$A = \begin{bmatrix} I_{N_{\boldsymbol{y}_{\mathcal{P}}}} \otimes K_0 & A_{IF} \\ A_{FI} & I_{N_{\boldsymbol{y}_{\mathcal{I}}}} \otimes K_0 \end{bmatrix}, \tag{5.16}$$

or

$$\begin{bmatrix} A_{II} & \\ & A_{FF} \end{bmatrix} = I_{N_{\boldsymbol{y}}} \otimes K_0 \text{ and } \begin{bmatrix} & A_{IF} \\ A_{FI} & \end{bmatrix} = \sum_{m=1}^{M} G_m \otimes K_m.$$

The advantage of this structure is the form of A_{II} and A_{FF}, so that the action of A_{II}^{-1} and A_{FF}^{-1} on a vector is not expensive. Thus, an approximation of A_{II} is not required but an approximation of the action of the inverse of the Schur complement is still needed.

5.4.1 Schur Complement Approximation

As we have seen in the previous section, the Schur complement S is one important component in the diagonal factor of the matrix A. If we could solve the linear system with

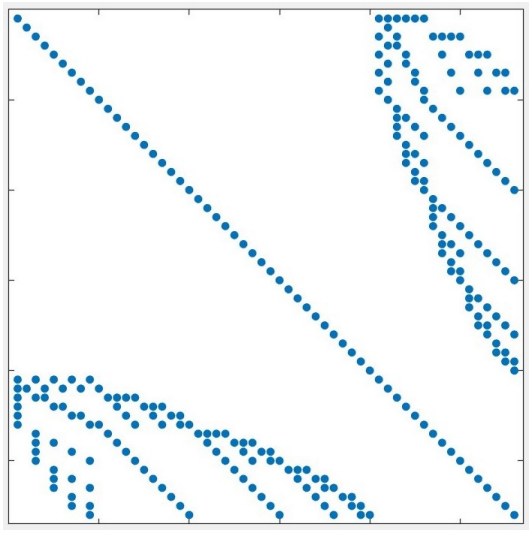


Figure 5.6: The sparsity pattern of the coefficient matrix A when $M = 5$ and $k = 3$ based on the partitioning outlined in Theorem 5.18.

the Schur complement exactly, the PCG using the preconditioner P_B will converge in one iteration. Unfortunately, the complexity of applying the inverse of the Schur complement to a vector is high.

As seen in Algorithm 3, in one iteration of PCG, it is required to solve the linear system

$$\tilde{S}\mathbf{z} = \mathbf{r} \tag{5.17}$$

for a given vector $\mathbf{r}$. This operation usually is a bottleneck in the process because the structure of $\tilde{S}$ or S does not facilitate parallelism. As a result, a suitable approximation of the Schur complement is required.

In this subsection, we will discuss some approximations of the Schur complement. Moreover, its complexity and the eigenvalue bounds for $\tilde{S}^{-1}S$ are also provided.

5.4.1.1 Block-diagonal Approximation of the Schur Complement

Recall that $S = A_{FF} - A_{FI}A_{II}^{-1}A_{IF}$ is symmetric and positive definite. We may approximate the Schur complement S by its first term. That is

$$S \approx \tilde{S}_1 \coloneqq A_{FF}.$$

Remark. Since $A_{FF} = \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{I}_i}^T A_{FF}^{(i)} \mathcal{R}_{\mathcal{I}_i}$ is a diagonal block matrix in even-odd partition, so

$$A_{FF}^{-1} = \sum_{i=1}^{N_{\mathcal{P}}} \mathcal{R}_{\mathcal{I}_i}^T \mathcal{D}_i \left[A_{FF}^{(i)} \right]^{-1} \mathcal{D}_i \mathcal{R}_{\mathcal{I}_i}, \tag{5.18}$$

where $\mathcal{D}_i = D_i \otimes I_{N_{\boldsymbol{x}}}$ and $D_i \in \mathbb{R}^{N_{\mathcal{I}_i} \times N_{\mathcal{I}_i}}$ are diagonal weighting matrices such that

$$\sum_{i=1}^{N_{\mathcal{P}}} R_{\mathcal{I}_i}^T D_i R_{\mathcal{I}_i} = I_{N_{\boldsymbol{y}_{\mathcal{I}}}}.$$

Hence, $\tilde{S}_1$ can be written in the form in (5.8).

The diagonal structure of A_{FF} benefits the parallelism. Consequently, its sequential complexity and parallel complexity are

$$\mathcal{F}\ell\left(\tilde{S}_1^{-1}\mathbf{r}\right) = N_{\boldsymbol{y}_{\mathcal{I}}} \mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) \text{ and } \mathcal{F}\ell p\left(\tilde{S}_1^{-1}\mathbf{r}\right) = \mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right).$$

Before we study the spectrum of $\tilde{S}_1^{-1}S$, the following lemma is required.

Lemma 5.19. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16). Define*

$$Q = A_{FF}^{-\frac{1}{2}} A_{FI} A_{II}^{-1} A_{IF} A_{FF}^{-\frac{1}{2}}.$$

Then, the spectrum of Q satisfies

$$\Lambda(Q) \subseteq \left[0, \tau^2\right],$$

where τ is defined by

$$\tau = \frac{1}{a_0^{\min}} \left\| \sum_{m=1}^{\infty} |a_m| \right\|_{L^\infty(D)}.$$

Proof. We split the matrix A as $A = A_0 + A_1$ where

$$A_0 = \begin{bmatrix} A_{II} & \\ & A_{FF} \end{bmatrix} \text{ and } A_1 = \begin{bmatrix} & A_{IF} \\ A_{FI} & \end{bmatrix},$$

or

$$A_0 = I_{N_{\boldsymbol{y}}} \otimes K_0 \text{ and } A_1 = \sum_{m=1}^{M} G_m \otimes K_m.$$

Let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}} \backslash \{\mathbf{0}\}$. By Theorem 4.2 and Proposition 3.8, we have that

$$1 - \tau \leq \frac{\mathbf{v}^T A \mathbf{v}}{\mathbf{v}^T A_0 \mathbf{v}} \leq 1 + \tau, \quad \text{for all } \mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}} \backslash \{\mathbf{0}\}.$$

Then,

$$-\tau \leq \frac{\mathbf{v}^T A_1 \mathbf{v}}{\mathbf{v}^T A_0 \mathbf{v}} \leq \tau, \quad \text{for all } \mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}} \backslash \{\mathbf{0}\}.$$

or

$$\Lambda\left(A_0^{-\frac{1}{2}} A_1 A_0^{-\frac{1}{2}}\right) \subseteq [-\tau, \tau].$$

Since the eigenvalues of $\left(A_0^{-\frac{1}{2}} A_1 A_0^{-\frac{1}{2}}\right)^2$ are the squares of the eigenvalues of $A_0^{-\frac{1}{2}} A_1 A_0^{-\frac{1}{2}}$,

$$0 \leq \frac{\mathbf{v}^T \left(A_0^{-\frac{1}{2}} A_1 A_0^{-\frac{1}{2}}\right)^2 \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq \tau^2, \quad \text{for all } \mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}} \backslash \{\mathbf{0}\}.$$

Next, by the change of variable $\mathbf{u} = A_0^{-\frac{1}{2}} \mathbf{v}$ and $\left(A_0^{-\frac{1}{2}} A_1 A_0^{-\frac{1}{2}}\right)^2 = A_0^{-\frac{1}{2}} A_1 A_0^{-1} A_1 A_0^{-\frac{1}{2}}$, we obtain

$$0 \leq \frac{\mathbf{u}^T A_1 A_0^{-1} A_1 \mathbf{u}}{\mathbf{u}^T A_0 \mathbf{u}} \leq \tau^2, \quad \text{for all } \mathbf{u} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}} \backslash \{\mathbf{0}\}.$$

Since

$$A_1 A_0^{-1} A_1 = \begin{bmatrix} A_{IF} A_{FF}^{-1} A_{FI} & \\ & A_{FI} A_{II}^{-1} A_{IF} \end{bmatrix},$$

by setting $\mathbf{u}^T = \begin{bmatrix} \mathbf{0}^T & \mathbf{w}^T \end{bmatrix}$ where $\mathbf{w} \in \mathbb{R}^{N_{\boldsymbol{y}_{\mathcal{I}}}} \backslash \{\mathbf{0}\}$, we obtain

$$\min_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^T A_{FI} A_{II}^{-1} A_{IF} \mathbf{w}}{\mathbf{w}^T A_{FF} \mathbf{w}} \geq 0,$$
$$\max_{\mathbf{w} \neq \mathbf{0}} \frac{\mathbf{w}^T A_{FI} A_{II}^{-1} A_{IF} \mathbf{w}}{\mathbf{w}^T A_{FF} \mathbf{w}} \leq \tau^2.$$

Finally, we set $\mathbf{z} = A_{FF}^{\frac{1}{2}}\mathbf{w}$ and get that

$$0 \leq \frac{\mathbf{z}^T Q \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \leq \tau^2, \quad \text{for all } \mathbf{z} \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}} \backslash \{\mathbf{0}\}.$$

□

Next, we derive bounds for the spectrum of $\tilde{S}_1^{-1} S$.

Proposition 5.20. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and let S be the Schur complement of A_{II} in the matrix A. Define*

$$\tilde{S}_1 = A_{FF}.$$

Then, the spectrum of $\tilde{S}_1^{-1} S$ satisfies

$$\Lambda\left(\tilde{S}_1^{-1} S\right) \subseteq \left[1 - \tau^2, 1\right].$$

Proof. Let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}} \backslash \{\mathbf{0}\}$. Consider the generalised Rayleigh quotient

$$\begin{aligned} \frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \tilde{S}_1 \mathbf{v}} &= \frac{\mathbf{v}^T \left(A_{FF} - A_{FI} A_{II}^{-1} A_{IF}\right) \mathbf{v}}{\mathbf{v}^T A_{FF} \mathbf{v}} \\ &= 1 - \frac{\mathbf{v}^T A_{FI} A_{II}^{-1} A_{IF} \mathbf{v}}{\mathbf{v}^T A_{FF} \mathbf{v}}. \end{aligned}$$

By the change of variable $\mathbf{v}_1 = A_{FF}^{\frac{1}{2}} \mathbf{v}$, we have

$$\frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \tilde{S}_1 \mathbf{v}} = 1 - \frac{\mathbf{v}_1^T Q \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1},$$

where $Q = A_{FF}^{-\frac{1}{2}} A_{FI} A_{II}^{-1} A_{IF} A_{FF}^{-\frac{1}{2}}$. Since Q is symmetric and positive semi-definite, this leads to

$$1 - \max_{\mathbf{v}_1 \neq \mathbf{0}} \frac{\mathbf{v}_1^T Q \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1} \leq \frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \tilde{S}_1 \mathbf{v}} \leq 1 - \min_{\mathbf{v}_1 \neq \mathbf{0}} \frac{\mathbf{v}_1^T Q \mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1}.$$

By Lemma 5.19, we have

$$1 - \tau^2 \leq \frac{\mathbf{v}^T S \mathbf{v}}{\mathbf{v}^T \tilde{S}_1 \mathbf{v}} \leq 1.$$

□

5.4.1.2 Symmetric Block Gauss-Seidel Approximation of the Schur Complement

Let $\mathbf{r} \in \mathbb{R}^{N_{\mathcal{I}}}$ and consider the linear system

$$S\mathbf{z} = \mathbf{r}, \tag{5.19}$$

where S denotes the Schur complement. It is easy to see that the vector $\mathbf{z}$ in (5.19) satisfies the following linear system

$$\begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix} \begin{bmatrix} \mathbf{z}_I \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r} \end{bmatrix}. \tag{5.20}$$

Since the action of inverse of A_{II} and A_{FF} on a vector are inexpensive, we utilise this advantage of A_{II} and A_{FF} by approximating the system matrix in equation (5.20). The system matrix is replaced by its corresponding symmetric block Gauss-Seidel approximation. Then, we have

$$\begin{bmatrix} A_{II} & A_{IF} \\ & A_{FF} \end{bmatrix} \begin{bmatrix} A_{II}^{-1} & \\ & A_{FF}^{-1} \end{bmatrix} \begin{bmatrix} A_{II} & \\ A_{FI} & A_{FF} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{z}}_I \\ \tilde{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{r} \end{bmatrix}$$

which leads to

$$\tilde{z} = A_{FF}^{-1} \left(A_{FF} + A_{FI} A_{II}^{-1} A_{IF} \right) A_{FF}^{-1} \mathbf{r}. \tag{5.21}$$

Thus, $\tilde{\mathbf{z}}$ is an approximation of $\mathbf{z}$ and the Schur complement S can be approximated by $\tilde{S}_2$ which is defined by

$$\tilde{S}_2 \coloneqq A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}.$$

Note that $\tilde{S}_2$ is symmetric and positive definite.

Remark. By (5.18), $\tilde{S}_2$ can be represented in the form in (5.8).

We can see in (5.21) that, to obtain $\tilde{\mathbf{z}}$, we need to apply A_{II}^{-1} and A_{FF}^{-1} to vectors, which are not expensive to compute.

Next, we rewrite the equation (5.21) as

$$\mathbf{z} = A_{FF}^{-1}\mathbf{r} + A_{FF}^{-1}A_{FI}A_{II}^{-1}A_{IF}A_{FF}^{-1}\mathbf{r},$$

and get the algorithm for solving the linear system (5.17) as shown in Algorithm 4.

Algorithm 4 Algorithm to solve linear system $\tilde{S}_2\mathbf{z} = \mathbf{r}$.

$\mathbf{z}_1 = A_{FF}^{-1}\mathbf{r}$
$\mathbf{z}_2 = A_{II}^{-1}A_{IF}\mathbf{z}_1$
$\mathbf{z}_3 = A_{FF}^{-1}A_{FI}\mathbf{z}_2$
$\mathbf{z} = \mathbf{z}_1 + \mathbf{z}_3$

According to Algorithm 4, to solve the linear system (5.17) requires solving three linear system with A_{II} or A_{FF} as coefficient matrices and two matrix vector multiplications. Consequently, we obtain the sequential complexity for solving linear system (5.17) as follows,

$$\mathcal{F}\ell\left(\tilde{S}_2^{-1}\mathbf{r}\right) = N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + N_{\boldsymbol{y}_{\mathcal{I}}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 4MN_{\boldsymbol{y}_{\mathcal{P}}}\mathrm{nnz}(K_0).$$

For parallel complexity, we gain the benefit from the structure of A_{II} and A_{FF}, thus

$$\mathcal{F}\ell p\left(\tilde{S}_2^{-1}\mathbf{r}\right) = 3\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 2\mathrm{nnz}(K_0).$$

The eigenvalue bounds for $\tilde{S}_2^{-1}S$ are derived as follows.

Proposition 5.21. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and let S be the Schur complement of A_{II} in the matrix A. Define*

$$\tilde{S}_2 = A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}.$$

Then, the spectrum of $\tilde{S}_2^{-1}S$ satisfies

$$\Lambda\left(\tilde{S}_2^{-1}S\right) \subseteq \left[1-\tau^4, 1\right].$$

Proof. Let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}}\backslash\{\mathbf{0}\}$. Consider the generalised Rayleigh quotient

$$\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} = \frac{\mathbf{v}^T\left(A_{FF} - A_{FI}A_{II}^{-1}A_{IF}\right)\mathbf{v}}{\mathbf{v}^T A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}\mathbf{v}}.$$

By the change of variable $\mathbf{v}_1 = A_{FF}^{\frac{1}{2}}\mathbf{v}$, we have

$$\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} = \frac{\mathbf{v}_1^T\left(I_{N_{\boldsymbol{y}_\mathcal{I}}} - Q\right)\mathbf{v}_1}{\mathbf{v}_1^T\left(I_{N_{\boldsymbol{y}_\mathcal{I}}} + Q\right)^{-1}\mathbf{v}_1}$$

where $Q = A_{FF}^{-\frac{1}{2}}A_{FI}A_{II}^{-1}A_{IF}A_{FF}^{-\frac{1}{2}}$. Since the matrix Q is symmetric and positive semi-definite, $I_{N_{\boldsymbol{y}_\mathcal{I}}} + Q$ is symmetric and positive definite. We set $\mathbf{v}_2 = \left(I_{N_{\boldsymbol{y}_\mathcal{I}}} + Q\right)^{-\frac{1}{2}}\mathbf{v}_1$ and obtain

$$\begin{aligned}\frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} &= \frac{\mathbf{v}_2^T\left(I_{N_{\boldsymbol{y}_\mathcal{I}}} - Q^2\right)\mathbf{v}_2}{\mathbf{v}_2^T\mathbf{v}_2} \\ &= 1 - \frac{\mathbf{v}_2^T Q^2\mathbf{v}_2}{\mathbf{v}_2^T\mathbf{v}_2}.\end{aligned}$$

Again, since Q is positive and semi-definite, we get

$$0 < 1 - \lambda_{\max}^2(Q) \leq \lambda(\tilde{S}_2^{-1}S) \leq 1 - \lambda_{\min}^2(Q). \tag{5.22}$$

Combine the results from (5.22) with Lemma 5.19 to obtain the eigenvalue bounds for $\tilde{S}_2^{-1}S$ as follows

$$\lambda_{\max}\left(\tilde{S}_2^{-1}S\right) \leq 1,$$
$$\lambda_{\min}\left(\tilde{S}_2^{-1}S\right) \geq 1 - \tau^4.$$

□

5.4.2 Even-odd Preconditioners

In this section, we will gather all knowledge from the previous sections to construct practical preconditioners based on the even-odd partition. Recall that the coefficient matrix A is a 2-by-2 block matrix, i.e.,

$$A = \begin{bmatrix} A_{II} & A_{IF} \\ A_{FI} & A_{FF} \end{bmatrix},$$

and there are two possible structures of preconditioners for the matrix A. That is

$$P_D = \begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix} \text{ and } P_B = \begin{bmatrix} I & \\ A_{FI}\tilde{A}_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} \tilde{A}_{II} & \\ & \tilde{S} \end{bmatrix} \begin{bmatrix} I & \tilde{A}_{II}^{-1}A_{IF} \\ & I \end{bmatrix}.$$

where $\tilde{A}_{II}$ and $\tilde{S}$ are approximations of A_{II} and the Schur complement S, respectively.

By the even-odd partitioning strategy, we have

$$A_{II} = I_{N_{\boldsymbol{y}_\mathcal{P}}} \otimes K_0 \text{ and } A_{FF} = I_{N_{\boldsymbol{y}_\mathcal{I}}} \otimes K_0,$$

which are block diagonal matrices with the stiffness matrix K_0 along the main diagonal on both A_{II} and A_{FF}. Thus, the application of the inverse of A_{II} should be inexpensive. We assume that we invert A_{II} exactly and that the Schur complement S is approximated

by

$$\tilde{S}_1 = A_{FF} \text{ or } \tilde{S}_2 = A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}.$$

The combination of preconditioner structures and Schur complement approximations leads to four versions of the even-odd preconditioners. However, one of them is the mean-based preconditioner when the preconditioner is block-diagonal and the Schur complement is approximated by A_{FF}.

We will consider each of the three preconditioners in detail. Additionally, we will combine the result from section 5.3 and subsection 5.4.1 to analyse the complexity and also to derive the eigenvalue bounds of the preconditioned system for the all three versions of the even-odd preconditioners.

Version I: Block-diagonal Preconditioner with the Schur Complement Approximation $\tilde{S}_2$

We define the first version of our even-odd preconditioners by

$$P_{D2} = \begin{bmatrix} A_{II} & \\ & \tilde{S}_2 \end{bmatrix}. \tag{5.23}$$

Given its structure, the complexities of applying P_{D2}^{-1} to a vector are

$$\mathcal{F}\ell\left(P_{D2}^{-1}\mathbf{r}\right) = 2N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 4MN_{\boldsymbol{y}_\mathcal{P}}\text{nnz}(K_0),$$

and

$$\mathcal{F}\ell p\left(P_{D2}^{-1}\mathbf{r}\right) = 3\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 2\text{nnz}(K_0).$$

Before we derive the eigenvalue bounds for $P_{D2}^{-1}A$, we need the following lemma.

Lemma 5.22. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and*

S denote the Schur complement. Define

$$\tilde{S}_2 = A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}.$$

Then, the spectrum of $\tilde{S}_2^{-1}A_{FF}$ satisfies

$$\Lambda\left(\tilde{S}_2^{-1}A_{FF}\right) \subseteq \left[1, 1+\tau^2\right].$$

Proof. Let $\mathbf{v} \in \mathbb{R}^{N_{y_\mathcal{I}}} \setminus \{\mathbf{0}\}$ and $\mathbf{u} = A_{FF}^{\frac{1}{2}}\mathbf{v}$. Consider

$$\frac{\mathbf{v}^T A_{FF}\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} = \frac{\mathbf{u}^T\mathbf{u}}{\mathbf{u}^T(I+Q)^{-1}\mathbf{u}},$$

where $Q = A_{FF}^{-\frac{1}{2}}A_{FI}A_{II}^{-1}A_{IF}A_{FF}^{-\frac{1}{2}}$.

Setting $\mathbf{z} = (I+Q)^{-\frac{1}{2}}\mathbf{u}$, we have

$$\frac{\mathbf{v}^T A_{FF}\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} = \frac{\mathbf{z}^T(I+Q)\mathbf{z}}{\mathbf{z}^T\mathbf{z}} = 1 + \frac{\mathbf{z}^T Q\mathbf{z}}{\mathbf{z}^T\mathbf{z}}.$$

By Lemma 5.19, we obtain

$$1 \leq \frac{\mathbf{v}^T A_{FF}\mathbf{v}}{\mathbf{v}^T\tilde{S}_2\mathbf{v}} \leq 1+\tau^2.$$

□

Next, combining the results from Proposition 5.21 and Lemma 5.22 with Proposition 5.15 we get the spectral bounds of $P_{D2}^{-1}A$ in the following theorem.

Theorem 5.23. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and P_{D2} be the even-odd preconditioner version I defined in (5.23). Then, the spectrum of $P_{D2}^{-1}A$ satisfies*

$$\Lambda\left(P_{D2}^{-1}A\right) \subseteq \left[\left(1-\tau^4\right)\frac{2}{\gamma}, \frac{\gamma}{2}\right],$$

where $\gamma = 2+\tau^2+\tau\sqrt{4+5\tau^2}$.

Additionally, P_{D2} is an optimal preconditioner with respect to the SGFEM discretisation parameters.

Proof. By Proposition 5.21 and Lemma 5.22, we have the constants in Proposition 5.15 to be

$$\theta_1 = 1 - \tau^4,\ \Theta_1 = 1 \text{ and } \theta_2 = 1,\ \Theta_2 = 1 + \tau^2.$$

Thus, by Proposition 5.15, we obtain

$$\Lambda\left(P_{D2}^{-1}A\right) \subseteq \left[\left(1-\tau^4\right)\frac{2}{\gamma}, \frac{\gamma}{2}\right],$$

where $\gamma = 2 + \tau^2 + \tau\sqrt{4+5\tau^2}$.

□

The lower eigenvalue bound of $P_{D2}^{-1}A$ is improved from the case of mean-based preconditioner whereas the upper bound of $P_{D2}^{-1}A$ from the analysis deteriorates rapidly compared with the one for the mean-based preconditioner.

Moreover, we would like to compare even-odd preconditioner version I with the ideal preconditioner P_{D0} defined by

$$P_{D0} = \begin{bmatrix} A_{II} & \\ & S \end{bmatrix}, \tag{5.24}$$

where S denotes the Schur complement.

Theorem 5.24. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and P_{D0} be defined in (5.24). Then, the spectrum of $P_{D0}^{-1}A$ satisfies*

$$\Lambda\left(P_{D0}^{-1}A\right) \subseteq \left[\frac{\gamma'-\tau}{\gamma'+\tau}, \frac{\gamma'+\tau}{\gamma'-\tau}\right],$$

where $\gamma' = \sqrt{4-3\tau^2}$.

Additionally, P_{D0} is an optimal preconditioner with respect to the SGFEM discretisation parameters.

Proof. By Lemma 5.20 and using the fact that the eigenvalues of the inverse matrix are the inverses of the eigenvalues of the matrix, we identify the constants in Proposition 5.15 to be

$$\theta_1 = 1,\ \Theta_1 = 1 \text{ and } \theta_2 = 1,\ \Theta_2 = \frac{1}{1-\tau^2}.$$

Thus, by Proposition 5.15, we obtain

$$\Lambda\left(P_{D0}^{-1}A\right) \subseteq \left[\frac{\gamma'-\tau}{\gamma'+\tau}, \frac{\gamma'+\tau}{\gamma'-\tau}\right],$$

where $\gamma' = \sqrt{4-3\tau^2}$.

□

According to the analyses, the lower bounds of the preconditioned system with block-diagonal preconditioners based on the even-odd partition are slightly improved if $\tilde{S}$ is close to the Schur complement. In contrast, the upper bounds of the preconditioned system deteriorate significantly when τ is close to one and $\tilde{S}$ is close to the Schur complement.

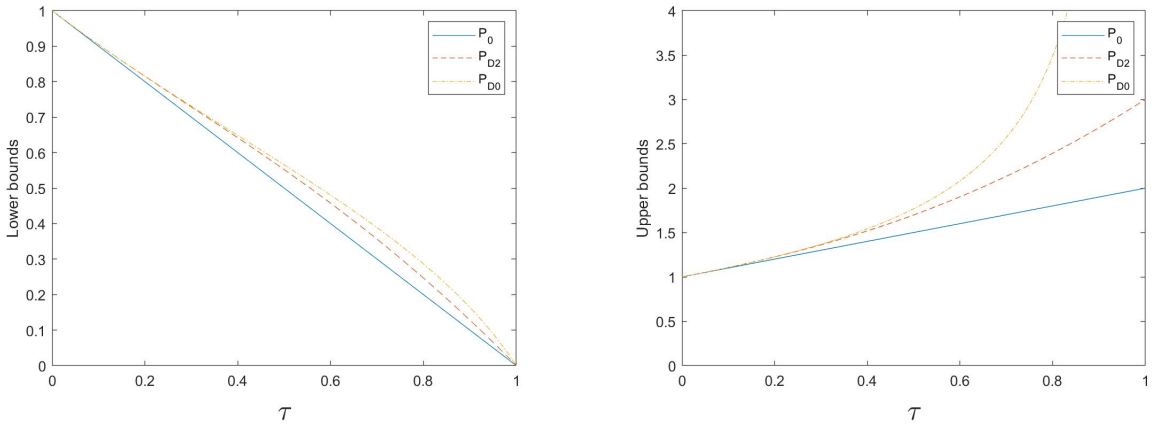


Figure 5.7: Eigenvalue bounds of the preconditioned systems $P_0^{-1}A$, $P_{D2}^{-1}A$ and $P_{D0}^{-1}A$.

Version II: Block Preconditioner with the Schur Complement Approximation $\tilde{S}_1$

In this version, we employ the block preconditioner and the Schur complement approximation $\tilde{S}_1$. Thus, we define the even-odd preconditioner version II by

$$P_{B1} = \begin{bmatrix} I & \\ A_{FI}A_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{II} & \\ & \tilde{S}_1 \end{bmatrix} \begin{bmatrix} I & A_{II}^{-1}A_{IF} \\ & I \end{bmatrix}. \tag{5.25}$$

Note that the preconditioner P_{B1} can be viewed as a symmetric block Gauss-Seidel approximation of the matrix A in (5.1). The complexities of $P_{B1}^{-1}\mathbf{r}$ are

$$\mathcal{F}\ell\left(P_{B1}^{-1}\mathbf{r}\right) = \left(N_{\boldsymbol{y}} + N_{\boldsymbol{y}_{\mathcal{P}}}\right)\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 4MN_{\boldsymbol{y}_{\mathcal{P}}}\mathrm{nnz}(K_0),$$

and

$$\mathcal{F}\ell p\left(P_{B1}^{-1}\mathbf{r}\right) = 3\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 2\mathrm{nnz}(K_0).$$

We analyse the spectrum of the preconditioned system $P_{B1}^{-1}A$ as follows.

Theorem 5.25. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and P_{B1} be the even-odd preconditioner version II defined in (5.25). Then, the spectrum of $P_{B1}^{-1}A$ satisfies*

$$\Lambda\left(P_{B1}^{-1}A\right) \subseteq \left[1-\tau^2, 1\right].$$

Additionally, P_{B1} is an optimal preconditioner with respect to the SGFEM discretisation parameters.

Proof. By Proposition 5.14, the eigenvalues for the generalised eigenvalues problem

$$A\mathbf{v} = \lambda P_{B1}\mathbf{v}$$

are identical to the eigenvalues of the matrix

$$\begin{bmatrix} I_{N_{\boldsymbol{y}_\mathcal{P}}} & \\ & \tilde{S}_1^{-1}S \end{bmatrix}. \tag{5.26}$$

Finally, by Lemma 5.20, we have

$$\Lambda\left(P_{B1}^{-1}A\right) = \{1\} \cup \Lambda\left(\tilde{S}_1^{-1}S\right) \subseteq \left[1-\tau^2, 1\right].$$

□

Version III: Block Preconditioner with the Schur Complement Approximation $\tilde{S}_2$

We define the even-odd preconditioner version III for the coefficient matrix A in (5.16) by

$$P_{B2} = \begin{bmatrix} I & \\ A_{FI}A_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{II} & \\ & \tilde{S}_2 \end{bmatrix} \begin{bmatrix} I & A_{II}^{-1}A_{IF} \\ & I \end{bmatrix}, \tag{5.27}$$

where $\tilde{S}_2 = A_{FF}\left(A_{FF} + A_{FI}A_{II}^{-1}A_{IF}\right)^{-1}A_{FF}$.

Using the results from the section 5.3.1.1 with $\mathcal{F}\ell\left(\tilde{S}_2^{-1}\mathbf{r}\right)$ and $\mathcal{F}\ell p\left(\tilde{S}_2^{-1}\mathbf{r}\right)$, the computational cost for solving the linear system with P_{B2} is

$$\mathcal{F}\ell\left(P_{B2}^{-1}\mathbf{r}\right) = \left(2N_{\boldsymbol{y}} + N_{\boldsymbol{y}_\mathcal{P}}\right)\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 8MN_{\boldsymbol{y}_\mathcal{P}}\mathrm{nnz}(K_0),$$

and

$$\mathcal{F}\ell p\left(P_{B2}^{-1}\mathbf{r}\right) = 5\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + 4\mathrm{nnz}(K_0).$$

The spectral analysis of P_{B2} is provided in the following theorem.

Theorem 5.26. *Let the coefficient matrix A have the 2-by-2 block structure in (5.16) and P_{B2} be the even-odd preconditioner version III defined in (5.27). Then, the spectrum of*

$P_{B2}^{-1}A$ satisfies

$$\Lambda\left(P_{B2}^{-1}A\right) \subseteq \left[1-\tau^4, 1\right].$$

Additionally, P_{B2} is an optimal preconditioner with respect to the SGFEM discretisation parameters.

Proof. By Proposition 5.14, the eigenvalues for the generalised eigenvalue problem

$$A\mathbf{v} = \lambda P_{B2}\mathbf{v}$$

are identical to the eigenvalues of the matrix

$$\begin{bmatrix} I_{N_{\boldsymbol{y}_{\mathcal{P}}}} & \\ & \tilde{S}_2^{-1}S \end{bmatrix}. \tag{5.28}$$

It is obvious that $1 \in \Lambda\left(P_{B2}^{-1}A\right)$ is a repeated eigenvalue and $\Lambda\left(\tilde{S}_2^{-1}S\right) \subseteq \Lambda\left(P_{B2}^{-1}A\right)$. Hence, we have that

$$\Lambda\left(P_{B2}^{-1}A\right) = \{1\} \cup \Lambda\left(\tilde{S}_2^{-1}S\right).$$

The result follows by applying Proposition 5.21.

□

5.5 Numerical Experiments

In this section, we present the numerical experiments for the even-odd preconditioners. Firstly, the sizes of the sets of even nodes and odd nodes in $\mathbb{I}_k^M$ are observed. We compare the performance of the even-odd preconditioners with other preconditioners such as the mean-based preconditioner, the Kronecker product preconditioner and the modified truncation preconditioners. The test problems in this section are chosen to be the case of fast decay in Example 4.1 and the test problem in Example 4.2. The spectral bounds and optimality of the preconditioned system are investigated to confirm our analysis.

It is worth recalling that

$$\mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right) < \left\{ \begin{array}{l} \mathcal{F}\ell\left(P_{\otimes}^{-1}\mathbf{r}\right) \\ \mathcal{F}\ell\left(P_{B1}^{-1}\mathbf{r}\right) \end{array} \right\} < \mathcal{F}\ell\left(\tilde{P}_r^{-1}\mathbf{r}\right),$$

and

$$\mathcal{F}\ell\left(\tilde{P}_r^{-1}\mathbf{r}\right) \approx \mathcal{F}\ell\left(P_{D2}^{-1}\mathbf{r}\right) < \mathcal{F}\ell\left(P_{B2}^{-1}\mathbf{r}\right),$$

by assuming that $\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right)$ dominates the cost of one PCG iteration.

We begin by observing the proportions of $\mathbb{I}_{\mathcal{P}}$ to $\mathbb{I}_k^M$ for $M = 1, ..., 8$ and $k = 1, ..., 6$. These numbers are important because the complexity per iteration by PCG depends on the size of $\mathbb{I}_{\mathcal{P}}$. Recall that one PCG iteration by the even-odd preconditioners requires to solve $2N_{\boldsymbol{y}}$, $N_{\boldsymbol{y}} + N_{\boldsymbol{y}_{\mathcal{P}}}$ and $2N_{\boldsymbol{y}} + N_{\boldsymbol{y}_{\mathcal{P}}}$ linear systems with coefficient matrix K_0 for version I, II and III, respectively. For fixed k and M, define $\mathbb{I}_E$ and $\mathbb{I}_O$ to be the sets of even multi-indices and odd multi-indices, respectively, i.e.,

$$\mathbb{I}_E = \left\{\boldsymbol{\alpha} \in \mathbb{I}_k^M \mid |\boldsymbol{\alpha}| \text{ is even.}\right\} \text{ and } \mathbb{I}_O = \left\{\boldsymbol{\alpha} \in \mathbb{I}_k^M \mid |\boldsymbol{\alpha}| \text{ is odd.}\right\}.$$

Since the spectral analyses of the three versions of the even-odd preconditioners say that they are optimal, i.e., the numbers of PCG iterations are bounded, $\mathbb{I}_{\mathcal{P}}$ should be selected to be the smaller set between $\mathbb{I}_E$ and $\mathbb{I}_O$ to reduce the cost per PCG iteration. The other set is set to be $\mathbb{I}_{\mathcal{I}}$. Thus, it is obvious that the ratios between the size of $\mathbb{I}_{\mathcal{P}}$ and $N_{\boldsymbol{y}}$ are bounded between 0 and 0.5. For instance, in the case of $M = 1$, if k is odd, then $\#\mathbb{I}_E = \#\mathbb{I}_O$. As a result, $N_{\boldsymbol{y}_{\mathcal{P}}}/N_{\boldsymbol{y}} = 0.5$. On the other hand, if k is even, then $\#\mathbb{I}_E = \#\mathbb{I}_O + 1$. Thus, $N_{\boldsymbol{y}_{\mathcal{P}}}/N_{\boldsymbol{y}} = \#\mathbb{I}_O/\left(\#\mathbb{I}_O + \#\mathbb{I}_E\right) = 0.5k/(k+1)$. These ratios are illustrated in Figure 5.8 with $1 \leq k \leq 6$ and $1 \leq M \leq 8$.

To compare the performance of the even-odd preconditioners with the other preconditioners in later experiments, the sizes of $\mathbb{I}_O$ and $\mathbb{I}_E$ for $M = 8$ with $k = 1, ..., 6$ and also the ratio between $N_{\boldsymbol{y}_{\mathcal{P}}}$ and $N_{\boldsymbol{y}}$ are observed in Table 5.1. According to Figure 5.8 and Table 5.1, we can see that the computational cost tends to increase with the parameter k

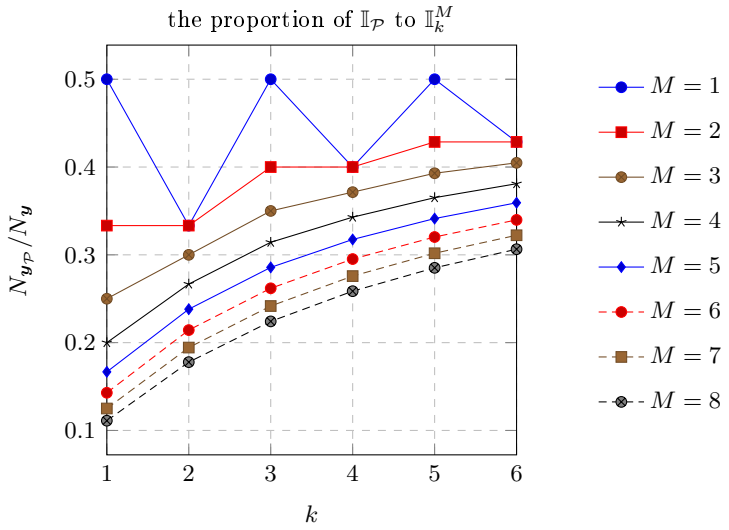


Figure 5.8: The proportion of the set $\mathbb{I}_{\mathcal{P}}$ to $\mathbb{I}_k^M$ with the parameters M and k, where $\mathbb{I}_{\mathcal{P}}$ is chosen to be the smaller set between $\mathbb{I}_E$ and $\mathbb{I}_O$.

k	1	2	3	4	5	6
$\#\mathbb{I}_E$	1	37	37	367	367	2083
$\#\mathbb{I}_O$	8	8	128	128	920	920
$N_{\boldsymbol{y}_{\mathcal{P}}}/N_{\boldsymbol{y}}$	0.1111	0.1778	0.2242	0.2586	0.2852	0.3064

Table 5.1: The numbers of even multi-indices and odd multi-indices in $\mathbb{I}_k^M$ for $M = 8$ and k ranging from 1 to 6.

whereas increasing M reduces the cost for one PCG iteration.

Thus, the costs per PCG iteration for P_{B1} and P_{B2} depend on the parameter k and M. For instance, according to Table 5.1, for $M = 8$, $k = 3$, we have $N_{\boldsymbol{y}_{\mathcal{P}}}/N_{\boldsymbol{y}} \approx 0.2242$ and then $\mathcal{F}\ell\left(P_{B1}^{-1}\mathbf{r}\right) \approx 1.2242\mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right)$ and $\mathcal{F}\ell\left(P_{B2}^{-1}\mathbf{r}\right) \approx 2.2242\mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right)$.

All the experiments were implemented in S-IFISS. In each case we solved for the variational formulation (2.11) with the diffusion coefficient $a(\boldsymbol{x}, \boldsymbol{y})$ assumed to be as in (4.2) and satisfying conditions (4.2) and (4.3). The forcing function f is set to be $f(\boldsymbol{x}) = 1$. Recall that the spaces $L^2_\rho(\boldsymbol{\Gamma})$ and $H^1_0(D)$ are discretised by the space of complete polynomials S_k^M and the space of continuous piecewise linear functions X_h, respectively. The number of parameters for the space S_k^M ranged from 1 to 8 ($M = 1, ..., 8$) with the degree 1 to 6 ($k = 1, ..., 6$) and the mesh size ranges from 2^{-4} to as fine as 2^{-7}. The linear system, which is obtained by SGFEM, is solved using PCG for the symmetric preconditioners, i.e., P_0, $\tilde{P}_1$, $\tilde{P}_2$, P_{D2}, P_{B1}, P_{B2}, $P_\otimes$, with initial guess $\mathbf{x}_0 = \mathbf{0}$ and stopping

		$P_{\otimes}$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	P_{D2}	P_{B1}	P_{B2}
$k =$	1	12	12	7	6	13	7	4
	2	16	16	8	7	17	8	6
	3	20	21	9	9	22	11	7
	4	24	24	10	9	25	12	8
	5	26	27	11	10	28	14	9
	6	29	29	12	11	30	14	10

Table 5.2: PCG iteration counts for Example 5.6.

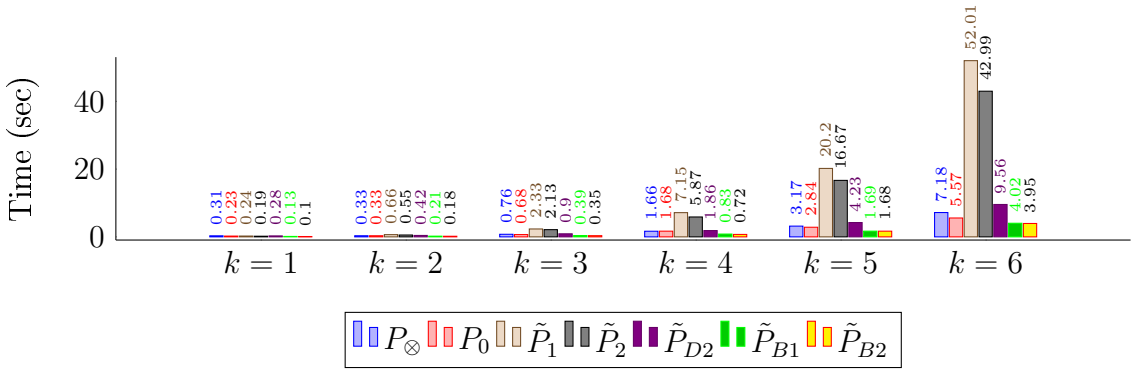


Figure 5.9: PCG runtimes (in seconds) for Example 5.6.

criterion given by the relative residual norm being brought below $tol = 10^{-6}$.

Example 5.6. In this experiment, the diffusion coefficient a is assumed to be as given in Example 4.1 but here, we consider only the case of fast decay, i.e., $\tilde{\sigma} = 4$ and $\bar{\alpha} = 0.9239$. These settings give $\tau = 0.9999$ and $\tau_8 = 0.9995$. We set the discretisation parameters to be $h = 2^{-4}$, $M = 8$ and vary k from 1 to 6.

We compare the performance of the mean-based preconditioner, the Kronecker product preconditioner, the three versions of the even-odd preconditioners and also the modified truncation preconditioner. For the later, only $\tilde{P}_1$ and $\tilde{P}_2$ are included. By the spectrum analysis, we expect P_{B2} outperform the others in terms of the PCG iteration counts. This is confirmed in Table 5.2 and Figure 5.9.

It is clear that the even-odd preconditioner versions II and III are more efficient than the mean-based preconditioner and the Kronecker product preconditioner whereas the iteration counts by P_{D2} are in the same level as the mean-based preconditioner and Kronecker product preconditioner. Moreover, in most cases, the performance by P_{B2} is still

	$k=1$		$k=2$		$k=3$	
	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$
$P_{\otimes}^{-1}A$	0.4902	1.5498	0.3042	1.7491	0.2160	1.8408
$P_0^{-1}A$	0.4718	1.5281	0.2886	1.7113	0.2036	1.7963
$\tilde{P}_1^{-1}A$	0.7210	1.0339	0.5750	1.0481	0.4675	1.0561
$\tilde{P}_2^{-1}A$	0.7210	1.0066	0.5810	1.0096	0.4803	1.0114
$P_{D2}^{-1}A$	0.5261	1.7528	0.3441	2.1619	0.2508	2.3833
$P_{B1}^{-1}A$	0.7211	1.0000	0.4940	1.0000	0.3659	1.0000
$P_{B2}^{-1}A$	0.9222	1.0000	0.7440	1.0000	0.5979	1.0000

Table 5.3: The eigenvalue bounds of preconditioned system $P_{D2}^{-1}A$, $P_{B1}^{-1}A$ and $P_{B2}^{-1}A$ for $k = 1, 2, 3$ and $\tau_M = 0.9995$.

superior to that of the modified truncation preconditioner, except for the case $k = 6$ where the overall complexity of $\tilde{P}_2$ is lower than the one of P_{B2} because the complexity per PCG iteration of P_{B2} increases with the parameter k. Furthermore, although the iteration counts by P_{B1} are higher than the ones by $\tilde{P}_r$ and P_{B2}, in term of overall performance, P_{B1} is the most efficient in every case due to the low cost per PCG iteration. However, the experiment shows that solving the linear systems with P_{B1} or P_{B2} consume significantly less time than other preconditioners. In addition, Table 5.2 shows the mild dependence of the three versions of even-odd preconditioners on the parameter k.

We also observe the eigenvalue bounds for the cases $k = 1, 2, 3$. These bounds are presented in Table 5.3. We can see that the upper bounds by P_{B1} and P_{B2} remain stable for all $k = 1, 2, 3$. In fact, they are identical to those in Theorem 5.26 but there are significant decreases in lower bounds of eigenvalue due to τ_M being very close to 1. Nevertheless, the spectrum of the preconditioned system by P_{B2} is tighter than the other preconditioned systems whereas the eigenvalue bounds for P_{D2} are not as tight as the ones for the mean-based preconditioner. This is consistent with Table 5.2 where the iteration counts by P_{B2} are the lowest for all k. Note also that the eigenvalue bounds for the even-odd preconditioner version II and III can be improved if we have sharp eigenvalue bounds for the mean-based preconditioner. For example, according to Table 5.3 when $k = 1$, we have the eigenvalue bounds for the mean-based preconditioner to be $[1 - \tau_{M,k}, 1 + \tau_{M,k}]$, where $\tau_{8,1} = 0.5282$ with $M = 8$ and $k = 1$. This $\tau_{8,1}$ gives the lower eigenvalue bound for

		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	P_{D2}	P_{B1}	P_{B2}
$k =$	1	6	5	6	9	5	3	2
	2	7	8	7	7	8	4	3
	3	8	9	8	8	9	5	3
	4	9	10	9	8	10	5	4
	5	10	11	9	9	11	6	4
	6	10	11	10	9	12	6	4

Table 5.4: PCG iteration counts for Example 5.7.

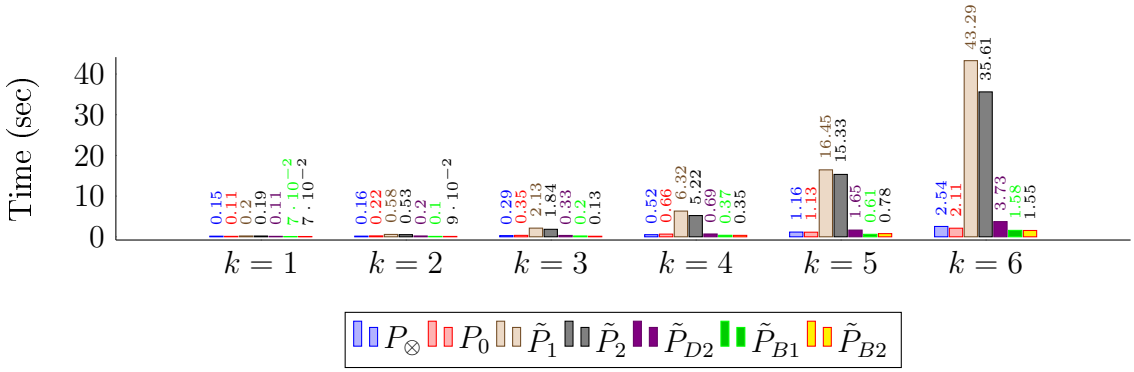


Figure 5.10: PCG runtimes (in seconds) for Example 5.7.

P_{B1} and P_{B2} to be $1 - \tau_{8,1}^2 \approx 0.7211$ and $1 - \tau_{8,1}^4 \approx 0.9222$, respectively, which are very close to the bounds we obtain from the experiment.

Next, we will observe the results by the three versions of even-odd preconditioners with Example 4.2.

Example 5.7. We set the diffusion coefficient a as in Example 4.2 with $\tau_8 = 0.8054$. The discretisation parameters are set to be $M = 8$, $h = 2^{-4}$ and k ranging from 1 to 6. Again, the performance of P_0, $P_\otimes$, P_{D2}, P_{B1}, P_{B2} and $\tilde{P}_r$ for $r = 1, 2$ are compared and the results are displayed in Table 5.4 and Figure 5.10.

The experiment shows that the even-odd preconditioners version II and III outperform the other preconditioners in terms of PCG iteration counts, runtimes and also overall performance except the first version of even-odd preconditioner. Since τ_8 is not close to 1, by Theorem 5.26, the spectral bounds for the even-odd preconditioner version II and III are clustered around one as shown in Table 5.5. Again, the upper bounds of preconditioned systems are constant at one for all $k = 1, 2, 3$ but the lower bounds deteriorates slowly

		$P_{D2}^{-1}A$		$P_{B1}^{-1}A$		$P_{B2}^{-1}A$	
		$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$	$\lambda_{\min}$	$\lambda_{\max}$
$k=$	1	0.8256	1.2098	0.9647	1.0000	0.9988	1.0000
	2	0.7250	1.3674	0.9078	1.0000	0.9915	1.0000
	3	0.6529	1.4972	0.8502	1.0000	0.9778	1.0000

Table 5.5: Eigenvalue bounds for the preconditioned systems $P_{D2}^{-1}A$, $P_{B1}^{-1}A$ and $P_{B2}^{-1}A$ for $k = 1, 2, 3$ and $\tau_M = 0.8054$.

h	Example 5.6						Example 5.7					
	$M=4$			$M=8$			$M=4$			$M=8$		
	P_{D2}	P_{B1}	P_{B2}	P_{D2}	P_{B1}	P_{B2}	P_{D2}	P_{B1}	P_{B2}	P_{D2}	P_{B1}	P_{B2}
2^{-3}	18	9	7	18	9	8	9	4	3	9	5	3
2^{-4}	22	10	8	22	10	8	9	4	2	9	5	3
2^{-5}	23	11	8	23	11	8	9	4	3	9	5	3
2^{-6}	25	12	8	25	12	8	9	4	3	9	5	3
2^{-7}	25	12	8	25	12	8	9	4	3	9	5	3

Table 5.6: The iteration counts by the even-odd preconditioners when h and M are varied.

with k in the case of P_{B1} and P_{B2}. This results in a significant decrease in the number of iterations for the other preconditioners which is consistent with the spectral analysis. On the other hand, the eigenvalue bounds for P_{D2} are not as tight as the bounds for P_{B1} or P_{B2}. As a result, the iteration counts for P_{D2} are higher than those in P_{B1} and P_{B2}.

Finally, as we have seen that the even-odd preconditioners are optimal with respect to the parameter k, the following experiment aims to confirm optimality of the even-odd preconditioners with respect to the parameters M and k.

Example 5.8. There are two cases in this experiment. One is the diffusion coefficient a in Example 5.6 and another one is in Example 5.7. The parameter k is fixed at $k = 3$ with M ranging from 4 to 8 and h from 2^{-3} to 2^{-7}. The iteration counts by the three versions of the even-odd preconditioners for both test cases are shown in Table 5.6.

We can see that the numbers of iterations by the even-odd preconditioners are virtually constant in both cases with different h and M. This shows that the even-odd preconditioners are also optimal preconditioners for the stochastic Galerkin matrix A in (5.16).

To conclude, in this chapter, we introduced a domain decomposition technique on

the parametric domain for the affine-parametric diffusion coefficients. We introduced the concept of parametric mesh and then partitioned the mesh into submeshes. Moreover, we presented three versions of the even-odd preconditioners based on the even-odd partition. The numerical experiments indicate that version II and version III of the even-odd preconditioners are more efficient than the others. In addition, the even-odd preconditioners are optimal with respect to discretisation parameters.

CHAPTER 6

BLOCK PRECONDITIONERS FOR SPDES WITH NON-AFFINE PARAMETRIC COEFFICIENTS

Elliptic problems with non-affine parametric diffusion coefficient are very challenging problems due to the representation of the diffusion coefficient a. In this chapter, the diffusion coefficient a is assumed to be expanded by a generalised polynomial chaos expansion. After applying the SGFEM to the problem, we obtain a linear system with system matrix A written as a sum of Kronecker products as in (2.20), i.e.,

$$A = \sum_{\boldsymbol{\alpha} \in \mathbb{I}_{2k}^{M}} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}.$$

Recall that the matrix A is symmetric and positive definite but block dense. Thus, devising a preconditioning technique for this type of problem is very challenging.

The mean-based preconditioner and Kronecker product preconditioner are two possible choices to tackle this problem. The structure of these two preconditioners is in the Kronecker product form, whose the right Kronecker product factor is chosen to be the stiffness matrix $K_{\mathbf{0}}$. Our experiments with truncation preconditioners for affine diffusion coefficients have shown that the other stiffness matrices in the coefficient matrix A are vital to improving the convergence rate of PCG. Moreover, the matrices $G_{\boldsymbol{\alpha}}$ are useful to change the pattern of the coefficient matrix to a 2-by-2 block matrix. This new pattern is utilised to design preconditioners as we have seen in the case of even-odd preconditioners

(see section 5.3).

Our aim is to design preconditioners for non-affine parametric coefficients by combining the ideas used in the design of truncation preconditioners and of domain decomposition preconditioners for affine parametric coefficients. Recall that the domain decomposition technique is a permutation technique to generate the 2-by-2 block matrix structure of the system matrix. The main challenge to design an efficient preconditioner for this problem is that the system matrix is block dense due to the representation of the diffusion coefficients. Thus, no permutation can achieve the structure in (5.1). To obtain a block-sparse preconditioner, we truncate the diffusion coefficient to capture its main feature. We use this truncated coefficient to design a preconditioner, and it leads to a sparse version of the system matrix, which we can use as a preconditioner. As a result, we apply the domain decomposition technique to the sparse preconditioner instead of the dense system matrix.

The outline of this chapter is as follows. We review non-affine parametric diffusion coefficients in section 6.1. We generalise the truncation preconditioners for affine-parametric diffusion coefficients to the case of non-affine parametric diffusion coefficients in section 6.2. Then the practical preconditioners and their computational cost are presented in section 6.3. In section 6.4 we introduce the parametric mesh for non-affine diffusion coefficients and describe how to partition it. We also present another technique to approximate the truncation preconditioners based on the partitioning and discuss its complexity. Next, the preconditioners for log-transformed diffusion coefficients are introduced in section 6.5, and we end this chapter with numerical results in section 6.6.

6.1 Non-affine Parametric Diffusion Coefficients

Assume the diffusion coefficient a to be non-affine parametric, i.e., not all $\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})$ for $\boldsymbol{\alpha} \in \mathbb{I}$ are linear functions, and a can be written as

$$a(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \mathbb{I}} a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{y}),$$

where $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha} \in \mathbb{I}}$ is an orthonormal polynomial basis of $L^2(\boldsymbol{\Gamma})$. Also, assume that the coefficient a satisfies condition (2.10). Note that the mean of the coefficient a correspond to the term with $\boldsymbol{\alpha} = \mathbf{0} \in \mathbb{I}$. This leads to the mean of the coefficient a, i.e., $a_{\mathbf{0}}$, to be bounded and positive. Thus, there exist positive real numbers $a_{\mathbf{0}}^{\min}$ and $a_{\mathbf{0}}^{\max}$ such that

$$0 < a_{\mathbf{0}}^{\min} \leq a_{\mathbf{0}}(\boldsymbol{x}) \leq a_{\mathbf{0}}^{\max}, \qquad \text{for all } \boldsymbol{x} \in D. \tag{6.1}$$

After we apply the SGFEM to the variational formulation (2.11), we obtain a linear system with coefficient matrix

$$A = \sum_{\boldsymbol{\alpha} \in \mathbb{I}_{2k}^M} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}},$$

where $G_{\boldsymbol{\alpha}}$ and $K_{\boldsymbol{\alpha}}$ are defined by

$$\begin{aligned} [G_{\boldsymbol{\alpha}}]_{js} &= \left\langle \psi_{\boldsymbol{\alpha}} \psi_{q(j)}, \psi_{q(s)} \right\rangle_{\rho}, && j, s = 1, 2, \ldots, N_{\boldsymbol{y}}, \\ [K_{\boldsymbol{\alpha}}]_{ir} &= \int_D a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \nabla \phi_i(\boldsymbol{x}) \cdot \nabla \phi_r(\boldsymbol{x}) d\boldsymbol{x}, && i, r = 1, 2, \ldots, N_{\boldsymbol{x}}, \end{aligned} \tag{6.2}$$

for $\boldsymbol{\alpha} \in \mathbb{I}_{2k}^M$.

Note that the matrix A is symmetric and positive definite. Since the matrix A is a summation over the set $\mathbb{I}_{2k}^M$, it leads to the matrix A being block-dense although the matrices $G_{\boldsymbol{\alpha}}$ are all sparse.

Furthermore, there are some cases where the solution of the variational formulation (2.11) uniquely exists although the diffusion coefficient a is not bounded away from zero

or from above. This means that there do not exist positive numbers $a_{\min}$ and $a_{\max}$ such that condition (2.10) holds. For example, let a be a lognormally distributed random field, i.e.,

$$a(\boldsymbol{x},\omega) = \exp\left(b_0 + \sum_{m=1}^{N} b_m(\boldsymbol{x}) Y_m(\omega)\right), \qquad (\boldsymbol{x},\omega) \in D \times \Omega,$$

where $Y_m(\omega) \in \Gamma_m \coloneqq (-\infty, \infty)$ for all $m = 1, 2, ..., N$. We can see that the coefficient a is positive but unbounded. However, we can prove the existence and uniqueness of the solution to the variational formulation (2.11) (see [34] and [9, Lemma 1.2]).

6.2 Truncation Preconditioners For Non-affine Diffusion Coefficients

Our truncation preconditioners for affine diffusion coefficients have shown that the matrices K_m are important to improve the rate of convergence of the solver. To design a preconditioner for non-affine coefficients, a sparse gPC expansion such as in those [24, 25] is needed. Essentially, we generalise the idea of truncation preconditioner for affine parametric coefficients. That is, we aim to find an approximation of the diffusion coefficients a by choosing a set of multi-indices $\tilde{\mathbb{I}} \subseteq \mathbb{I} \backslash \{\mathbf{0}\}$; then a can be approximated by

$$\tilde{a}(\boldsymbol{x},\boldsymbol{y}) = a_{\mathbf{0}}(\boldsymbol{x}) + \sum_{\boldsymbol{\alpha}\in\tilde{\mathbb{I}}} a_{\boldsymbol{\alpha}}(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}). \tag{6.3}$$

Therefore, we reorder the terms of a based on magnitudes $\|a_{\boldsymbol{\alpha}}\|_{L^\infty(D)}$ but always start with $a_{\mathbf{0}}$. Suppose $\mathbb{I}_r \subseteq \mathbb{I}_{2k}^M \backslash \{\mathbf{0}\}$, where $\#\mathbb{I}_r \ll \#\mathbb{I}_{2k}^M - 1$, is the set of multi-indices of the first r terms after the reordering. Define a_r by

$$a_r(\boldsymbol{x},\boldsymbol{y}) = a_{\mathbf{0}}(\boldsymbol{x}) + \sum_{\boldsymbol{\alpha}\in\mathbb{I}_r} a_{\boldsymbol{\alpha}}(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}).$$

Note that a_r is not necessarily positive on $D \times \boldsymbol{\Gamma}$.

Next, define a bilinear form $B_r : V \times V \to \mathbb{R}$ by

$$B_r(u,v) = \int_{\mathbf{\Gamma}} \rho(\boldsymbol{y}) \int_D a_r(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}.$$

Consequently, the bilinear form B_r induces the truncation preconditioner P_r via

$$P_r = G_{\mathbf{0}} \otimes K_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}.$$

Since we cannot guarantee that $a_r(\boldsymbol{x},\boldsymbol{y})$ is positive for all $(\boldsymbol{x},\boldsymbol{y}) \in D \times \mathbf{\Gamma}$, the preconditioner P_r is symmetric but may not be positive definite.

6.3 Modified Truncation Preconditioners for Non-affine Diffusion Coefficients

Since PCG requires the preconditioner to be symmetric and positive definite, we need to find a symmetric and positive definite matrix based on the above matrix P_r which is likely to be symmetric and indefinite. In this section, we modify P_r by using its symmetric block Gauss-Seidel approximation. Define $\tilde{P}_r$ to be the symmetric block Gauss-Seidel approximation of P_r, i.e.,

$$\tilde{P}_r = (D + L)\, D^{-1} \left(D + L^T\right),$$

where D is the block-diagonal matrix of P_r and L is the strictly lower block-triangular matrix of P_r.

To ensure that the modified version of P_r is symmetric and positive definite, it is required only that the blocks along the main diagonal of P_r, i.e., D, are symmetric and positive definite.

In the following, we introduce an additional assumption on the diffusion coefficient a.

Let

$$\mathbb{I}_e = \{\boldsymbol{\alpha} \in \mathbb{I}_r \mid \alpha_m \text{ is even for all } m = 1, ..., M\}.$$

Assume that $a_{\boldsymbol{\alpha}}$ is non-negative and bounded on the domain D for any $\boldsymbol{\alpha} \in \mathbb{I}_e$. That is, for $\boldsymbol{\alpha} \in \mathbb{I}_e$,

$$a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \geq 0, \quad \text{for all } \boldsymbol{x} \in D. \tag{6.4}$$

The assumption (6.4) causes the stiffness matrices $K_{\boldsymbol{\alpha}}$ to be symmetric and positive semi-definite for all $\boldsymbol{\alpha} \in \mathbb{I}_e$. With assumption (6.4), our goal is to show that the preconditioner P_r can be written as

$$P_r = D + L + L^T, \tag{6.5}$$

where D is the block-diagonal matrix whose blocks are symmetric and positive definite and L is the strictly lower block-triangular matrix.

To show that P_r can be split as in (6.5), we need to refer to the following corollary.

Corollary 6.1 ([51, Corollary 24]). *Let P_i be an orthonormal polynomial of degree i generated by an even probability of density function. Let $i, j, k \in \mathbb{N}_0$. Then, $\langle P_k P_i, P_j \rangle_\rho = 0$ if one of the following condition holds*

1. $|i - j| > k$,
2. $k > i + j$,
3. $i + j + k$ *is an odd number.*

In addition, it is reasonable to assume

$$\left\langle P_k^m, (P_i^m)^2 \right\rangle_{\rho_m} \geq 0, \qquad \text{for } k, i \in \mathbb{N}_0 \text{ and } m = 1, ..., M. \tag{6.6}$$

This assumption holds, for example, if P_k is a Hermite polynomial or a Legendre polynomial (see [51, Appendix A] for other orthogonal polynomials). The following result shows that the main diagonal of P_r is symmetric and positive definite.

Lemma 6.2. *Assume conditions (6.6) and (6.4) hold. Then, the preconditioner P_r can be represented as in (6.5), where D is symmetric and positive definite.*

Proof. Let $\boldsymbol{\alpha} \in \mathbb{I}_r \backslash \mathbb{I}_e$ and $\boldsymbol{\beta} \in \mathbb{I}_k^M$. Consider

$$\left\langle \psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}^2 \right\rangle = \prod_{m=1}^{M} \left\langle P_{\alpha_m}^m, \left(P_{\beta_m}^m\right)^2 \right\rangle_{\rho_m}.$$

Since $\boldsymbol{\alpha} \notin \mathbb{I}_e$, there exists an odd index α_{m^*}. Thus, $\alpha_{m^*} + \beta_{m^*} + \beta_{m^*}$ is odd. By Corollary 6.1, we have $\left\langle P_{\alpha_{m^*}}^{m^*}, \left(P_{\beta_{m^*}}^{m^*}\right)^2 \right\rangle_{\rho_{m^*}} = 0$ and then, $\left\langle \psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}^2 \right\rangle = 0$. As a result, all the blocks along the main diagonal of $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$ are zero matrices. In other words, $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$ can be written as

$$G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}} = L_{\boldsymbol{\alpha}} + L_{\boldsymbol{\alpha}}^T,$$

where $L_{\boldsymbol{\alpha}}$ is a strictly lower block-triangular matrix of $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$.

On the other hand, if $\boldsymbol{\alpha} \in \mathbb{I}_r \cap \mathbb{I}_e$, then the main diagonal of $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$ is $\left\langle \psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}^2 \right\rangle K_{\boldsymbol{\alpha}}$, for $\boldsymbol{\beta} \in \mathbb{I}_k^M$. Since $\left\langle P_{\alpha_m}^m, \left(P_{\beta_m}^m\right)^2 \right\rangle_{\rho_m} \geq 0$ for all $m = 1, ..., M$ and $K_{\boldsymbol{\alpha}}$ is symmetric and positive semi-definite, then $\left\langle \psi_{\boldsymbol{\alpha}}, \psi_{\boldsymbol{\beta}}^2 \right\rangle K_{\boldsymbol{\alpha}}$ is also a symmetric and positive semi-definite matrix for all $\boldsymbol{\beta} \in \mathbb{I}_k^M$. Hence, the matrix $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$ can be written as

$$G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}} = D_{\boldsymbol{\alpha}} + L_{\boldsymbol{\alpha}} + L_{\boldsymbol{\alpha}}^T,$$

where $D_{\boldsymbol{\alpha}}$ is block-diagonal, symmetric and positive semi-definite and L is strictly lower block-triangular matrix of $G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}$.

Therefore,

$$P_r = \left(G_{\mathbf{0}} \otimes K_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r \cap \mathbb{I}_e} D_{\boldsymbol{\alpha}} \right) + \left(\sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} L_{\boldsymbol{\alpha}} \right) + \left(\sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} L_{\boldsymbol{\alpha}} \right)^T,$$

where $G_{\mathbf{0}} \otimes K_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r \cap \mathbb{I}_e} D_{\boldsymbol{\alpha}}$ is block-diagonal and symmetric and positive definite.

□

Recall that

$$\tilde{P}_r = (D + L)\, D^{-1} \left(D + L^T\right).$$

Since D is symmetric and positive definite, so is $\tilde{P}_r$.

Although P_r is a symmetric and indefinite matrix, the blocks on its main diagonal of P_r are symmetric and positive definite. We can utilise its main diagonal to construct a suitable preconditioner, i.e., symmetric and positive definite. Thus, we can use $\tilde{P}_r$ as a preconditioner for non-affine parametric cases.

6.3.1 Computational Cost

At each PCG iteration, we need to solve a linear system with $\tilde{P}_r$. Let $\mathbf{r} \in \mathbb{R}^{N_{\boldsymbol{x}} N_{\boldsymbol{y}}}$. Consider the linear system

$$\tilde{P}_r \mathbf{z} = \mathbf{r}. \tag{6.7}$$

The following procedure may be used to obtain the solution $\mathbf{z}$ of linear system (6.7).

1. Solve for $\mathbf{z}_1$

$$(D + L)\mathbf{z}_1 = \mathbf{r}.$$

2. Solve for $\mathbf{z}$

$$(D + L^T)\mathbf{z} = D\mathbf{z}_1.$$

Since the pattern and the numbers of non-zero entries in the matrices $G_{\boldsymbol{\alpha}}$ are different from the case of affine-parametric diffusion coefficients, to determine the complexity of the action of $\tilde{P}_r^{-1}$ on the vector $\mathbf{r}$, the number of non-zero entries in these matrices is needed.

Lemma 6.3 ([51, Lemma 28]). *For even probability density functions, the number of non-zero entries of the matrix $G_{\boldsymbol{\alpha}}$ defined in (6.2) is bounded by*

$$\mathrm{nnz}(G_{\boldsymbol{\alpha}}) \leq 2\frac{k}{M+k} N_{\boldsymbol{y}}.$$

Each step requires to solve $N_{\boldsymbol{y}}$ linear systems with stiffness matrices in the matrix D and the number of multiplication depends on non-zero block matrices in the preconditioner P_r. Thus. the sequential complexity of $\tilde{P}_r^{-1}\mathbf{r}$ is bounded as

$$\mathcal{F}\ell\left(\tilde{P}_r^{-1}\mathbf{r}\right) \leq 2N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + \frac{2kr}{M+k}N_{\boldsymbol{y}}\mathrm{nnz}(K_{\mathbf{0}}).$$

On the other hand, the parallel complexity of $\tilde{P}_r^{-1}\mathbf{r}$ is not obvious because the pattern of P_r does not facilitate parallelism.

6.4 Domain Decomposition Preconditioners For Non-affine Parametric Diffusion Coefficients

Since the stochastic Galerkin matrix A is block dense, any permutation of orthogonal polynomials cannot ensure the A_{II} in (5.1) has a block diagonal structure. So we cannot apply the domain decomposition technique to the matrix A directly but we can apply it to the block sparse preconditioner instead. In this section, we want to partition the preconditioner P_r as a 2-by-2 block matrix, i.e.,

$$P_r = \begin{bmatrix} P_{II} & P_{IF} \\ P_{FI} & P_{FF} \end{bmatrix},$$

where P_{II} is a non-singular block diagonal matrix. This can be done by extending the idea of even-odd partitioning for affine parametric coefficients to non-affine cases. We start by studying the pattern of the preconditioner P_r. Next, we introduce the parametric mesh associated with $\mathbb{I}_k^M$ for the case of non-affine parametric diffusion coefficients. We then partition the parametric mesh to many small submeshes and obtain a 2-by-2 block structure of P_r.

6.4.1 Parametric Mesh

We represent P_r in block form by

$$[P_r]_{js} = [G_{\mathbf{0}}]_{js} K_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} [G_{\boldsymbol{\alpha}}]_{js} K_{\boldsymbol{\alpha}}, \qquad j, s = 1, 2, \ldots, N_{\boldsymbol{y}} \ .$$

Recall that we can observe the sparsity pattern of the preconditioner P_r by considering the pattern of the matrix

$$G_r = G_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} G_{\boldsymbol{\alpha}}.$$

Theorem 6.4. *Let q be a bijection from $\{1, 2, \ldots, N_{\boldsymbol{y}}\}$ to $\mathbb{I}_k^M$. Let $j, s \in \{1, 2, \ldots, N_{\boldsymbol{y}}\}$ and $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_k^M$ such that $\boldsymbol{\beta} = q(j)$ and $\boldsymbol{\beta}' = q(s)$. If $[G_r]_{js} \neq \mathbf{0}$, then one of the following conditions holds*

1. *The multi-indices $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ are identical, i.e.,*

$$\boldsymbol{\beta} = \boldsymbol{\beta}'. \tag{6.8}$$

2. *There exists $\boldsymbol{\alpha} \in \mathbb{I}_r$ such that*

$$|\beta_m - \beta'_m| \leq \alpha_m \leq \beta_m + \beta'_m \text{ and } \alpha_m + \beta_m + \beta'_m \text{ is even for all } m = 1, ..., M. \tag{6.9}$$

Proof. Suppose conditions (6.8) and (6.9) do not hold. If condition (6.8) does not hold, then $j \neq s$ and we must have $[G_{\mathbf{0}}]_{js} = 0$ because $G_{\mathbf{0}}$ is the identity matrix. If condition (6.9) does not hold, for any $\boldsymbol{\alpha} \in \mathbb{I}_r$, there exists $m^* \in \{1, ..., M\}$ such that $|\beta_{m^*} - \beta'_{m^*}| > \alpha_{m^*}$ or $\alpha_{m^*} > \beta_{m^*} + \beta'_{m^*}$ or $\alpha_{m^*} + \beta_{m^*} + \beta'_{m^*}$ is odd. By Corollary 6.1, it results in

$\left\langle P^{m^*}_{\alpha_{m^*}} P^{m^*}_{\beta_{m^*}}, P^{m^*}_{\beta'_{m^*}} \right\rangle_{\rho_{m^*}} = 0$ and then

$$\begin{aligned}
\left\langle \psi_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\beta}}, \psi_{\boldsymbol{\beta}'} \right\rangle_{\rho} &= \prod_{m=1}^{M} \left\langle P^{m}_{\alpha_m} P^{m}_{\beta_m}, P^{m}_{\beta'_m} \right\rangle_{\rho_m} \\
&= \left\langle P^{m^*}_{\alpha_{m^*}} P^{m^*}_{\beta_{m^*}}, P^{m^*}_{\beta'_{m^*}} \right\rangle_{\rho_{m^*}} \prod_{m=1, m\neq m^*}^{M} \left\langle P^{m}_{\alpha_m} P^{m}_{\beta_m}, P^{m}_{\beta'_m} \right\rangle_{\rho_m} \\
&= 0.
\end{aligned}$$

Thus, $[G_{\boldsymbol{\alpha}}]_{js} = 0$ for all $\boldsymbol{\alpha} \in \mathbb{I}_r$ and $[G_r]_{js} = 0$, which is a contradiction.

□

The sparsity pattern of the preconditioner P_r can be identified by the following corollary.

Corollary 6.5. *Let q be a bijection from $\{1, 2, \ldots, N_{\boldsymbol{y}}\}$ to $\mathbb{I}_k^M$. Let $j, s \in \{1, 2, \ldots, N_{\boldsymbol{y}}\}$ and $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_k^M$ such that $\boldsymbol{\beta} = q(j)$ and $\boldsymbol{\beta}' = q(s)$. If $[P_r]_{js} \neq \mathbf{0}$, then either condition (6.8) or condition (6.9) holds.*

Recall that the multi-indices in $\mathbb{I}_k^M$ represent the nodes of the parametric mesh associated with $\mathbb{I}_k^M$. There is an edge linking the nodes $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_k^M$ if they satisfy condition (6.9). It is easy to see that the number of edges in the parametric mesh will increase with the size of the set $\mathbb{I}_r$.

Note that, for convenience, we ignore self-loops in the parametric mesh.

Example 6.1. Let $r = 3$ and $\mathbb{I}_r = \{100, 200, 010\} \subseteq \mathbb{I}_{2k}^M$. The multi-indices in the set $\mathbb{I}_2^3$ represent the nodes of the parametric mesh associated with $\mathbb{I}_k^M$. To construct the parametric mesh, we find the edges in the mesh for each multi-index $\boldsymbol{\alpha} \in \mathbb{I}_r$ before we combine all the edges to get the parametric mesh.

The edges for each $\boldsymbol{\alpha} \in \mathbb{I}_r$ are shown in Table 6.1.

Thus, we combine all the edges for each multi-indices in $\mathbb{I}_r$ but do not include self-loops and obtain the parametric mesh associated with $\mathbb{I}_k^M$ as shown in Figure 6.1. Note that we label the edges to show which multi-index or multi-indices in $\mathbb{I}_r$ induce that edge.

$\boldsymbol{\alpha}$	Edges			
$(1,0,0)$	$(000,100)$	$(100,200)$	$(010,110)$	$(001,101)$
$(2,0,0)$		$(000,200)$	$(200,200)$	
$(0,1,0)$	$(000,010)$	$(010,020)$	$(100,110)$	$(001,001)$

Table 6.1: List of edges for each $\boldsymbol{\alpha} \in \mathbb{I}_r$.

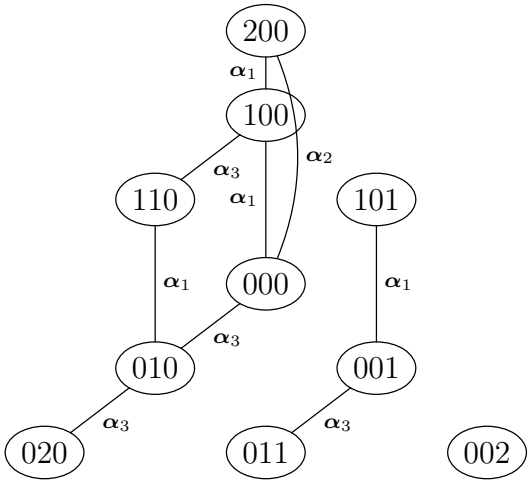


Figure 6.1: The parametric mesh associated with $\mathbb{I}_2^3$ and $\mathbb{I}_r = \{(1,0,0),(2,0,0),(0,1,0)\}$.

6.4.2 Even-odd Partition for Non-affine Parametric Diffusion Coefficients

Recall that domain decomposition techniques aim to partition the matrix P_r as a 2-by-2 block matrix, i.e.,

$$P_r = \begin{bmatrix} P_{II} & P_{IF} \\ P_{FI} & P_{FF} \end{bmatrix}.$$

where the block matrix P_{II} is required to be non-singular. Since P_{II} is induced by submeshes, then we have to ensure that, after the partitioning, each block matrix corresponding to a submesh is invertible. If we know that

$$a_{\mathbf{0}} + \sum_{\boldsymbol{\alpha} \in \mathbb{I}_r} a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{y}) > 0, \qquad \text{for all } (\boldsymbol{x},\boldsymbol{y}) \in D \times \boldsymbol{\Gamma}$$

or the preconditioner P_r is symmetric and positive definite, then each submesh always yields a symmetric and positive definite block matrix which is invertible.

To obtain a 2-by-2 block structure for P_r, we permute the nodes in the parametric

mesh in such a way that P_{II} is invertible. This partition utilises the positivity of the block-diagonal matrix of P_r.

To ensure that each block on the main diagonal matrix of P_{II} is invertible, we partition the parametric mesh into many submeshes which contain only a single node. Since the structure of the parametric mesh for non-affine parametric diffusion coefficients depends on the set $\mathbb{I}_r$, it may not be a bipartite graph. Consequently, we cannot obtain the unit submeshes by even-odd partitioning (partitioning of the set of multi-indices $\mathbb{I}_k^M$ by even nodes or odd nodes) and we need an algorithm to partition the parametric mesh for the case of non-affine parametric coefficients.

Algorithm 5 Algorithm to partition the parametric mesh.

$\mathbb{I}_{\mathcal{P}} = \emptyset$, $\mathbb{I}_{\mathcal{I}} = \emptyset$
while $\mathbb{I}_{\mathcal{P}} \cup \mathbb{I}_{\mathcal{I}} \neq \mathbb{I}_k^M$
    add a node $\boldsymbol{\alpha} \in \mathbb{I}_k^M \setminus (\mathbb{I}_{\mathcal{P}} \cup \mathbb{I}_{\mathcal{I}})$ which has the lowest degree to the set $\mathbb{I}_{\mathcal{P}}$
    add all the adjacent nodes of $\boldsymbol{\alpha}$ to the set $\mathbb{I}_{\mathcal{I}}$
end while

Algorithm 5 partitions the parametric mesh into many submeshes with one node and tries to maximise the number of submeshes. The algorithm starts by initialising the set $\mathbb{I}_{\mathcal{P}}$ and $\mathbb{I}_{\mathcal{I}}$ to be $\emptyset$. It will choose an available node in the set $\mathbb{I}_k^M \setminus (\mathbb{I}_{\mathcal{P}} \cup \mathbb{I}_{\mathcal{I}})$ with the lowest degree to be a new submesh and then mark all the adjacent nodes to be the nodes on the interface. This process will be repeated until all the nodes in $\mathbb{I}_k^M$ are assigned to be a submesh or interface. Note that Algorithm 5 does not guarantee that $[P_r]_{FF}$ will be a block-diagonal matrix because of the structure of the parametric mesh.

Example 6.2. From Example 6.1, recall that $\mathbb{I}_r = \{(1,0,0),(2,0,0),(0,1,0)\}$. Apply Algorithm 5 to the parametric mesh associated with $\mathbb{I}_k^M$ and obtain the following partition as shown in Figure 6.2. Therefore, we have

$$\mathbb{I}_{\mathcal{P}} = \{110, 000, 101, 020, , 011, 002\} \text{ and } \mathbb{I}_{\mathcal{I}} = \{010, 001, 100, 200\}.$$

We define the map q from $\{1, ..., N_{\boldsymbol{y}}\}$ to $\mathbb{I}_k^M$ corresponding to the partition to obtain a 2-by-2 block matrix of the preconditioner P_r. After the truncation preconditioner P_r is

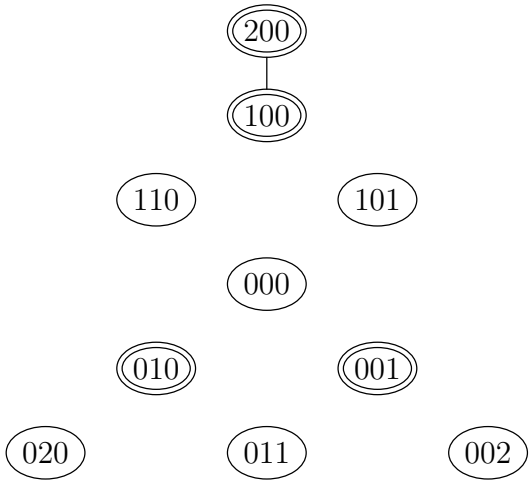


Figure 6.2: The partition of the parametric mesh associated with $\mathbb{I}_k^M$ and $\mathbb{I}_r = \{(1,0,0),(2,0,0),(0,1,0)\}$ by Algorithm 5.

permuted to 2-by-2 block form, we factorise the preconditioner P_r, i.e.,

$$P_r = \begin{bmatrix} I & \\ P_{FI}P_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} P_{II} & \\ & S_P \end{bmatrix} \begin{bmatrix} I & P_{II}^{-1}P_{IF} \\ & I \end{bmatrix},$$

where $S_P = P_{FF} - P_{FI}P_{II}^{-1}P_{IF}$ denotes the Schur complement of P_{II} in P_r.

Suppose P_{II} and S_P are approximated by symmetric and positive definite matrices $\tilde{P}_{II}$ and $\tilde{S}_P$, respectively. We can define a preconditioner by

$$\begin{bmatrix} I & \\ P_{FI}\tilde{P}_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} \tilde{P}_{II} & \\ & \tilde{S}_P \end{bmatrix} \begin{bmatrix} I & \tilde{P}_{II}^{-1}P_{IF} \\ & I \end{bmatrix}. \tag{6.10}$$

Remark. This procedure can be viewed as an approximation technique for the truncation preconditioners P_r for affine parametric diffusion coefficients. Recall that the stochastic Galerkin matrix for affine parametric diffusion coefficients can be written as

$$A = \sum_{\boldsymbol{\alpha}\in\mathbb{I}_1^M} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}.$$

Selecting $\mathbb{I}_r = \left\{\boldsymbol{\alpha} \in \mathbb{I}_1^M \mid \sum_{m=1}^r \alpha_m = 1\right\}$ leads to the truncation preconditioner P_r which

is symmetric and positive definite. Thus, the truncation preconditioner P_r for affine parametric diffusion coefficients can be represented by the block matrix in (6.10).

Recall that the block matrices along the main diagonal of P_{FF} are symmetric and positive definite. Therefore, the matrix $\tilde{P}_{FF}$, which denotes the symmetric block Gauss-Seidel approximation of P_{FF}, is symmetric and positive definite. So we approximate S_P by $\tilde{P}_{FF}$. As a result, we can define the preconditioner with domain decomposition technique by

$$\hat{P}_r = \begin{bmatrix} I & \\ P_{FI}P_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} P_{II} & \\ & \tilde{P}_{FF} \end{bmatrix} \begin{bmatrix} I & P_{II}^{-1}P_{IF} \\ & I \end{bmatrix}.$$

Note that $\hat{P}_r$ is symmetric and positive definite.

6.4.3 Computational Cost

Let $\mathbf{r} \in \mathbb{R}^{N_{\boldsymbol{x}}N_{\boldsymbol{y}}}$ such that

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_F \end{bmatrix},$$

where $\mathbf{r}_I \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{P}}}$ and $\mathbf{r}_F \in \mathbb{R}^{N_{\boldsymbol{y}_\mathcal{I}}}$. Consider now the linear system

$$\hat{P}_r\mathbf{z} = \mathbf{r}.$$

We apply Algorithm 3 to solve the linear system. In Algorithm 3, it is required to solve the linear system with $\tilde{P}_{FF}$ which does not facilitate parallelism.

Let D be the block-diagonal matrix of P_{FF} and L be the strictly lower block-triangular matrix of P_{FF}. The solution of the linear system $\hat{P}_r\mathbf{z}_F = \mathbf{r}_F$ can be obtained as follows

1. Solve the linear system $(D + L)\mathbf{z}_1 = \mathbf{r}_F$.

2. Solve the linear system $(D + L^T)\mathbf{z}_F = D\mathbf{z}_1$.

Thus, the complexity for solving the linear system with $\hat{P}_{FF}$ is

$$\mathcal{F}\ell\left(\tilde{P}_{FF}^{-1}\mathbf{r}_F\right) \approx 2N_{\boldsymbol{y}_\mathcal{I}}\mathcal{F}\ell\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + \mathrm{nnz}(P_{FF}).$$

Combine the complexity of $\tilde{P}_{FF}^{-1}\mathbf{r}_F$ with the complexity from Algorithm 3. As a result, the sequential complexity of $\hat{P}_r$ is

$$\mathcal{F}\ell\left(\hat{P}_r^{-1}\mathbf{r}\right) \approx 2N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + \frac{2kr}{M+k}N_{\boldsymbol{y}}\mathrm{nnz}(K_{\mathbf{0}}),$$

and parallel complexity of $\hat{P}_r$ is

$$\mathcal{F}\ell p\left(\hat{P}_r^{-1}\mathbf{r}\right) \approx 2\left(1+N_{\boldsymbol{y}_\mathcal{I}}\right)\mathcal{F}\ell p\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + 2\mathrm{nnz}(K_{\mathbf{0}}).$$

6.5 Special Case: log-transformed Diffusion Coefficients

Suppose the diffusion coefficient a is a log-transformed coefficient, i.e.,

$$a(\boldsymbol{x},\boldsymbol{y}) = \exp\left(b_0(\boldsymbol{x}) + \sum_{m=1}^{N} b_m(\boldsymbol{x})y_m\right), \tag{6.11}$$

where $b_m \in L^\infty(D)$ for all $m = 0, 1, ..., N$. Then, we represent a by using the gPC expansion as follows

$$a(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{\alpha}\in\mathbb{I}} a_{\boldsymbol{\alpha}}(\boldsymbol{x})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}),$$

where

$$a_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y})a(\boldsymbol{x},\boldsymbol{y})\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}. \tag{6.12}$$

In the case of log-transformed diffusion coefficients, we approximate the coefficient a by a truncation of a. The following result shows that a truncation of a preserves positivity.

Lemma 6.6. *Let a be a log-transformed coefficient in the form (6.11). Define $\mathbb{I}^1 =$*

$\{\boldsymbol{\alpha} \in \mathbb{I} \mid \alpha_m = 0 \text{ for all } m \geq 2\}$ *and a truncated expansion of* a *by* $a_{e_1} : D \times \boldsymbol{\Gamma} \to \mathbb{R}$ *by*

$$a_{e_1}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \mathbb{I}^1} a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{y}), \tag{6.13}$$

where $a_{\boldsymbol{\alpha}}$ *are defined in (6.12).*

Then, a_{e_1} *is positive on* $D \times \boldsymbol{\Gamma}$.

Proof. Consider $a_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathbb{I}^1$,

$$a_{\boldsymbol{\alpha}} = \exp(b_0(\boldsymbol{x})) a_{\alpha_1}(\boldsymbol{x}) \prod_{m=2}^{N} \mathbb{E}_m \left[\exp\left(b_m(\boldsymbol{x}) y_m\right)\right],$$

where

$$a_{\alpha_m} = \int_{\Gamma_m} \rho(y_m) \exp\left(b_m(\boldsymbol{x}) y_m\right) P^m_{\alpha_m}(y_m) \mathrm{d}y_m$$

and

$$\mathbb{E}_m \left[F(y_m)\right] = \int_{\Gamma_m} \rho(y_m) F(y_m) \mathrm{d}y_m$$

with $P^m_n(y_m)$ a generic orthonormal polynomial defined in (2.16).

Thus, we substitute back in a_{e_1} and obtain

$$a_{e_1}(\boldsymbol{x}, \boldsymbol{y}) = \exp(b_0(\boldsymbol{x})) \prod_{m=2}^{N} \mathbb{E}_m \left[\exp\left(b_m(\boldsymbol{x}) y_m\right)\right] \left(\sum_{n=0}^{\infty} a_n(\boldsymbol{x}) P^1_n(y_1)\right).$$

Using the gPC expansion, we get

$$\exp\left(b_1(\boldsymbol{x}) y_1\right) = \sum_{n=0}^{\infty} a_n(\boldsymbol{x}) P^1_n(y_1).$$

This leads to

$$a_{e_1}(\boldsymbol{x}, \boldsymbol{y}) = \exp(b_0(\boldsymbol{x}) + b_1(\boldsymbol{x}) y_1) \prod_{m=2}^{N} \mathbb{E}_m \left[\exp\left(b_m(\boldsymbol{x}) y_m\right)\right].$$

Since $\mathbb{E}_m \left[\exp\left(b_m(\boldsymbol{x}) y_m\right)\right]$ are positive for all $m = 2, ..., N$, $a_{e_1}(\boldsymbol{x}, \boldsymbol{y})$ is positive for all $(\boldsymbol{x}, \boldsymbol{y}) \in D \times \boldsymbol{\Gamma}$.

□

We define a bilinear form $B_{e_1} : V \times V \to \mathbb{R}$ via a_{e_1} by

$$B_{e_1}(u,v) = \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \int_D a_{e_1}(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \tag{6.14}$$

We use the bilinear form B_{e_1} to define a preconditioner P_{e_1}

$$P_{e_1} \coloneqq \sum_{\boldsymbol{\alpha} \in \mathbb{I}^1_{2k}} G_{\boldsymbol{\alpha}} \otimes K_{\boldsymbol{\alpha}}. \tag{6.15}$$

where $\mathbb{I}^1_{2k} = \{\boldsymbol{\alpha} \in \mathbb{I}^1 \mid |\boldsymbol{\alpha}| \leq 2k\}$.

Lemma 6.7. *Let P_{e_1} be defined in (6.15). Then, P_{e_1} is symmetric and positive definite.*

Proof. It is clear that the bilinear B_{e_1} is symmetric, therefore so is P_{e_1}.

Let $\mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{x}} N_{\boldsymbol{y}}} \setminus \{\mathbf{0}\}$ and let $v \in S_k^M$ correspond to $\mathbf{v}$. Consider

$$\mathbf{v}^T P_{e_1} \mathbf{v} = B_{e_1}(v,v).$$

Since $a_{e_1}(\boldsymbol{x},\boldsymbol{y}) > 0$ for all $\boldsymbol{x} \in D$ and $\boldsymbol{y} \in \boldsymbol{\Gamma}$, then $B_{e_1}(v,v) > 0$ for all $v \in S_k^M$. Thus, P_{e_1} is symmetric and positive definite.

□

6.5.1 Parametric Mesh

Since all multi-indices $\boldsymbol{\alpha} \in \mathbb{I}^1_{2k}$ have $\alpha_m = 0$ for $m = 2, ..., M$, we partition the set of multi-indices $\mathbb{I}^M_k$ into many subsets, namely $\mathbb{I}_1$,..., $\mathbb{I}_N$, by grouping the multi-indices with the same second index to the last index together. That is, for $l = 1, ..., N$ and for any $\boldsymbol{\beta}$, $\tilde{\boldsymbol{\beta}} \in \mathbb{I}_l$,

$$\beta_m = \tilde{\beta}_m, \qquad \text{for all } m \geq 2,$$

and for any $\mathbb{I}_{l_1} \neq \mathbb{I}_{l_2}$ and for each $\boldsymbol{\beta} \in \mathbb{I}_{l_1}$, $\boldsymbol{\beta}' \in \mathbb{I}_{l_2}$, there exists $m^* \geq 2$ such that

$$\beta_{m^*} \neq \beta'_{m^*}.$$

Additionally, since the multi-indices in the sets $\mathbb{I}_l$, $l = 1, ..., N$, are grouped by second index to the last index of the multi-indices in $\mathbb{I}_k^M$, then

$$N = \begin{pmatrix} M+k-1 \\ M-1 \end{pmatrix}.$$

Furthermore, the number of multi-indices in set $\mathbb{I}_l$ is between 1 to $k+1$, inclusive. For instance, $\mathbb{I}_l = \{(0, k, 0, ..., 0)\}$ or $\mathbb{I}_l = \left\{\boldsymbol{\alpha} \in \mathbb{I}_k^M \mid \alpha_m = 0 \text{ for all } m \geq 2\right\}$. Moreover, the number of the set $\mathbb{I}_l$ with j members is

$$n_j = \begin{pmatrix} k+M-j-1 \\ M-2 \end{pmatrix}.$$

In practice, we need to balance the size of each submesh to avoid a bottleneck. We may combine some small submeshes and assign to one processor.

Next, to study the property of the induced subgraphs whose set of nodes is $\mathbb{I}_l$, the following definition is required.

Definition 6.8. A graph is called a complete graph if there exists an edge linking each pair of distinct nodes.

Lemma 6.9. *Let $\mathcal{M}$ be a parametric mesh associated with $\mathbb{I}_k^M$ induced by the set $\mathbb{I}_{2k}^1$. For $l = 1, ..., N$, let $G_l = (\mathbb{I}_l, E_l)$ be an induced subgraph of $\mathcal{M}$. Then, G_l is a complete graph.*

Proof. Let $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{I}_l$ such that $\boldsymbol{\beta} \neq \boldsymbol{\beta}'$. Choose $\boldsymbol{\alpha} = (\beta_1 + \beta'_1, 0, ...0) \in \mathbb{I}_{2k}^1$. We can see that

$$|\beta_1 - \beta'_1| \leq \alpha_1 \leq \beta_1 + \beta'_1 \text{ and } \alpha_1 + \beta_1 + \beta'_1 \text{ is even}$$

and

$$0 = |\beta_m - \beta'_m| \leq \alpha_m \leq \beta_m + \beta'_m \text{ and } \alpha_m + \beta_m + \beta'_m \text{ is even}, \qquad \text{for all } m \geq 2.$$

Thus, $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ satisfy condition (6.9) and then there exists an edge linking $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$.

□

The following lemma shows that the parametric mesh induced by the set $\mathbb{I}^1_{2k}$ can be partitioned into many submeshes corresponding to the sets $\mathbb{I}_1$,..., $\mathbb{I}_N$.

Lemma 6.10. *Let $\mathcal{M}$ be a parametric mesh associated with $\mathbb{I}^M_k$ induced by the set of multi-indices $\mathbb{I}^1_{2k}$. Let $G = (\mathbb{I}_{l_1} \cup \mathbb{I}_{l_2}, E)$ be an induced subgraph in $\mathcal{M}$ where $\mathbb{I}_{l_1}$ and $\mathbb{I}_{l_2}$ are disjoint. Then, G is disconnected.*

Proof. Let $\boldsymbol{\beta} \in \mathbb{I}_{l_1}$ and $\boldsymbol{\beta}' \in \mathbb{I}_{l_2}$. By the definition of the set $\mathbb{I}_l$, for $l \in \{1, ..., N\}$, there exists $m^* \geq 2$ such that $\beta_{m^*} \neq \beta'_{m^*}$. Thus, $0 < |\beta_{m^*} - \beta'_{m^*}|$ and then $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ do not satisfy condition (6.9). Therefore, there is no edge linking the multi-indices $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$. Consequently, no node in $\mathbb{I}_{l_1}$ is linked with a node in $\mathbb{I}_{l_2}$. Hence, there is no path to connect a node in $\mathbb{I}_{l_1}$ to a node in $\mathbb{I}_{l_2}$.

□

As a result of Lemma 6.10, the parametric mesh induced by the set $\mathbb{I}^1_{2k}$ can be partitioned into many non-overlapping submeshes, namely $\mathcal{P}_1$, $\mathcal{P}_2$, ..., $\mathcal{P}_N$, and the sets $\mathbb{I}_l$, for $l = 1, ..., N$, are the set of nodes of each submesh. Note that there is no node on the interface, i.e., $\mathbb{I}_{\mathcal{I}} = \emptyset$. In addition, each submesh is a complete graph.

If the parametric mesh is a complete graph, then it means that each pair of multi-indices in $\mathbb{I}^M_k$ satisfy condition (6.9). So, the preconditioner P is a block dense matrix.

Example 6.3. For $M = 3$, $k = 2$, let $\mathcal{M}$ be the parametric mesh induced by the set $\mathbb{I}^1_{2k}$ where

$$\mathbb{I}^1_{2k} = \{(0,0,0), (1,0,0), (2,0,0), (3,0,0), (4,0,0)\}.$$

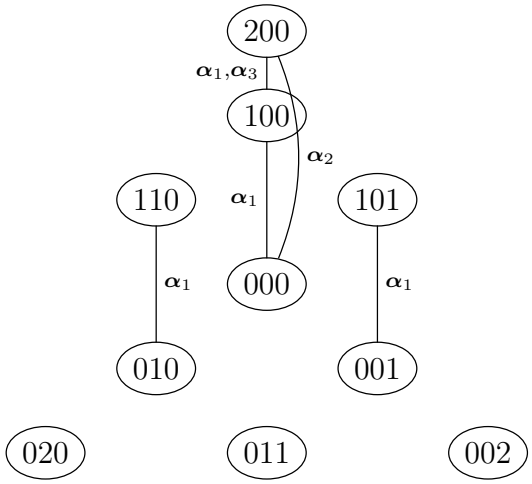


Figure 6.3: The parametric mesh associated with $\mathbb{I}_2^3$ induced by $\mathbb{I}_{2k}^1 = \{\boldsymbol{\alpha}_i \coloneqq (i,0,0) \mid i = 1,2,3,4\}$.

By Lemma 6.10, the set of multi-indices $\mathbb{I}_k^M$ is partitioned into

$$\mathbb{I}_1 = \{000, 100, 200\} \qquad \mathbb{I}_2 = \{010, 110\} \qquad \mathbb{I}_3 = \{001, 101\}$$

$$\mathbb{I}_4 = \{020\} \qquad \mathbb{I}_5 = \{011\} \qquad \mathbb{I}_6 = \{002\}$$

The edges for each $\boldsymbol{\alpha} \in \mathbb{I}_{2k}^1$ are shown in Table 6.2.

$\boldsymbol{\alpha}$	Edges			
$(1,0,0)$	$(000,100)$	$(100,200)$	$(010,110)$	$(001,101)$
$(2,0,0)$		$(000,200)$	$(200,200)$	
$(3,0,0)$		$(100,200)$		
$(4,0,0)$		$(200,200)$		

Table 6.2: List of edges for each $\boldsymbol{\alpha} \in \mathbb{I}_{2k}^1$.

Again, we combine all the edges in Table 6.2 but do not include self-loops and get the parametric mesh as shown in Figure 6.3. We can see that there is no node on the interface and each submesh is a complete graph.

6.5.2 Computational Cost

In the previous subsection, we have seen that partitioning the parametric mesh induced by the multi-indices set $\mathbb{I}_{2k}^1$ gives many complete submeshes with no interface. As a result, the preconditioner for log-transformed diffusion coefficients can be presented by block diagonal structure where each block is block dense as shown in Figure 6.4. The largest block in the preconditioner P_{e_1} has block size $(k+1)$-by-$(k+1)$ and the smallest blocks are single block matrices.

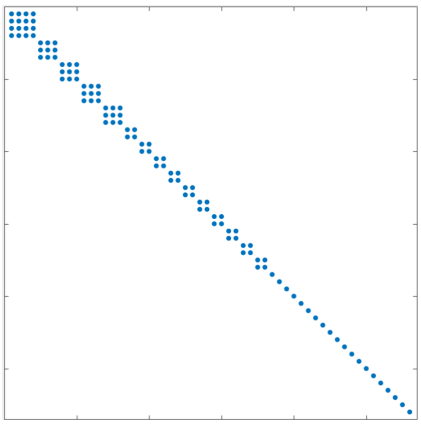


Figure 6.4: Block pattern of a preconditioner for log-transformed diffusion coefficient with $M = 5$ and $k = 3$.

In the following, let $n = 1, ..., N_{\mathcal{P}}$ and $S^{(n)}$ be a vector subspace of S_k^M associated with the set $\mathbb{I}_n$, i.e.,

$$S^{(n)} = \text{span}\left\{\psi_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{I}_n\right\}.$$

Let $V^{(n)} := S^{(n)} \otimes X_h$. Since $V^{(n)}$ is a subspace of V_{hk}^M and $B_{e_1}(v, v)$ is positive for all $v \in V_{hk}^M$ and $v \neq 0$, each block in the preconditioner P_{e_1} is symmetric and positive definite.

If the discretisation parameter k is not too high, we may assign one block matrix to one processor which requires $\sum_{j=1}^{k+1} n_j = \frac{k}{k+M} N_{\boldsymbol{y}}$ processors or less and solve by a direct solver.

On the other hand, we may apply symmetric block Gauss-Seidel approximation to each block in the preconditioner P_{e_1}. However, we choose to approximate the preconditioner P_{e_1}

by an inner-outer iteration technique. This technique is useful if the preconditioner has a block diagonal structure, and the cost of solving the linear system with the preconditioner is high. For each block in P_{e_1}, we apply one PCG iteration with its symmetric block Gauss-Seidel as a preconditioner and we denote the preconditioner P_{e_1} with one PCG inner iteration by $\tilde{P}_{e_1}$.

As a result, the sequential complexity of $\tilde{P}_{e_1}$ is

$$\begin{aligned}\mathcal{F}\ell\left(\tilde{P}_{e_1}^{-1}\mathbf{r}\right) &\approx \mathcal{F}\ell\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right)\left(2\sum_{j=2}^{k+1} jn_j + n_1\right) + \mathrm{nnz}(K_{\mathbf{0}})\sum_{j=2}^{k+1} j^2 n_j, \\ &\leq 2N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + \left(1+\frac{2k}{M+1}\right)N_{\boldsymbol{y}}\mathrm{nnz}(K_{\mathbf{0}}),\end{aligned}$$

and parallel complexity of $\tilde{P}_{e_1}$ depends on the largest block which has the size of $k+1$. Thus,

$$\mathcal{F}\ell p\left(\tilde{P}_{e_1}^{-1}\mathbf{r}\right) \approx 2\left(1+k\right)\mathcal{F}\ell p\left(K_{\mathbf{0}}^{-1}\mathbf{v}\right) + 2\mathrm{nnz}(K_{\mathbf{0}}).$$

Remark. Consider

$$\sum_{j=1}^{k+1} j^2 n_j = \sum_{i=M-2}^{k+M-2} (k+M-1-i)^2 \begin{pmatrix} i \\ M-2 \end{pmatrix}.$$

Let $N = k+M-2$ and $r = M-2$. Then,

$$\begin{aligned}\sum_{j=1}^{k+1} j^2 n_j &= \sum_{i=r}^{N} (N+1-i)^2 \begin{pmatrix} i \\ r \end{pmatrix} \\ &= (N+1)^2 \sum_{i=r}^{N} \begin{pmatrix} i \\ r \end{pmatrix} - 2(N+1)\sum_{i=r}^{N} i \begin{pmatrix} i \\ r \end{pmatrix} + \sum_{i=r}^{N} i^2 \begin{pmatrix} i \\ r \end{pmatrix}.\end{aligned}$$

Recall that, by the hockey stick identity,

$$\sum_{i=r}^{N} \begin{pmatrix} i \\ r \end{pmatrix} = \begin{pmatrix} N+1 \\ r+1 \end{pmatrix}.$$

Moreover, by mathematical induction, we can prove the generalised extended hockey stick identities [75]

$$\sum_{i=r}^{N} i \begin{pmatrix} i \\ r \end{pmatrix} = \left(\frac{Nr+N+r}{r+2}\right) \begin{pmatrix} N+1 \\ r+1 \end{pmatrix},$$
$$\sum_{i=r}^{N} i^2 \begin{pmatrix} i \\ r \end{pmatrix} = \left(\frac{N^2(r+1)(r+2)+N(2r+1)(r+1)+r(r-1)}{(r+2)(r+3)}\right) \begin{pmatrix} N+1 \\ r+1 \end{pmatrix}.$$

Thus,

$$\sum_{j=1}^{k+1} j^2 n_j = \left(\frac{2N-r+3}{r+3}\right) \begin{pmatrix} N+2 \\ r+2 \end{pmatrix} = \left(1+\frac{2k}{M+1}\right) N_{\boldsymbol{y}}.$$

6.5.3 Spectral Analysis

We analyse the bilinear equivalence between bilinear forms B_{e_1} and B in the case of a bounded parametric domain, such as loguniform coefficient, as follows.

Theorem 6.11. *Assume that $\Gamma_m = [-1,1]$ for all $m = 1,...,N$. Let a be the log-transformed coefficient in (6.11). Let a_{e_1} be the function in (6.13) and B_{e_1} be the bilinear form in (6.14). Then, the bilinear form B_{e_1} is equivalent to the bilinear form B in (2.12). To be precise, there exist positive numbers θ and Θ independent of discretisation parameters, i.e., h, k and M, such that*

$$\theta B_{e_1}(v,v) \leq B(v,v) \leq \frac{1}{\theta} B_{e_1}(v,v), \qquad \text{for all } v \in V$$

where

$$\theta = \exp\left(-2\sum_{m=2}^{N} \|b_m\|_{L^\infty(D)}\right). \tag{6.16}$$

Proof. It is easy to see that for $m = 1,...,N$,

$$\exp\left(-\|b_m\|_{L^\infty(D)}\right) \leq \exp\left(b_m(\boldsymbol{x})y_m\right) \leq \exp\left(\|b_m\|_{L^\infty(D)}\right)$$

and then

$$\exp\left(-\|b_m\|_{L^\infty(D)}\right) \leq \mathbb{E}_m\left[\exp\left(b_m(\boldsymbol{x})y_m\right)\right] \leq \exp\left(\|b_m\|_{L^\infty(D)}\right).$$

Therefore, for any $v \in V$,

$$\begin{aligned} B(v,v) &= \int_{\boldsymbol{\Gamma}}\int_D \rho(\boldsymbol{y})\left(\prod_{m=2}^{N}\frac{\exp\left(b_m(\boldsymbol{x})y_m\right)}{\mathbb{E}_m\left[\exp\left(b_m(\boldsymbol{x})y_m\right)\right]}\right) a_{e_1}(\boldsymbol{x},\boldsymbol{y})\left(\nabla v\right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &\leq \left(\max_{\boldsymbol{x}\in D}\prod_{m=2}^{N}\frac{\exp\left(\|b_m\|_{L^\infty(D)}\right)}{\mathbb{E}_m\left[\exp\left(b_m(\boldsymbol{x})y_m\right)\right]}\right) B_{e_1}(v,v) \\ &\leq \exp\left(2\sum_{m=2}^{N}\|b_m\|_{L^\infty(D)}\right) B_{e_1}(v,v). \end{aligned}$$

The lower bound can be achieved in the same manner.

□

Remark. For the case of an unbounded parametric domain such as lognormal coefficient, we cannot prove the bilinear equivalence or the bilinear forms B_{e_1} and B are not equivalent.

This leads to the optimality of the preconditioner P_{e_1} in the case of a bounded parametric domain.

Corollary 6.12. *Assume that $\Gamma_m = [-1,1]$ for all $m = 1,...,N$. Let a be the log-transformed coefficient in (6.11) and a_{e_1} be the function in (6.13). Let P_{e_1} and A be induced by the bilinear form B_{e_1} in (6.14) and B in (2.12), respectively. Then*

$$\Lambda\left(P_{e_1}^{-1}A\right) \subseteq \left[\theta, \frac{1}{\theta}\right],$$

where θ defined in (6.16) is independent of discretisation parameters h, k and M.

6.6 Numerical Experiments

In this section, we compare the performance of modified truncation preconditioners $\tilde{P}_r$, truncation preconditioners with domain decomposition technique $\hat{P}_r$ and preconditioners for log-transformed diffusion coefficients $\tilde{P}_{e_1}$ with the performance of existing preconditioners such as the mean-based preconditioner and the Kronecker product preconditioner. We consider test problems with the lognormal and loguniform diffusion coefficients in the form

$$a(\boldsymbol{x}, \boldsymbol{y}) = \exp(b_0(\boldsymbol{x}) + \sum_{m=1}^{N} b_m(\boldsymbol{x}) y_m). \tag{6.17}$$

We employ a gPC expansion for the diffusion coefficients a, i.e., the coefficient a can be written as

$$a(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in \mathbb{I}} a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{y}).$$

Recall that the lognormal diffusion coefficient a does not satisfy condition (2.10) but the solution of the variational formulation (2.11) exists. On the other hand, if we let $\Gamma_m = [-1, 1]$ for all $m = 1, ..., N$, the loguniform diffusion coefficient satisfies condition (2.10) with

$$a_{\min} = \exp\left(- \|b_0\|_{L^\infty(D)} - \sum_{m=1}^{N} \|b_m\|_{L^\infty(D)} \right),$$
$$a_{\max} = \exp\left(\|b_0\|_{L^\infty(D)} + \sum_{m=1}^{N} \|b_m\|_{L^\infty(D)} \right).$$

The coefficients $a_{\boldsymbol{\alpha}}$ in the case of a lognormal distribution are calculated in [125, p. 926],

$$a_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \mathbb{E}\left[a(\boldsymbol{x}, \boldsymbol{y})\right] \prod_{m=1}^{N} \frac{a_m^{\alpha_m}(\boldsymbol{x})}{\sqrt{\alpha_m!}}, \qquad \text{for } \boldsymbol{\alpha} \in \mathbb{I}$$

and the mean of the lognormal coefficient a is

$$\mathbb{E}\left[a(\boldsymbol{x}, \boldsymbol{y})\right] = a_{\boldsymbol{0}}(\boldsymbol{x}) = \exp\left(b_0(\boldsymbol{x}) + \frac{1}{2} \sum_{m=1}^{N} b_m^2(\boldsymbol{x}) \right).$$

In the case of loguniform coefficients, for $\boldsymbol{\alpha} \in \mathbb{I}$,

$$\begin{aligned} a_{\boldsymbol{\alpha}}(\boldsymbol{x}) &= \int_{\boldsymbol{\Gamma}} \rho(\boldsymbol{y}) \exp(b_0(\boldsymbol{x}) + \sum_{m=1}^{N} b_m(\boldsymbol{x}) y_m) \psi_{\boldsymbol{\alpha}}(\boldsymbol{y}) \mathrm{d}\boldsymbol{y} \\ &= \exp(b_0(\boldsymbol{x})) \prod_{m=1}^{N} \int_{\Gamma_m} \rho_m(y_m) \exp(b_m(\boldsymbol{x}) y_m) P_{\alpha_m}(y_m) \mathrm{d}y_m \\ &= \exp(b_0(\boldsymbol{x})) \prod_{m=1}^{N} a_{\alpha_m}(\boldsymbol{x}). \end{aligned}$$

By the properties of Legendre polynomials in [33, p 215], we have that, for $\boldsymbol{x} \in D$ such that $b_m(\boldsymbol{x}) = 0$,

$$a_{\alpha_m}(\boldsymbol{x}) = \begin{cases} 1, & \alpha_m = 0, \\ 0, & \alpha_m \neq 0, \end{cases}$$

and, otherwise,

$$a_{\alpha_m}(\boldsymbol{x}) = \sum_{i=0}^{\alpha_m} \frac{\sqrt{2\alpha_m + 1}}{(-2b_m(\boldsymbol{x}))^{i+1}} \frac{(\alpha_m + i)!}{i!(\alpha_m - i)!} \left((-1)^{\alpha_m + i} \exp(-b_m(\boldsymbol{x})) - \exp(b_m(\boldsymbol{x})) \right).$$

As a result, the mean of a loguniform coefficient is

$$\mathbb{E}\left[a(\boldsymbol{x}, \boldsymbol{y})\right] = a_{\mathbf{0}}(\boldsymbol{x}) = \exp\left(b_0(\boldsymbol{x})\right) \prod_{m=1}^{N} a_{0,m}(\boldsymbol{x}),$$

where

$$a_{0,m}(\boldsymbol{x}) = \begin{cases} 1, & b_m(\boldsymbol{x}) = 0, \\ \frac{\sinh\left(b_m(\boldsymbol{x})\right)}{b_m(\boldsymbol{x})}, & b_m(\boldsymbol{x}) \neq 0. \end{cases}$$

Next, we review the complexity of each preconditioner for non-affine diffusion coefficients. The mean-based preconditioner P_0 is still the mean term of the coefficient matrix A, i.e.,

$$P_0 = I_{N_{\boldsymbol{y}}} \otimes K_{\mathbf{0}}.$$

Thus, by the structure of P_0, the computational cost of P_0 for non-affine diffusion coeffi-

cients is identical to the case of affine diffusion coefficients and the computational costs of $\tilde{P}_r$, $\hat{P}_r$ and $\tilde{P}_{e_1}$ are higher than the cost of P_0 by a factor of 2. Additionally, the Kronecker product preconditioner $P_\otimes$ is defined by

$$P_\otimes = G \otimes K_{\mathbf{0}}.$$

Note that in the case of non-affine diffusion coefficients, the matrix G is dense. As a result, the computational cost of $P_\otimes$ is

$$\begin{aligned}\mathcal{F}\ell\left(P_\otimes^{-1}\mathbf{r}\right) &\approx N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + N_{\boldsymbol{x}}\mathcal{F}\ell\left(G^{-1}\mathbf{v}\right), \\ &= N_{\boldsymbol{y}}\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right) + N_{\boldsymbol{x}}N_{\boldsymbol{y}}^2,\end{aligned}$$

if the Cholesky factorisation of G is provided.

To conclude, we have that

$$\mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right) < \mathcal{F}\ell\left(\tilde{P}_r^{-1}\mathbf{r}\right) \approx \mathcal{F}\ell\left(\hat{P}_r^{-1}\mathbf{r}\right) \approx \mathcal{F}\ell\left(\tilde{P}_{e_1}^{-1}\mathbf{r}\right),$$

and

$$\mathcal{F}\ell\left(P_\otimes^{-1}\mathbf{r}\right) \approx \mathcal{F}\ell\left(P_0^{-1}\mathbf{r}\right) + N_{\boldsymbol{x}}N_{\boldsymbol{y}}^2,$$

by assuming that $\mathcal{F}\ell\left(K_0^{-1}\mathbf{v}\right)$ dominates the cost of one PCG iteration. Note that the preconditioners $P_\otimes$, $\tilde{P}_r$ and $\hat{P}_r$ require a one time setup before we apply PCG. For the Kronecker product preconditioner, it is required to construct the matrix G and apply the Cholesky factorisation to the matrix G. In the case of $\tilde{P}_r$ and $\hat{P}_r$, we need ordering the terms in a to obtain the set $\mathbb{I}_r$.

Example 6.4. In this experiment, the diffusion coefficient a is assumed to be a log-transformed coefficient in (6.17) with $N = 20$ where b_m are chosen to be the coefficients in Example 4.1 with $\tilde{\sigma} = 2$ and $\bar{\alpha} = 0.547$.

Before comparing the performance of our preconditioners, $\tilde{P}_r$ and $\hat{P}_r$ with $r = 1, ..., 6$

and $\tilde{P}_{e_1}$, with the mean-based preconditioner and the Kronecker product preconditioner, the first eight largest magnitudes of $a_{\boldsymbol{\alpha}}$ and corresponding multi-indices in $\mathbb{I}_{2k}^M$ with $M = k = 6$ are illustrated in Table 6.3. We can see that the magnitudes of $a_{\boldsymbol{\alpha}}$ in both cases (lognormal and loguniform) drop fairly fast. Although there is no clear pattern of multi-indices in Table 6.3, we can see that the magnitude of $a_{\boldsymbol{\alpha}}$ tends to be large with small $|\boldsymbol{\alpha}|$. Note that the multi-index $(3, 0, 0, 0, 0, 0)$ can not be a member in $\mathbb{I}_r$ with $k = 1$ because it is not in the set $\mathbb{I}_{2k}^M$.

In the following, we want to check the condition (6.4) for lognormal and loguniform diffusion coefficients. It is easy to see that $a_{\boldsymbol{\alpha}}(\boldsymbol{x})$, for $\boldsymbol{\alpha} \in \mathbb{I}_e$, is non-negative on D in the case of lognormal coefficients. Thus, condition (6.4) holds for lognormal coefficients.

In the case of loguniform coefficients, consider $a_{\alpha_m}(\boldsymbol{x})$. For $\boldsymbol{x} \in D$ such that $b_m(\boldsymbol{x}) = 0$, we have

$$a_{\alpha_m}(\boldsymbol{x}) = \begin{cases} 1, & \alpha_m = 0, \\ 0, & \text{otherwise.} \end{cases}$$

For $\boldsymbol{x} \in D$ such that $b_m(\boldsymbol{x}) \neq 0$,

case $\alpha_m = 0$: we get

$$a_{\alpha_m}(\boldsymbol{x}) = \frac{\exp(b_m(\boldsymbol{x})) - \exp(-b_m(\boldsymbol{x}))}{2b_m(\boldsymbol{x})} = \frac{\sinh\left(b_m(\boldsymbol{x})\right)}{b_m(\boldsymbol{x})}.$$

Since the hyperbolic sine is an odd function, $a_{\alpha_m}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in D$.

Lognormal		Loguniform	
$\boldsymbol{\alpha}$	$\|a_{\boldsymbol{\alpha}}\|_{L^\infty(D)}$	$\boldsymbol{\alpha}$	$\|a_{\boldsymbol{\alpha}}\|_{L^\infty(D)}$
$(0, 0, 0, 0, 0, 0)$	3.1960	$(0, 0, 0, 0, 0, 0)$	2.8675
$(1, 0, 0, 0, 0, 0)$	1.7482	$(1, 0, 0, 0, 0, 0)$	0.8880
$(2, 0, 0, 0, 0, 0)$	0.6762	$(0, 1, 0, 0, 0, 0)$	0.2261
$(0, 1, 0, 0, 0, 0)$	0.4371	$(2, 0, 0, 0, 0, 0)$	0.1244
$(1, 1, 0, 0, 0, 0)$	0.2391	$(0, 0, 1, 0, 0, 0)$	0.1006
$(3, 0, 0, 0, 0, 0)$	0.2135	$(1, 1, 0, 0, 0, 0)$	0.0700
$(0, 0, 1, 0, 0, 0)$	0.1942	$(0, 0, 0, 1, 0, 0)$	0.0566
$(0, 0, 0, 1, 0, 0)$	0.1093	$(0, 0, 0, 0, 1, 0)$	0.0362

Table 6.3: First 8 terms of lognormal and loguniform diffusion coefficients in Example 6.4.

		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_6$	P_{e_1}	$\tilde{P}_{e_1}$
$k=$	1	12	12	6	7	6	6	6	6	6	7	7	6	6	6	6	7
	2	18	19	8	10	9	9	8	8	13	10	10	10	9	9	8	9
	3	25	26	10	12	11	11	10	10	17	14	13	12	12	12	9	11
	4	32	34	13	15	13	13	12	11	27	17	17	16	14	14	10	12
	5	40	43	17	19	16	17	13	12	37	20	21	20	17	17	10	14
	6	49	52	24	22	19	20	14	14	53	25	23	24	19	19	11	15

Table 6.4: The numbers of iteration by PCG with lognormal diffusion coefficient in Example 6.4.

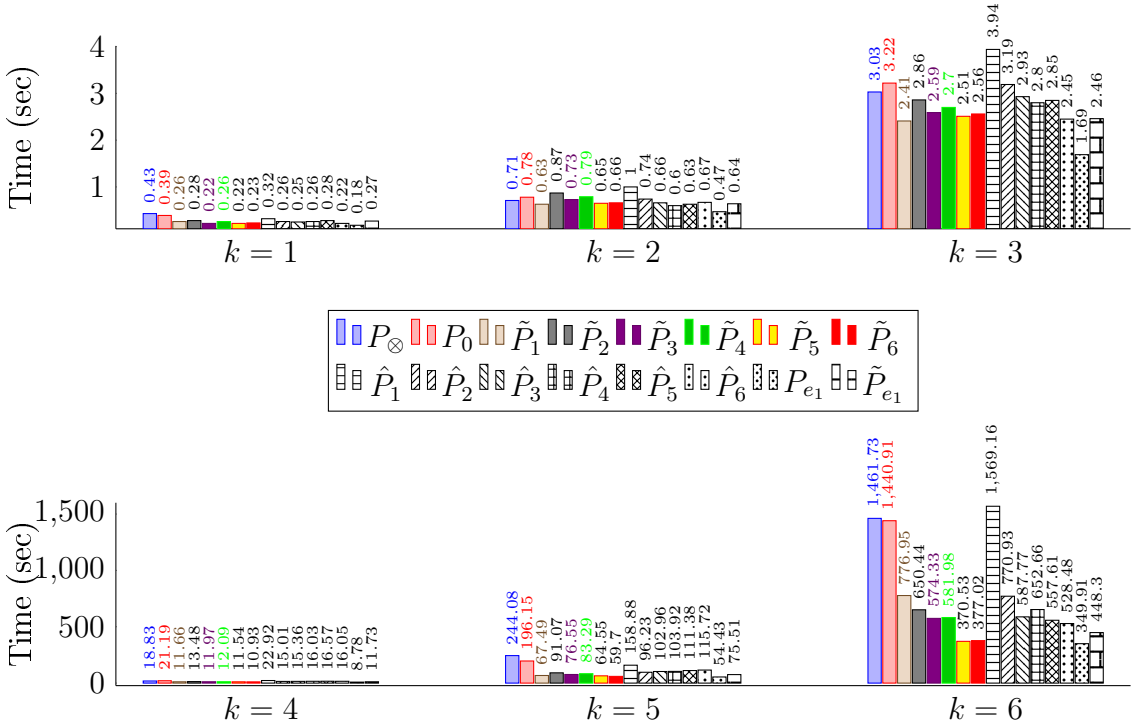


Figure 6.5: The runtimes by PCG with lognormal diffusion coefficient in Example 6.4.

case $\alpha_m = 2$: we get

$$a_{\alpha_m}(\boldsymbol{x}) = \frac{\sqrt{5}}{b_m^3(\boldsymbol{x})}\left((b_m^2(\boldsymbol{x})+3)\sinh\left(b_m(\boldsymbol{x})\right) - 3b_m(\boldsymbol{x})\cosh\left(b_m(\boldsymbol{x})\right)\right).$$

By basic calculus, we can check that $a_{\alpha_m}(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in D$.

Hence, we know that for $\boldsymbol{\alpha} \in \mathbb{I}_e$ where $\alpha_m \leq 2$ for all $m \leq M$, $a_{\boldsymbol{\alpha}}(\boldsymbol{x}) \geq 0$ on D and the test problem with loguniform coefficient satisfies condition (6.4).

Table 6.4 and Figure 6.5 show the PCG iteration counts and PCG runtimes, respectively, in the case of lognormal diffusion coefficient. The discretisation parameters are

		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_6$	P_{e_1}	$\tilde{P}_{e_1}$
$k=$	1	9	9	5	4	5	5	5	5	5	5	5	5	5	5	5	5
	2	11	11	6	5	5	5	5	5	6	6	6	5	5	5	6	6
	3	12	12	6	5	5	5	5	5	7	6	6	5	5	5	6	6
	4	12	12	7	6	6	5	5	5	7	6	6	6	6	6	6	7
	5	13	13	7	6	6	5	5	5	7	7	6	6	6	5	6	7
	6	13	13	7	6	6	5	5	5	7	7	6	6	6	6	7	7

Table 6.5: The numbers of iteration by PCG with loguniform diffusion coefficient in Example 6.4.

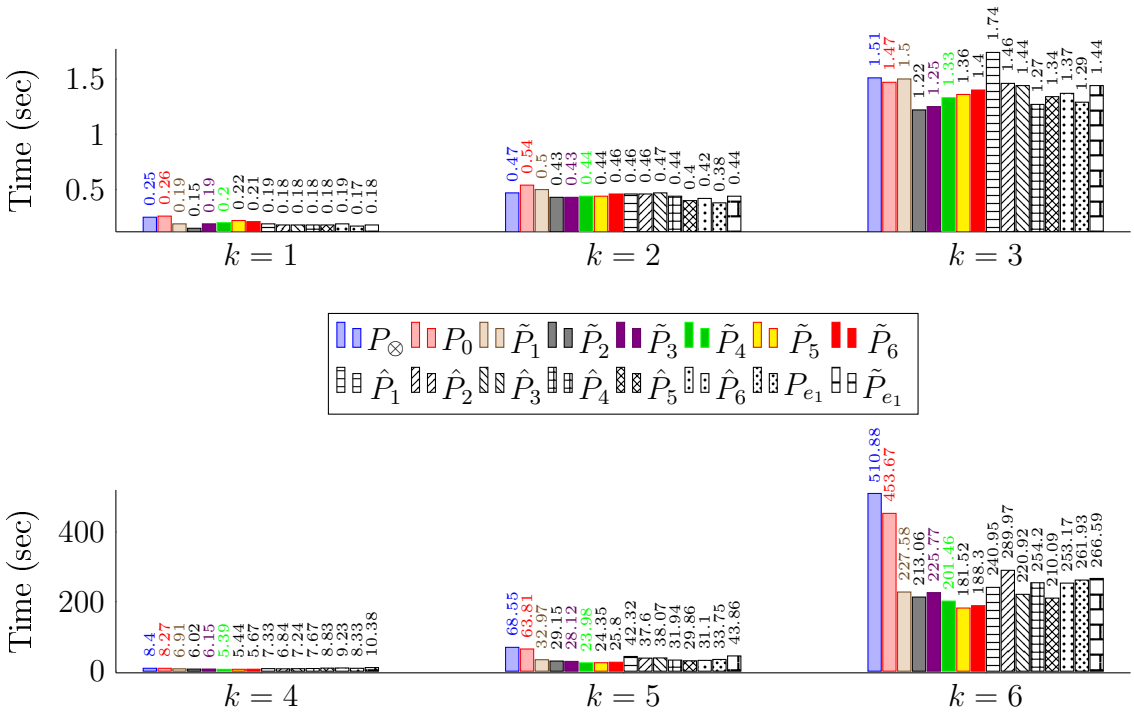


Figure 6.6: The runtimes by PCG with loguniform diffusion coefficient in Example 6.4.

set to be $h = 2^{-4}$, $M = 6$ and the range of k is from 1 to 6. The results show that our preconditioners, i.e., $\tilde{P}_r$, $\hat{P}_r$ and $\tilde{P}_{e_1}$, outperform P_0 and $P_\otimes$ in term of iteration counts, runtimes and total complexity. More precisely, $\tilde{P}_{e_1}$ and $\tilde{P}_r$ are more efficient than $\hat{P}_r$ and there are small improvements for $\tilde{P}_r$ and $\hat{P}_r$ with increasing r. Moreover, for all preconditioners, the results show that the PCG iteration counts rise with the parameter k. This is more pronounced for the mean-based preconditioner and the Kronecker product preconditioner, whereas the iteration counts by P_{e_1} go up very slowly with k. The preconditioner P_{e_1} seems to be independent of the parameter k.

h	Lognormal						Loguniform					
	$M=3$			$M=6$			$M=3$			$M=6$		
	$\tilde{P}_6$	$\hat{P}_6$	$\tilde{P}_{e_1}$	$\tilde{P}_6$	$\hat{P}_6$	$\tilde{P}_{e_1}$	$\tilde{P}_6$	$\hat{P}_6$	$\tilde{P}_{e_1}$	$\tilde{P}_6$	$\hat{P}_6$	$\tilde{P}_{e_1}$
2^{-3}	8	10	10	9	10	10	4	5	6	5	5	6
2^{-4}	9	11	10	10	12	11	5	5	6	5	5	6
2^{-5}	9	12	11	10	12	11	5	5	6	5	5	7
2^{-6}	10	12	11	10	13	11	5	5	6	5	5	7
2^{-7}	10	12	11	10	13	11	5	5	6	5	5	7

Table 6.6: PCG iteration counts for lognormal and loguniform diffusion coefficients with various h and M and $k = 3$.

The results in the case of loguniform diffusion coefficient are displayed in Table 6.5 and Figure 6.6 with the same discretisation parameters as in the case of the lognormal coefficient. The results shows that the PCG iteration counts and PCG runtimes by $\tilde{P}_r$, $\hat{P}_r$ and $\tilde{P}_{e_1}$ are only about a half of the iteration counts and runtimes by P_0 or $P_\otimes$. However, only $\tilde{P}_r$ and $\hat{P}_r$ are more efficient than P_0 and $P_\otimes$ in term of total complexity. Additionally, all preconditioners show optimality with respect to the parameter k in the case of the loguniform coefficient.

Table 6.6 shows that no significant increase of PCG iteration counts with discretisation parameters h and M. This means that our preconditioners are optimal with respect to h and M.

Example 6.5. Here, we assume that the diffusion coefficient a is the log-transformed coefficient in (6.17) with $N = 20$ where b_m are chosen to be the coefficients in Example 4.2. The first eight largest terms of $a_{\boldsymbol{\alpha}}$ with corresponding multi-indices in $\mathbb{I}_{2k}^M$ with $M = k = 6$ are shown in Table 6.7.

Table 6.7 displays the magnitudes of $a_{\boldsymbol{\alpha}}$ in the cases of lognormal and loguniform. The magnitudes of $a_{\boldsymbol{\alpha}}$ decay slower than the magnitudes of $a_{\boldsymbol{\alpha}}$ in the previous test problem.

In the case of the lognormal coefficient, the PCG iteration counts and PCG runtimes are shown in Table 6.8 and Figure 6.7, respectively. The iteration counts by $\hat{P}_r$, $\tilde{P}_{e_1}$ and $P_\otimes$ are about the same or more than half of those corresponding to P_0. Thus, the total complexities by $\hat{P}_r$ and $\tilde{P}_{e_1}$ are higher than those for the mean-based preconditioner. However, all of our preconditioners increase the efficiency of the solver in terms of com-

Lognormal		Loguniform	
$\boldsymbol{\alpha}$	$\|a_{\boldsymbol{\alpha}}\|_{L^\infty(D)}$	$\boldsymbol{\alpha}$	$\|a_{\boldsymbol{\alpha}}\|_{L^\infty(D)}$
$(0,0,0,0,0,0)$	2.8759	$(0,0,0,0,0,0)$	2.7663
$(1,0,0,0,0,0)$	0.8461	$(1,0,0,0,0,0)$	0.4674
$(0,0,1,0,0,0)$	0.4518	$(0,1,0,0,0,0)$	0.2506
$(0,1,0,0,0,0)$	0.4518	$(0,0,1,0,0,0)$	0.2506
$(0,0,0,1,0,0)$	0.2696	$(0,0,0,1,0,0)$	0.1497
$(0,0,0,0,1,0)$	0.2696	$(0,0,0,0,1,0)$	0.1497
$(0,0,0,0,0,1)$	0.2414	$(0,0,0,0,0,1)$	0.1340
$(2,0,0,0,0,0)$	0.1761	$(1,0,1,0,0,0)$	0.0367

Table 6.7: The first 8 terms of lognormal and loguniform diffusion coefficients in Example 6.5.

	$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_6$	P_{e_1}	$\tilde{P}_{e_1}$
$k=$ 1	7	6	7	7	6	6	5	2	7	7	6	6	6	5	7	7
2	10	9	9	9	7	7	6	5	9	9	7	7	7	6	9	10
3	11	13	11	10	8	8	7	6	11	10	8	8	8	7	11	11
4	13	16	13	11	9	8	7	6	13	11	10	10	10	10	13	13
5	14	18	14	13	10	9	8	7	14	12	11	11	11	10	14	14
6	15	21	15	13	10	10	9	8	15	14	14	14	14	14	15	15

Table 6.8: PCG iteration counts: lognormal diffusion coefficient in Example 6.5.

putational time. Furthermore, the modified truncation preconditioners outperform the others in term of iteration counts, runtimes and also the total computational cost. Also, note that the iteration counts by $\tilde{P}_r$, $\hat{P}_r$, P_{e_1}, and $\tilde{P}_{e_1}$ show mild dependence with the parameter k.

On the other hand, for the loguniform coefficient, the PCG iteration counts and PCG runtimes are displayed in Table 6.9 and Figure 6.8, respectively. It indicates that $\tilde{P}_r$ and $\hat{P}_r$ are more efficient than P_0 in terms of iteration counts and time consumption whereas the numbers of PCG iterations and PCG runtimes for $\tilde{P}_{e_1}$ are virtually identical to those for $P_\otimes$.

In this chapter, we presented preconditioners for non-affine-parametric diffusion coefficients. We extended the ideas from the case of affine-parametric diffusion coefficients. The truncation preconditioners for non-affine-parametric diffusion coefficients aim to capture the main features of the coefficient matrix but preserve the sparsity. The modified truncation preconditioners are an approximation of the truncation preconditioners to ensure

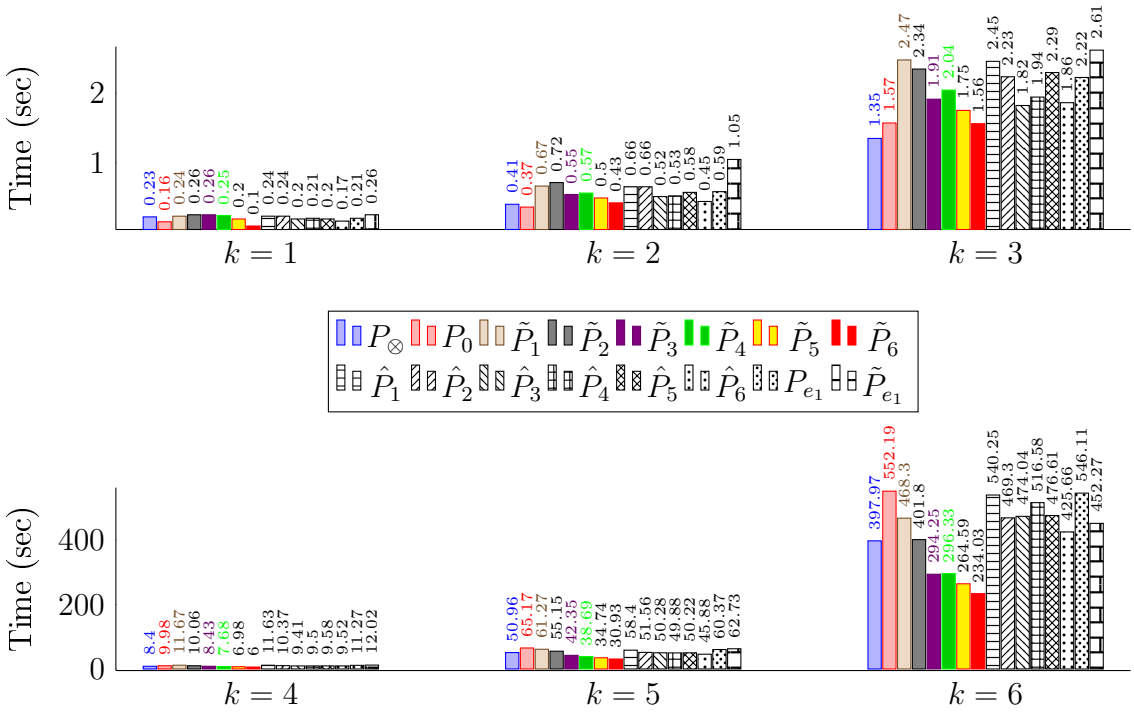


Figure 6.7: PCG runtimes: lognormal diffusion coefficient in Example 6.5.

		$P_\otimes$	P_0	$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$	$\tilde{P}_5$	$\tilde{P}_6$	$\hat{P}_1$	$\hat{P}_2$	$\hat{P}_3$	$\hat{P}_4$	$\hat{P}_5$	$\hat{P}_6$	P_{e_1}	$\tilde{P}_{e_1}$
$k =$	1	6	6	6	6	5	5	4	3	6	6	5	5	4	4	6	6
	2	7	8	7	7	6	5	5	4	7	7	6	5	5	4	7	7
	3	8	9	8	7	6	6	5	4	8	7	6	6	5	4	8	8
	4	8	10	8	7	6	6	5	4	8	7	6	6	5	4	8	8
	5	9	10	8	8	7	7	5	4	8	8	6	6	5	5	8	8
	6	9	11	9	8	6	6	5	4	9	8	7	6	6	5	9	9

Table 6.9: PCG iteration counts: loguniform diffusion coefficient in Example 6.5.

that the practical preconditioners are symmetric and positive definite. We also generalised the ideas of the domain decomposition on parametric domain for affine-parametric coefficients to enhance the parallelism of the preconditioner. Furthermore, we designed a preconditioner for log-transformed diffusion coefficients. The experiments indicate that the performance of our preconditioners are at least as good as those for the mean-based preconditioner and the Kronecker product preconditioner.

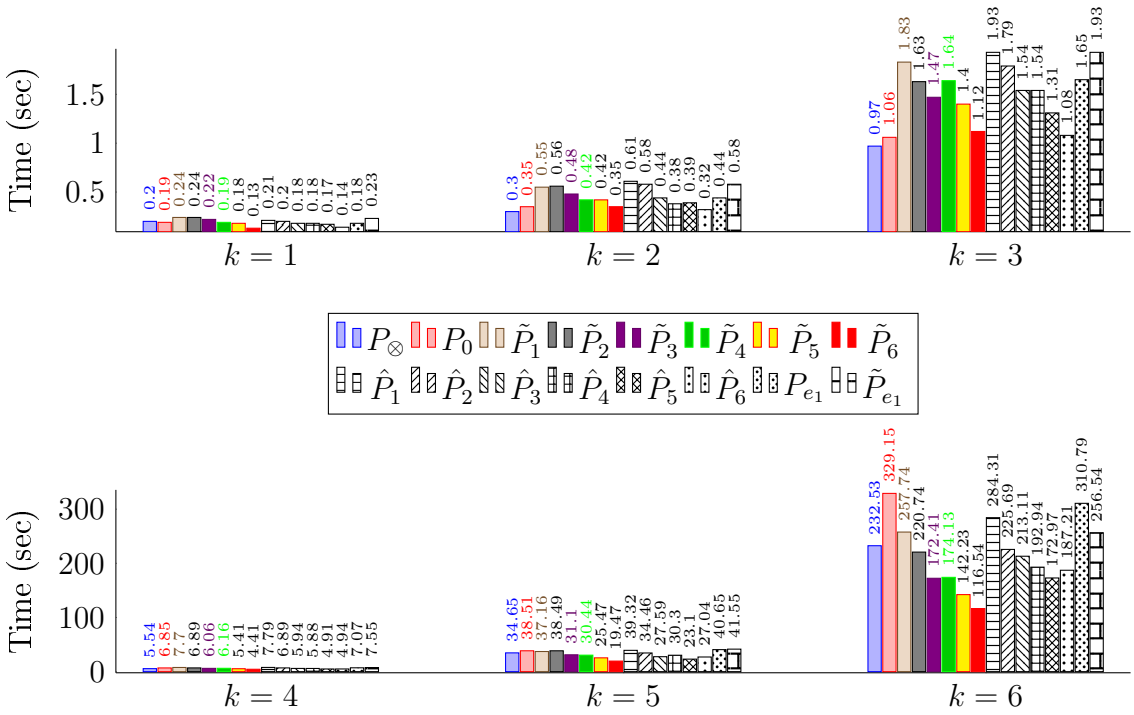


Figure 6.8: PCG runtimes: loguniform diffusion coefficient in Example 6.5.

CHAPTER 7

CONCLUDING REMARKS

In this thesis, we designed and analysed effective preconditioners for stochastic Galerkin linear systems which arise when solving elliptic partial differential equations models with random data. We considered in the cases of affine and non-affine parametric representations of diffusion coefficients.

Truncation preconditioners for affine-parametric diffusion coefficients rely on truncations of the diffusion coefficient. We establish a class of equivalent bilinear forms via truncated diffusion coefficients. This implies that the truncation preconditioners, which are constructed via the bilinear forms, are optimal with respect to discretisation parameters. However, the complexity of applying their inverses on a given vector is very high. To reduce the complexity, modified truncation preconditioners are obtained by approximating the truncation preconditioners using the symmetric block Gauss-Seidel approximation. The computational cost per iteration of modified truncation preconditioners is twice the computational cost of the mean-based preconditioner. Moreover, we also provided a spectral analysis of the modified truncation preconditioners to guarantee their optimality with respect to all discretisation parameters. The limitation of the modified truncation preconditioners is that, in general, their structure does not facilitate parallelism. However, parallelisation is possible for a very small number of terms in the preconditioners by permuting the unknowns corresponding to the parametric space in a certain way.

We also considered domain decomposition methods on the parametric domain for

affine-parametric diffusion coefficients. We developed and used a certain permutation technique so that the system matrix has a particular structure which is useful for parallel computing. We studied the pattern of the stochastic Galerkin matrix for the case of affine-parametric diffusion coefficient. Using graph theory, we introduced the concept of parametric mesh, including submesh and the interface of the parametric mesh as a tool for devising our permutation. This procedure leads to a 2-by-2 block matrix structure of the system matrix. According to the structure of the system matrix, we present two types of domain decomposition preconditioners: block preconditioners and block-diagonal preconditioners. Our study of the parametric mesh leads to the even-odd partitioning technique. The Schur complement is approximated in two ways: block-diagonal approximation and symmetric block Gauss-Seidel approximation. We introduced three versions of the even-odd preconditioners. Their complexities and spectral analyses were also presented. Additionally, the analyses indicate the optimality of the even-odd preconditioners with respect to discretisation parameters.

We also provided experiments to show that our preconditioners are promising choices due to their efficiency. All the experiments in the thesis were implemented using the MATLAB toolbox S-IFISS [112]. The experiments show that modified truncation preconditioner is an improvement over existing choices. It reduces the iteration counts significantly from those for the mean-based preconditioner and the Kronecker product preconditioner, especially in the case of fast decaying terms in the parametric representation of the diffusion coefficient. However, increasing the number of terms in the modified truncation preconditioners appears to improve their efficiency only negligibly. Therefore, we recommend choosing only two or three terms in the representation of the system matrix to be included in the truncation preconditioner to balance the efficiency and the computational cost. The even-odd preconditioners outperform the other preconditioners in terms of iteration counts and overall complexity. To be precise, the third version of the even-odd preconditioner, which is a combination of the block-structured preconditioner and a symmetric block Gauss-Seidel approximation of the Schur complement, has the least

iteration counts compared with the other preconditioners. In contrast, the second version of the even-odd preconditioners, which is a combination between a block-structured preconditioner and a block-diagonal approximation of the Schur complement, has the lowest computational cost due to its lower complexity per iteration. The numerical experiments also confirm the optimality of the modified truncation preconditioners and the even-odd preconditioners with respect to discretisation parameters.

For the case of non-affine parametric diffusion coefficient, we extended the two ideas from truncation preconditioners and domain decomposition preconditioners for the affine diffusion coefficients. Since the stochastic Galerkin matrix for the case of non-affine coefficient is block-dense, we introduced the truncation preconditioners in this case as a sparse approximation of the system matrix. Next, we modify the truncation preconditioners using symmetric block Gauss-Seidel approximation and obtain modified truncation preconditioners for non-affine parametric coefficients. We also studied the pattern of the truncation preconditioners with a view to parallelism and generalised the concept of parametric mesh to the case of non-affine coefficients. This lead to domain decomposition preconditioners for the case of non-affine parametric coefficients. In particular, we considered log-transformed coefficients, e.g., lognormal and loguniform coefficients. We employed the techniques developed in this chapter to derive another preconditioner and its spectral analysis for log-transformed coefficients. We also calculated the complexities of these preconditioners.

According to our experiments, the performance of the truncation preconditioners is superior in the case of lognormal and loguniform coefficients. The preconditioners for log-transformed coefficients perform very well only in the case of lognormal coefficients but still less efficient than truncation preconditioners. Although the efficiency of domain decomposition preconditioners is not outstanding, they perform at least on the same level as the mean-based preconditioner and the Kronecker product preconditioner. Moreover, the experiments show that these preconditioners, e.g., truncation preconditioners, domain decomposition preconditioners and preconditioner for log-transformed coefficients, are

optimal with respect to at least two discretisation parameters, i.e., the mesh size and the number of random variables.

Future research on this topic could include improvements to the modified truncation preconditioners for affine coefficients by devising other approximation techniques in order to decrease the complexity per iteration and improve the accuracy. For the domain decomposition method on the parametric domain, there are several possible extensions. Since the parametric mesh is defined via a graph, we need a sophisticated graph partitioning algorithm to partition the parametric mesh but minimise the size of the interface. The algorithm will lead to a new partitioning strategy. Moreover, the Schur complement approximation could be improved, for example, by using norm-equivalence and other techniques. To achieve optimal performance, we need to balance between the accuracy of the Schur complement approximation and the complexity per iteration. For the case of non-affine parametric coefficients, a sparse gPC expansion, which is positive on the spatial domain and the parametric domain, is important. Similarly to the domain decomposition methods for the case of affine-parametric coefficients, we could also consider the extension of the partitioning strategy and the Schur complement approximation to the case of non-affine parametric coefficients in order to improve the efficiency of the preconditioners. Finally, once could also consider extending our results to other model problems such as optimal control problems, incompressible elasticity equations, Navier-Stokes equations, saddle point problems.

BIBLIOGRAPHY

[1] S. Adhikari, *Doubly Spectral Stochastic Finite-Element Method for Linear Structural Dynamics*, Journal of Aerospace Engineering, 24 (2011), pp. 264–276.

[2] N. Agarwal and N. R. Aluru, *Stochastic Modeling of Coupled Electromechanical Interaction for Uncertainty Quantification in Electrostatically Actuated MEMS*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 3456–3471.

[3] M. Anders and M. Hori, *Three-Dimensional Stochastic Finite Element Method for Elasto-Plastic Bodies*, International Journal for Numerical Methods in Engineering, 51 (2001), pp. 449–478.

[4] A. Atangana and P. Vermeulen, *Analytical Solutions of a Space-Time Fractional Derivative of Groundwater Flow Equation*, in Abstract and Applied Analysis, vol. 2014, Hindawi, 2014.

[5] O. Axelsson and R. Blaheta, *Preconditioning of Matrices Partitioned in 2 × 2 Block Form: Eigenvalue Estimates and Schwarz DD for Mixed FEM*, Numerical Linear Algebra with Applications, 17 (2010), pp. 787–810.

[6] I. Babuška and P. Chatzipantelidis, *On Solving Elliptic Stochastic Partial Differential Equations*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 4093–4122.

[7] I. Babuska, R. Tempone, and G. E. Zouraris, *Galerkin Finite Element Approximations of Stochastic Elliptic Partial Differential Equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[8] I. Babuška, R. Tempone, and G. E. Zouraris, *Solving Elliptic Boundary Value Problems with Uncertain Coefficients by the Finite Element Method: the Stochastic Formulation*, Computer methods in applied mechanics and engineering, 194 (2005), pp. 1251–1294.

[9] I. Babuška, F. Nobile, and R. Tempone, *A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.

[10] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone, *Stochastic Spectral Galerkin and Collocation Methods for PDEs with Random Coefficients: a Numerical Comparison*, in Spectral and high order methods for partial differential equations, Springer, 2011, pp. 43–62.

[11] A. Barth, C. Schwab, and N. Zollinger, *Multi-Level Monte Carlo Finite Element Method for Elliptic PDEs with Stochastic Coefficients*, Numerische Mathematik, 119 (2011), pp. 123–161.

[12] M. Bebendorf, *Efficient Inversion of the Galerkin Matrix of General Second-order Elliptic Operators with Nonsmooth Coefficients*, Mathematics of Computation, 74 (2004), pp. 1179–1200.

[13] M. Bebendorf, *Approximate Inverse Preconditioning of Finite Element Discretizations of Elliptic Operators with Nonsmooth Coefficients*, SIAM journal on matrix analysis and applications, 27 (2006), pp. 909–929.

[14] ——, *Why Finite Element Discretizations Can Be Factored by Triangular Hierarchical Matrices*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1472–1494.

[15] M. Bebendorf, *Low-Rank Approximation of Elliptic Boundary Value Problems with High-Contrast Coefficients*, SIAM Journal on Mathematical Analysis, 48 (2016), pp. 932–949.

[16] M. Bebendorf, M. Bollhöfer, and M. Bratsch, *Hierarchical Matrix Approximation with Blockwise Constraints*, BIT Numerical Mathematics, 53 (2013), pp. 311–339.

[17] ——, *On the Spectral Equivalence of Hierarchical Matrix Preconditioners for Elliptic Problems*, Mathematics of Computation, 85 (2016), pp. 2839–2861.

[18] K. Beddek, Y. Le Menach, S. Clenet, and O. Moreau, *3-D Stochastic Spectral Finite-Element Method in Static Electromagnetism Using Vector Potential Formulation*, IEEE Transactions on Magnetics, 47 (2011), pp. 1250–1253.

[19] P. Benner, A. Onwunta, and M. Stoll, *Block-Diagonal Preconditioning for Optimal Control Problems Constrained by PDEs with Uncertain Inputs*, SIAM Journal on Matrix Analysis and Applications, 37 (2016), pp. 491–518.

[20] A. Bespalov, D. Loghin, and R. Youngnoi, *Truncation Preconditioners for Stochastic Galerkin Finite Element Discretizations*, arXiv preprint arXiv:2006.06428, (2020).

[21] A. Bespalov, D. Praetorius, L. Rocchi, and M. Ruggeri, *Convergence of Adaptive Stochastic Galerkin FEM*, SIAM Journal on Numerical Analysis, 57 (2019), pp. 2359–p2382.

[22] A. Bespalov and D. Silvester, *Efficient Adaptive Stochastic Galerkin Methods for Parametric Operator Equations*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2118–A2140.

[23] M. Bieri, R. Andreev, and C. Schwab, *Sparse Tensor Discretization of Elliptic SPDEs*, SIAM Journal on Scientific Computing, 31 (2010), pp. 4281–4304.

[24] G. Blatman and B. Sudret, *An Adaptive Algorithm to Build Up Sparse Polynomial Chaos Expansions for Stochastic Finite Element Analysis*, Probabilistic Engineering Mechanics, 25 (2010), pp. 183–197.

[25] G. Blatman and B. Sudret, *Adaptive Sparse Polynomial Chaos Expansion Based on Least Angle Regression*, Journal of Computational Physics, 230 (2011), pp. 2345 – 2367.

[26] S. Börm, L. Grasedyck, and W. Hackbusch, *An Introduction to Hierarchical Matrices with Applications*, Engineering analysis with boundary elements, 27 (2003), pp. 405–422.

[27] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Cambridge University Press, 2007.

[28] S. Brenner and R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15, Springer Science & Business Media, 2007.

[29] M. Brezina, A. Doostan, T. Manteuffel, S. McCormick, and J. Ruge, *Smoothed Aggregation Algebraic Multigrid for Stochastic PDE Problems With Layered Materials*, Numer. Linear Algebra Appl., 21 (2014), pp. 239–255.

[30] R. E. Caflisch, *Monte Carlo And Quasi-Monte Carlo Methods*, in Acta numerica, 1998, vol. 7 of Acta Numer., Cambridge Univ. Press, Cambridge, 1998, pp. 1–49.

[31] A. Cañavate-Grimal, A. Falcó, P. Calderón, and I. Payá-Zaforteza, *On The Use Of Stochastic Spectral Methods In Deep Excavation Inverse Problems*, Computers & Structures, 159 (2015), pp. 41–60.

[32] J. Carrera, *An Overview of Uncertainties In Modelling Groundwater Solute Transport*, Journal of contaminant hydrology, 13 (1993), pp. 23–48.

[33] Y. Cengel and M. Ãzişik, *Integrals Involving Legendre Polynomials That Arise in The Solution Of Radiation Transfer*, Journal of Quantitative Spectroscopy and Radiative Transfer, 31 (1984), pp. 215–219.

[34] J. Charrier, *Strong and Weak Error Estimates For The Solutions Of Elliptic Partial Differential Equations With Random Coefficients*, Research Report RR-7300, INRIA, June 2010.

[35] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo Methods and Applications To Elliptic PDEs with Random Coefficients*, Computing and Visualization in Science, 14 (2011), p. 3.

[36] A. J. Crowder, C. E. Powell, and A. Bespalov, *Efficient Adaptive Multilevel Stochastic Galerkin Approximation Using Implicit A Posteriori Error Estimation*, SIAM Journal on Scientific Computing, 41 (2019), pp. A1681–A1705.

[37] M. K. Deb, I. M. Babuška, and J. T. Oden, *Solution of Stochastic Partial Differential Equations Using Galerkin Finite Element Techniques*, Computer Methods in Applied Mechanics and Engineering, 190 (2001), pp. 6359–6372.

[38] J. Dick, F. Y. Kuo, and I. H. Sloan, *High-Dimensional Integration: The Quasi-Monte Carlo Way*, Acta Numerica, 22 (2013), pp. 133–288.

[39] P. Dostert, Y. Efendiev, and T. Y. Hou, *Multiscale Finite Element Methods for Stochastic Porous Media Flow Equations And Application To Uncertainty Quantification*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 3445–3455.

[40] Y. D'Yakonov, *The Construction of Iterative Methods Based on The Use of Spectrally Equivalent Operators*, USSR Computational Mathematics and Mathematical Physics, 6 (1966), pp. 14–46.

[41] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander, *Adaptive Stochastic Galerkin FEM*, Computer Methods in Applied Mechanics and Engineering, 270 (2014), pp. 247–269.

[42] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander, *A Convergent Adaptive Stochastic Galerkin Finite Element Method with Quasi-Optimal Spatial Meshes*, ESAIM: Mathematical Modelling and Numerical Analysis, 49 (2015), pp. 1367–1398.

[43] M. Eigel, M. Pfeffer, and R. Schneider, *Adaptive Stochastic Galerkin FEM With Hierarchical Tensor Representations*, Numerische Mathematik, 136 (2016), pp. 765–803.

[44] M. Eldred, *Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design*, in 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, may 2009, p. 2274.

[45] M. Eldred and J. Burkardt, *Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification*, in 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2009, p. 976.

[46] H. Elman and D. Furnival, *Solving the Stochastic Steady-State Diffusion Problem Using Multigrid*, IMA J. Numer. Anal., 27 (2007), pp. 675–688.

[47] H. C. Elman, O. G. Ernst, D. P. OâLeary, and M. Stewart, *Efficient Iterative Algorithms for The Stochastic Finite Element Method with Application to Acoustic Scattering*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 1037–1055.

[48] H. C. Elman, C. W. Miller, E. T. Phipps, and R. S. Tuminaro, *Assessment of Collocation and Galerkin Approaches to Linear Diffusion Equations with Random Data*, International Journal for Uncertainty Quantification, 1 (2011).

[49] H. C. Elman and T. Su, *A Low-Rank Multigrid Method for The Stochastic Steady-State Diffusion Problem*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 492–509.

[50] 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 J. Sci. Comput., 31 (2008/09), pp. 1424–1447.

[51] O. G. Ernst and E. Ullmann, *Stochastic Galerkin Matrices*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1848–1872.

[52] V. Faber, T. A. Manteuffel, and S. V. Parter, *On the Theory of Equivalent Operators and Application to the Numerical Solution of Uniformly Elliptic Partial Differential Equations*, Adv. Appl. Math., 11 (1990), pp. 109–163.

[53] J. Faragher, *Probabilistic Methods for the Quantification of Uncertainty and Error in Computational Fluid Dynamic Simulations*, (2004).

[54] M. Faustmann, J. M. Melenk, and D. Praetorius, *Existence of H-matrix Approximants to the Inverses of BEM Matrices: The Simple-layer Operator*, Mathematics of Computation, 85 (2015), pp. 119–152.

[55] W. Gautschi, *Orthogonal Polynomials : Computation and Approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford New York, 2004. Oxford Science Publications.

[56] S. E. Geneser, R. M. Kirby, and R. S. MacLeod, *Application of Stochastic Finite Element Methods to Study the Sensitivity of ECG Forward Modeling to Organ Conductivity*, IEEE Transactions on Biomedical Engineering, 55 (2008), pp. 31–40.

[57] R. Ghanem, *Ingredients for A General Purpose Stochastic Finite Elements Implementation*, Computer Methods in Applied Mechanics and Engineering, 168 (1999), pp. 19 – 34.

[58] R. Ghanem and S. Dham, *Stochastic Finite Element Analysis for Multiphase Flow in Heterogeneous Porous Media*, Transport in porous media, 32 (1998), pp. 239–262.

[59] R. Ghanem, G. Saad, and A. Doostan, *Efficient Solution of Stochastic Systems: Application to The Embankment Dam Problem*, Structural safety, 29 (2007), pp. 238–251.

[60] R. Ghanem and P. D. Spanos, *Polynomial Chaos in Stochastic Finite Elements*, Journal of Applied Mechanics, 57 (1990), pp. 197–202.

[61] R. G. Ghanem, *Uncertainty Quantification in Computational and Prediction Science*, International Journal for Numerical Methods in Engineering, 80 (2009), pp. 671–672.

[62] R. G. Ghanem and A. Doostan, *On the Construction and Analysis of Stochastic Models: Characterization and Propagation of The Errors Associated with Limited Data*, Journal of Computational Physics, 217 (2006), pp. 63–81.

[63] R. G. Ghanem and R. M. Kruger, *Numerical Solution of Spectral Stochastic Finite Element Systems*, Comput. Methods Appl. Mech. Engrg., 129 (1996), pp. 289–303.

[64] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

[65] D. Ghosh, P. Avery, and C. Farhat, *A FETI-Preconditioned Conjugate Gradient Method for Large-Scale Stochastic Finite Element Problems*, International journal for numerical methods in engineering, 80 (2009), pp. 914–931.

[66] M. B. Giles, *Multilevel Monte Carlo Methods*, Acta Numerica, 24 (2015), pp. 259–328.

[67] I. G. Graham, F. Y. Kuo, J. A. Nichols, R. Scheichl, C. Schwab, and I. H. Sloan, *Quasi-Monte Carlo Finite Element Methods for Elliptic PDEs With Lognormal Random Coefficients*, Numerische Mathematik, 131 (2015), pp. 329–368.

[68] 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 (2011), pp. 3668–3694.

[69] L. Grasedyck, R. Kriemann, and S. L. Borne, *Parallel Black Box H-LU Preconditioning for Elliptic Boundary Value Problems*, Computing and Visualization in Science, 11 (2008), pp. 273–291.

[70] M. D. Gunzburger, C. G. Webster, and G. Zhang, *Stochastic Finite Element Methods for Partial Differential Equations with Random Input Data*, Acta Numer., 23 (2014), pp. 521–650.

[71] W. Hackbusch, *A Sparse Matrix Arithmetic Based on H-Matrices. Part I: Introduction to H-Matrices*, Computing, 62 (1999), pp. 89–108.

[72] A.-L. Haji-Ali, F. Nobile, and R. Tempone, *Multi-Index Monte Carlo: When Sparsity Meets Sampling*, Numerische Mathematik, 132 (2016), pp. 767–806.

[73] C. Jin and X.-C. Cai, *A Preconditioned Recycling GMRES Solver for Stochastic Helmholtz Problems*, Commun. Comput. Phys., 6 (2009), pp. 342–353.

[74] C. Jin, X.-C. Cai, and C. Li, *Parallel Domain Decomposition Methods for Stochastic Elliptic Equations*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2096–2114.

[75] C. H. Jones, *Generalized Hockey Stick Identities and n-Dimensional Blockwalking*, Fibonacci Q., 34 (1996), pp. 280–288.

[76] A. Keese, *Review of Recent Developments in the Numerical Solution of Stochastic Partial Differential Equations (Stochastic Finite Elements)*, Informatik-Berichte der Technischen Universität Braunschweig, 2003-06 (2003).

[77] ——, *Numerical Solution of Systems with Stochastic Uncertainties: A General Purpose Framework for Stochastic Finite Elements*, PhD thesis, 2004.

[78] A. Khan, C. E. Powell, and D. J. Silvester, *Robust Preconditioning for Stochastic Galerkin Formulations of Parameter-Dependent Nearly Incompressible Elasticity Equations*, SIAM Journal on Scientific Computing, 41 (2019), pp. A402–A421.

[79] P. K. Kitanidis, *Groundwater Flow in Heterogeneous Formations*, Subsurface Flow and Transport: A Stochastic Approach, (2005), p. 83.

[80] O. Knio and O. Le Maitre, *Uncertainty Propagation In CFD Using Polynomial Chaos Decomposition*, Fluid dynamics research, 38 (2006), p. 616.

[81] O. M. Knio, H. N. Najm, R. G. Ghanem, et al., *A Stochastic Projection Method for Fluid Flow: I. Basic Formulation*, Journal of computational Physics, 173 (2001), pp. 481–511.

[82] F. Y. Kuo, C. Schwab, and I. H. Sloan, *Quasi-Monte Carlo Finite Element Methods for A Class of Elliptic Partial Differential Equations with Random Coefficients*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 3351–3374.

[83] O. Le Maitre, O. Knio, B. Debusschere, H. Najm, and R. Ghanem, *A Multigrid Solver for Two-Dimensional Stochastic Diffusion Equations*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 4723–4744.

[84] O. Le Maître and O. M. Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Springer Science & Business Media, 2010.

[85] B. Lee, *Parallel Preconditioners and Multigrid Solvers for Stochastic Polynomial Chaos Discretizations Of the Diffusion Equation at The Large Scale*, Numer. Linear Algebra Appl., 23 (2016), pp. 5–36.

[86] M. Leinonen, H. Hakula, and N. Hyvönen, *Application of Stochastic Galerkin FEM to The Complete Electrode Model of Electrical Impedance Tomography*, Journal of Computational Physics, 269 (2014), pp. 181–200.

[87] M. Loeve, *Probability Theory II*, Springer New York, 1994.

[88] D. Loghin and A. J. Wathen, *Analysis of Preconditioners for Saddle-Point Problems*, SIAM Journal on Scientific Computing, 25 (2004), pp. 2029–2049.

[89] G. J. Lord, C. E. Powell, and T. Shardlow, *An Introduction to Computational Stochastic PDEs*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2014.

[90] P. S. Mohan, P. B. Nair, and A. J. Keane, *Multi-Element Stochastic Reduced Basis Methods*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 1495–1506.

[91] P. B. Nair and A. J. Keane, *Stochastic Reduced Basis Methods*, AIAA journal, 40 (2002), pp. 1653–1664.

[92] H. N. Najm, *Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics*, Annual review of fluid mechanics, 41 (2009), pp. 35–52.

[93] H. N. Najm, B. J. Debusschere, Y. M. Marzouk, S. Widmer, and O. Le Maître, *Uncertainty Quantification in Chemical Systems*, International journal for numerical methods in engineering, 80 (2009), pp. 789–814.

[94] A. Nouy, *Recent Developments in Spectral Stochastic Methods for The Numerical Solution of Stochastic Partial Differential Equations*, Archives of Computational Methods in Engineering, 16 (2009), pp. 251–285.

[95] M. Pellissetti and R. Ghanem, *Iterative Solution of Systems of Linear Equations Arising in The Context of Stochastic Finite Elements*, Advances in Engineering Software, 31 (2000), pp. 607 – 616.

[96] C. E. Powell, *Robust Preconditioning for Second-Order Elliptic PDEs with Random Field Coefficients*, (2006). Technical report, Manchester Institute for Mathematical Sciences,School of Mathematics, University o f Manchester.

[97] C. E. Powell and H. C. Elman, *Block-Diagonal Preconditioning for Spectral Stochastic Finite-Element Systems*, IMA Journal of Numerical Analysis, 29 (2008), pp. 350–375.

[98] C. E. Powell and D. J. Silvester, *Preconditioning Steady-State Navier–Stokes Equations with Random Data*, SIAM Journal on Scientific Computing, 34 (2012), pp. A2482–A2506.

[99] C. E. Powell and E. Ullmann, *Preconditioning Stochastic Galerkin Saddle Point Systems*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2813–2840.

[100] I. Pultarová, *Block and Multilevel Preconditioning for Stochastic Galerkin Problems with Lognormally Distributed Parameters and Tensor Product Polynomials*, Int. J. Uncertain. Quantif., 7 (2017), pp. 441–462.

[101] M. Reagan, H. Najm, B. Debusschere, O. Le Maître, O. Knio, and R. Ghanem, *Spectral Stochastic Uncertainty Quantification in Chemical Systems*, Combustion Theory and Modelling, 8 (2004), pp. 607–632.

[102] M. T. Reagana, 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 (2003), pp. 545–555.

[103] E. Rosseel, T. Boonen, and S. Vandewalle, *Algebraic Multigrid for Stationary and Time-Dependent Partial Differential Equations with Stochastic Coefficients*, Numerical Linear Algebra with Applications, 15 (2008), pp. 141–163.

[104] E. Rosseel and S. Vandewalle, *Iterative Solvers for The Stochastic Finite Element Method*, SIAM J. Sci. Comput., 32 (2010), pp. 372–397.

[105] R. Y. Rubinstein and D. P. Kroese, *Simulation and The Monte Carlo Method*, vol. 10, John Wiley & Sons, 2016.

[106] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed., 2003.

[107] S. K. Sachdeva, P. B. Nair, and A. J. Keane, *Comparative Study of Projection Schemes for Stochastic Finite Element Analysis*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 2371–2392.

[108] ——, *Hybridization of Stochastic Reduced Basis Methods with Polynomial Chaos Expansions*, Probabilistic Engineering Mechanics, 21 (2006), pp. 182–192.

[109] M. Schevenels, G. Lombaert, and G. Degrande, *Application of The Stochastic Finite Element Method For Gaussian And Non-Gaussian Systems*, in ISMA2004 International Conference on Noise and Vibration Engineering, KATHOLIEKE UNIV LEUVEN, DEPT WERKTUIGKUNDE, 2004, pp. 3299–3314.

[110] C. Schwab and C. J. Gittelson, *Sparse Tensor Discretizations of High-Dimensional Parametric and Stochastic PDEs*, Acta Numer., 20 (2011), pp. 291–467.

[111] D. Silvester and Pranjal, *An Optimal Solver for Linear Systems Arising from Stochastic FEM Approximation of Diffusion Equations with Random Coefficients*, SIAM/ASA Journal on Uncertainty Quantification, 4 (2016), pp. 298–311.

[112] D. J. Silvester, A. Bespalov, and C. E. Powell, *Stochastic IFISS (S-IFISS), version* 1.04, October 2017. Available online at `http://www.manchester.ac.uk/ifiss/sifiss.html`.

[113] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, vol. 12, SIAM, 2013.

[114] B. Sousedík and H. C. Elman, *Stochastic Galerkin Methods for The Steady-State Navier–Stokes Equations*, Journal of Computational Physics, 316 (2016), pp. 435–452.

[115] B. Sousedík and R. G. Ghanem, *Truncated Hierarchical Preconditioning for The Stochastic Galerkin FEM*, Int. J. Uncertain. Quantif., 4 (2014), pp. 333–348.

[116] B. Sousedík, R. G. Ghanem, and E. T. Phipps, *Hierarchical Schur Complement Preconditioner for The Stochastic Galerkin Finite Element Methods*, Numer. Linear Algebra Appl., 21 (2014), pp. 136–151.

[117] G. Stefanou, *The Stochastic Finite Element Method: Past, Present and Future*, Computer methods in applied mechanics and engineering, 198 (2009), pp. 1031–1051.

[118] W. Subber and S. Loisel, *Schwarz Preconditioners for Stochastic Elliptic PDEs*, Comput. Methods Appl. Mech. Engrg., 272 (2014), pp. 34–57.

[119] W. Subber and A. Sarkar, *Domain Decomposition of Stochastic PDEs: A Novel Preconditioner and Its Parallel Performance*, in International Symposium on High Performance Computing Systems and Applications, Springer, 2009, pp. 251–268.

[120] ——, *Dual-Primal Domain Decomposition Method for Uncertainty Quantification*, Computer Methods in Applied Mechanics and Engineering, 266 (2013), pp. 112–124.

[121] W. Subber and A. Sarkar, *A Domain Decomposition Method of Stochastic PDEs: An Iterative Solution Techniques Using A Two-Level Scalable Preconditioner*, Journal of Computational Physics, 257 (2014), pp. 298–317.

[122] D. M. Tartakovsky, *Assessment and Management of Risk in Subsurface Hydrology: A Review and Perspective*, Advances in Water Resources, 51 (2013), pp. 247–260.

[123] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further Analysis of Multilevel Monte Carlo Methods for Elliptic PDEs with Random Coefficients*, Numerische Mathematik, 125 (2013), pp. 569–600.

[124] R. Tipireddy, P. Stinis, and A. M. Tartakovsky, *Basis Adaptation and Domain Decomposition for Steady-State Partial Differential Equations with Random Coefficients*, Journal of Computational Physics, 351 (2017), pp. 203–215.

[125] E. Ullmann, *A Kronecker Product Preconditioner for Stochastic Galerkin Finite Element Discretizations*, SIAM Journal on Scientific Computing, 32 (2010), pp. 923–946.

[126] E. Ullmann, H. C. Elman, and O. G. Ernst, *Efficient Iterative Solvers for Stochastic Galerkin Discretizations of Log-Transformed Random Diffusion Problems*, SIAM Journal on Scientific Computing, 34 (2012), pp. A659–A682.

[127] E. Ullmann and C. E. Powell, *Solving Log-Transformed Random Diffusion Problems by Stochastic Galerkin Mixed Finite Element Methods*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 509–534.

[128] N. Wiener, *The Homogeneous Chaos*, American Journal of Mathematics, 60 (1938), pp. 897–936.

[129] D. Xiu, *Efficient Collocational Approach for Parametric Uncertainty Analysis*, Communications in computational physics, 2 (2007), pp. 293–309.

[130] ——, *Fast Numerical Methods for Stochastic Computations: A Review*, Communications in computational physics, 5 (2009), pp. 242–272.

[131] ——, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton university press, 2010.

[132] D. Xiu and J. S. Hesthaven, *High-Order Collocation Methods for Differential Equations with Random Inputs*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1118–1139.

[133] D. Xiu and G. E. Karniadakis, *Modeling Uncertainty in Steady State Diffusion Problems Via Generalized Polynomial Chaos*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 4927–4948.

[134] D. Xiu and G. E. Karniadakis, *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal on Scientific Computing, 24 (2002), pp. 619–644.

[135] D. Xiu, D. Lucor, C.-H. Su, and G. E. Karniadakis, *Stochastic Modeling of Flow-Structure Interactions Using Generalized Polynomial Chaos*, J. Fluids Eng., 124 (2002), pp. 51–59.

[136] X. Zhu, *Uncertainty Simulation Using Domain Decomposition and Stratified Sampling*, PhD thesis, Carleton University, 2006.