

ABSTRACT

Title of dissertation: FAST AND OPTIMAL SOLUTION
ALGORITHMS FOR PARAMETERIZED
PARTIAL DIFFERENTIAL EQUATIONS

Kookjin Lee, Doctor of Philosophy, 2017

Dissertation directed by: Professor Howard Elman
Department of Computer Science

This dissertation presents efficient and optimal numerical algorithms for the solution of parameterized partial differential equations (PDEs) in the context of *stochastic Galerkin* discretization. The stochastic Galerkin method often leads to a large coupled system of algebraic equations, whose solution is computationally expensive to compute using traditional solvers. For efficient computation of such solutions, we present low-rank iterative solvers, which compute low-rank approximations to the solutions of those systems while not losing much accuracy. We first introduce a low-rank iterative solver for linear systems obtained from the stochastic Galerkin discretization of linear elliptic parameterized PDEs. Then we present a low-rank nonlinear iterative solver for efficiently computing approximate solutions of nonlinear parameterized PDEs, the incompressible Navier–Stokes equations.

Along with the computational issue, the stochastic Galerkin method suffers

from an optimality issue. The method, in general, does not minimize the solution error in any measure. To address this issue, we present an optimal projection method, a least-squares Petrov–Galerkin (LSPG) method. The proposed method is optimal in the sense that it produces the solution that minimizes a weighted ℓ^2 -norm of the solution error over all solutions in a given finite-dimensional subspace. The method can be adapted to minimize the solution error in different weighted ℓ^2 -norms by simply choosing a specific weighting function within the least-squares formulation.

FAST AND OPTIMAL SOLUTION ALGORITHMS FOR
PARAMETERIZED PARTIAL DIFFERENTIAL EQUATIONS

by

Kookjin Lee

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Howard Elman, Chair/Advisor
Professor Thomas Goldstein,
Professor David Jacobs,
Professor Ricardo Nochetto,
Professor Alan Sussman

© Copyright by
Kookjin Lee
2017

Acknowledgments

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. Without their help, my academic journey could not have started and continued. I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I would like to thank my advisor, Howard Elman. I thank him for his patience and trust in me over my graduate studies. I am truly grateful for his guidance, thoughtful advice, and scholarly inspiration. I have been very fortunate to learn so much from his academic insight as well as his wisdom of life.

Kevin Carlberg at Sandia National Laboratories provided me with the internship opportunity. I am grateful to him for collaborating on a very interesting research subject and being a great mentor with his passion on research. I also enjoyed working with Bedřich Sousedík on various interesting research projects. I appreciate his sharing brilliant ideas, which made the collaboration even more delightful.

I owe my gratitude to Thomas Goldstein, David Jacobs, Alan Sussman, and Ricardo Nochetto for taking their time to serve on my dissertation committee.

Most importantly, none of my academic achievement during my graduate years would have been possible without love and support of my family in Seoul. To my parents, I owe much for having a faith in me and supporting me.

Above all, I thank my dear wife Heewon. Her love, emotional support, and thoughtful advices coming from her own experiences have helped me go through all

the hardships during my graduate studies. She has always been by my side and often brought unexpected and pleasant troubles, which enrich my life. This dissertation is dedicated to her for her sacrifice, faith, and love.

Table of Contents

Acknowledgements	ii
List of Tables	vi
List of Figures	viii
List of Abbreviations	x
1 Introduction	1
1.1 Outline of Thesis	9
2 Background: The stochastic Galerkin method	11
2.1 Overview of the stochastic Galerkin method	11
2.2 Discretization	14
2.3 Iterative solvers for stochastic Galerkin systems	19
3 Low-rank approximation method for linear PDEs	25
3.1 Introduction	25
3.2 Stochastic Galerkin formulation in tensor notation	28
3.3 A preconditioned projection method in tensor format	32
3.4 Truncation methods	35
3.5 Numerical experiments	40
3.6 Statistical Computations	58
3.7 Conclusion	60
4 Low-rank approximation method for parameterized Navier–Stokes equations	62
4.1 Introduction	62
4.2 Stochastic Navier–Stokes equations	64
4.3 Low-rank Newton–Krylov method	71
4.4 Inexact nonlinear iteration	81
4.5 Numerical results	83
4.6 Conclusion	98

5	Stochastic Least-Square Petrov Galerkin method	100
5.1	Introduction	100
5.2	Spectral methods for parameterized linear systems	102
5.3	Stochastic least-squares Petrov–Galerkin method	107
5.4	Error analysis	112
5.5	Numerical experiments	114
5.6	Conclusion	132
6	Conclusion	135
	Bibliography	137

List of Tables

2.1	Probability distribution and the type of gPC basis.	16
3.1	Rank (κ) of coarse-grid solutions satisfying ϵ^c of (3.20), and CPU time (t_c) for coarse-grid computation using the PGD method, for varying γ and M	42
3.2	CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ using the preconditioned low-rank projection method. Numbers of GMRES cycles are shown in parentheses.	43
3.3	CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-6}$ using the preconditioned low-rank projection method. Numbers of GMRES cycles are shown in parentheses.	43
3.4	Number of degrees of freedom of the fine-grid discretizations with $p = 3$, for varying spatial-grid refinement level, ℓ , and number of random variables, M	44
3.5	CPU times t to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection method for varying maximal polynomial degree p (stochastic dofs, n_ξ , in the parenthesis).	45
3.6	CPU times t and rank κ to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection method for varying σ	46
3.7	CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection (LRP) methods with the coarse-grid rank-reduction and the singular value based truncation on the level 8 spatial grid (i.e., $n_x = 257^2$).	47
3.8	CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the PGD method and the preconditioned low-rank projection methods on the level 8 spatial grid (i.e., $n_x = 257^2$).	47
3.9	CPU times to compute low-rank solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ using the preconditioned low-rank projection methods for varying ν . Numbers of GMRES cycles are shown in parentheses.	51

3.10	CPU times to compute approximate solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-6}$ using the preconditioned low-rank projection methods for varying ν . Numbers of GMRES cycles are shown in parentheses.	53
3.11	CPU times to compute low-rank solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection (LRP) methods with the coarse-grid rank-reduction and the singular value based truncation on the level 8 spatial grid (i.e., $n_x = 257^2$).	54
3.12	Largest values of θ_k or θ_k^* of eigenfunctions (3.29) in the KL expansion, required grid refinement level ℓ^c , half wavelength π/θ , and element size $h^c = 2^{-\ell^c}$ for different values of M	56
4.1	Tolerances and adaptive parameters.	86
5.1	Different choices for the LSPG weighting function.	112
5.2	Stability constant C in (5.32).	113
5.3	Stability constant C of Diffusion problem 1.	120

List of Figures

2.1	Block nonzero structure of the Galerkin matrix.	19
3.1	Mean solutions and contour plots on the level 6 spatial grid for varying ν	49
3.2	Mean solutions $\langle u(x, \xi) \rangle_\rho$ at $x = 1$ and $y = [0.9, 1]$ illustrating the exponential boundary layer for varying spatial grid refinement level, $\ell = \{4, 5, 8\}$, (top) and lengths in y -direction of first few elements from $y = 1$ (bottom).	57
3.3	Errors in the mean and the variance of the low-rank approximate solutions shown in (3.31) and (3.32) for the stochastic diffusion problem (a)-(d) and the stochastic convection-diffusion problem (e)-(f).	60
4.1	Spatial domain and finite element discretization.	84
4.2	Convergence of both exact and inexact nonlinear iterations (full-rank) and the low-rank inexact nonlinear iteration.	87
4.3	Mean and variances of full-rank velocity solutions $\vec{u}^x(x, \xi)$, $\vec{u}^y(x, \xi)$, and pressure solution $p(x, \xi)$ for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$	88
4.4	Difference in the means and variances of the full-rank and the low-rank solutions for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$	89
4.5	Estimated pdfs of the velocities \vec{u}^x , \vec{u}^y , and the pressure p at the point (3.6436, 0).	90
4.6	Norms of the gPC coefficients $\ \bar{u}_i\ _2$ for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$	90
4.7	Plots of coefficients of gPC components 2–7 of $\vec{u}^x(x, \xi)$ and coefficients v_i of $\theta_i(\xi)$ for $i = 2, \dots, 7$ for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$	92
4.8	A heat map of $(W^x)^T$	93
4.9	Eigenvalue decay of the AE and the SE covariance kernels.	95
4.10	Computational costs and ranks for varying correlation lengths with SE and AE covariance kernel.	95
4.11	Computational costs and ranks for varying correlation lengths and varying CoV with $\text{Re}_0 = 100$	96
4.12	Computational costs and ranks for varying correlation lengths and varying Re_0	98

5.1	Relative error measures versus polynomial degree for diffusion problem 1: lognormal random coefficient and deterministic forcing. Note that each LSPG method performs best in the error measure it minimizes.	118
5.2	Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 10 in increments of 1 going from left to right) for diffusion problem 1: lognormal random coefficient and deterministic forcing.	119
5.3	Relative errors versus polynomial degree for stochastic Galerkin (i.e., LSPG(A)/SG) for diffusion problem 2: lognormal random coefficient and random forcing. Note that monotonic convergence is observed only in the minimized error measure η_A	121
5.4	Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 2: lognormal random coefficient and random forcing	122
5.5	Relative error measures versus polynomial degree for a varying dimension n_o of the output matrix $F = F_1$ for diffusion problem 2: lognormal random coefficient and random forcing. Note that LSPG($F^T F$) has controlled errors only when $n_o = n_x$, in which case $\sigma_{\min}(F) > 0$	124
5.6	Plots of the error norm of output QoI for diffusion problem 2: lognormal random coefficient and random forcing when a linear functional is (a) $F(\xi) \equiv \sin(\xi) \times [0, 1]^{100 \times n_x}$ and (b) $F(\xi) = \xi \times [0, 1]^{100 \times n_x}$ for varying p and varying n_o	125
5.7	Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 3: Gamma random coefficient and random forcing. Note that each method is Pareto optimal in terms of minimizing its targeted error measure and computational wall time.	126
5.8	Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 10 in increments of 1 going from left to right) for stochastic convection-diffusion problem: lognormal random coefficient and deterministic forcing term.	127
5.9	Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 2: Lognormal random coefficient and random forcing. Analytic computations are used as much as possible to evaluate expectations.	130

List of Abbreviations

AE	Absolute difference exponential
AMG	Algebraic multigrid
CoV	Coefficient of variation
CG	Conjugate gradient
FEM	Finite element method
GCR	Generalized conjugate residual
GMRES	General minimum residual
gPC	Generalized polynomial chaos
IFISS	Incompressible Flow and Iterative Solver Software
KL	Karhunen-Loève
LRP	Low-rank projection
LSC	Least-squares commutator
LSPG	Least-squares Petrov–Galerkin
MINRES	Minimum residual
OoM	Out of memory
PDE	Partial differential equation
PGD	Proper generalized decomposition
PS	Pseudo-spectral
QoI	Quantity of interest
SE	Squared difference exponential
SG	Stochastic Galerkin
s.p.d.	Symmetric positive definite
SVD	Singular value decomposition
TD	Total degree
TP	Tensor product

Chapter 1: Introduction

Forward uncertainty propagation for parameterized algebraic systems is important in a range of applications to characterize the effects of input uncertainties on the output of computational models. Such parameterized algebraic systems arise in many important problems in science and engineering, for which models using stochastic partial differential equations (PDEs) are formulated and where uncertain input parameters are treated as a set of random variables. Examples of such problems include diffusion/ground water flow simulations with uncertain diffusivity/permeability [53,115], solid mechanics with uncertain material properties [43,44], incompressible fluid flow problems with uncertain viscosity [63,113], thermofluid flow problems [58,62], and reacting flow problems with chemical kinetics [29,88] with uncertain inputs. Parameterized algebraic systems also arise in other computational models such as models for reconstructing a high-resolution image from a set of low resolution images [22], and the PageRank algorithm [15,26].

There is a number of sources that cause input uncertainties, for example, an inherent stochastic nature of physical phenomena, and errors in measuring physical properties of objects of interest [46]. If the source of uncertainty comes from a lack of knowledge about physical properties, one feasible approach to handle this

is to collect a finite number of observations, characterize the statistical quantities of the properties, and model the properties as random fields governed by a set of random variables. That is, the physical property can be modeled as a random function such that the value of the random field varies over the spatial domain and the “stochastic domain” (i.e., the image space of the random variables). Suppose, for example, that we are interested in diffusion of chemicals in a medium with an unknown diffusivity. Then the diffusivity can be modeled as a random field based on the statistical quantities (e.g., sample mean and covariance) obtained from a finite number of observations.

There are several ways to model a random field. If the mean and the covariance function of a random field over the spatial domain are known, the random field can be represented as a Karhunen-Loève expansion [67], which is a linear expansion of orthogonal functions that depend on the spatial parameters and for which the coefficients of those functions are pairwise uncorrelated random variables. The orthogonal functions can be obtained by solving an eigenvalue problem associated with the covariance function. In a discrete sense, the KL-expansion is equivalent to principal component analysis [79]. To simulate a random field in terms of a finite number of random variables, the random field can be approximated by truncating the KL-expansion with a finite number of terms so that only the terms with larger variances are retained. There are also alternatives to using the KL-expansion; a random field can be modeled as a linear expansion of certain orthogonal polynomials, which will be introduced in Section 2.1, and as a linear expansion of trigonometric polynomials for weakly stationary random fields [49].

When the uncertain input is modeled as a random field, the model output (i.e., the solution of the algebraic system) also can be modeled as a random field, that is, a random function depending on the spatial location and the same random variables associated with the input random field. This can be seen from the fact that a specific realization of the input parameters gives rise to a deterministic problem and, consequently, leads to a specific realization of the solution function, the function evaluated at those specific values of the input parameters. Thus, the effects of the uncertain input on the model output (i.e., the solution) can be characterized by the statistical properties of the solution such as the mean, the variance, and higher moments of the solution.

The most straightforward approach to obtain statistical moments of the solution is to use the Monte Carlo method [72], which estimates the statistical moments of the solution from a finite number of sample solutions. That is, the Monte Carlo method requires a set of realizations of an input random field and collects solutions of deterministic problems associated with given input random field realizations. Then the statistical moments of the solution can be approximated by the sample moments. The Monte Carlo method is very simple and powerful; the method exhibits $\frac{1}{\sqrt{N}}$ convergence, where N is the number of samples, regardless of the dimension of the sample space. At the same time, however, if high accuracy is required in the approximation, N may need to be very large. Moreover, the Monte Carlo method can be very expensive if solving each deterministic problem associated with a sample is expensive. For faster convergence, there have been many improvements made such as Quasi-Monte Carlo methods [74] using pseudo-random sequences and sampling

methods based on Markov-Chain Monte Carlo methods [71]. Developing an optimal sampling strategy in the uncertainty quantification framework is an active area of research.

Recently, many advanced algorithms have been developed to achieve such statistical characterization of the solution with efficiency. One of the widely used approaches is spectral methods [46, 110], where the solution is expressed as a linear expansion of a finite number of certain orthogonal basis polynomials depending on the input random variables. Once the solution expansion is computed using numerical algorithms, the statistical quantities of the solution can be computed directly and inexpensively by sampling the solution expansion. This approach was inspired by the work [108], which studied the decomposition of a Gaussian random process (or, Gaussian random field), where the Gaussian random process is represented as a linear expansion of Hermite polynomials, which are orthogonal with respect to an inner product induced by the Gaussian probability density function. The series expansion displays a *mean-square convergence*; the expected value of a squared error goes to zero as the number of terms in the expansion goes to infinity.

In the early work on spectral methods [42, 44, 102] in uncertainty quantification, the Hermite polynomials are used to represent the solution function and numerical algorithms were developed to compute the coefficients of the solution expansion. This approach has been shown to be very successful when the random variables parameterizing the problems follow the Gaussian distribution; the Hermite polynomial expansion exhibits exponential convergence rate for the Gaussian random field [68]. For non-Gaussian random fields, however, the use of the Hermite

polynomials may result in significantly slower convergence [112]. With the recent development of the generalized polynomial chaos (gPC) expansion [112], the type of the orthogonal basis polynomials can be chosen based on the underlying measure of the input random variables, which results in better convergence. The effectiveness of the gPC expansion in characterizing the solution statistics of the stochastic PDEs has been demonstrated in [112–114].

After choosing the type of the orthogonal polynomials, spectral methods require a numerical algorithm to compute coefficients of the solution expansion. The first class of numerical algorithms developed for spectral methods is known as *stochastic Galerkin methods* [1, 3, 28, 46, 69], which extends a classical Galerkin approach for deterministic equations and, thus, is based on a Galerkin projection technique. As in the finite element method (FEM) for solving PDEs, the stochastic Galerkin method enforces a Galerkin orthogonality condition on the residual of stochastic PDEs with respect to the span of the gPC polynomial basis using an inner product associated with an underlying probability measure of the random variables. This procedure results in a system of (non-)linear equations where the number of equations is the same as the number of unknown coefficients and, thus, the coefficients can be obtained by solving the system of equations. The stochastic Galerkin method is popular for its simplicity (i.e., the trial and test bases are the same) and its optimality in terms of minimizing an energy norm of solution errors when the underlying PDE operator is elliptic and self-adjoint.

Another class of numerical algorithms can be thought of as specialized sampling-based methods. These methods generate a set of independent realizations of random

inputs based on their probability distribution and solve the corresponding deterministic realizations of the problems, and then use the resulting solutions to construct spectral approximations that can be used for simulation. There are several types of numerical algorithms of this type. One of the popular methods is the *stochastic collocation method* [2,111], which solves deterministic problems on a set of predetermined nodes in the space defined by the random variables. The stochastic collocation method computes the spectral approximation by constructing a Lagrange interpolating polynomial. A variant of the stochastic collocation is to compute the coefficients of the solution expansion using quadrature rules. In this approach, the solution is directly projected onto each polynomial basis function exploiting the orthogonality of the polynomial basis functions. This approach is known as a *pseudo-spectral* approach [109,110]. Another sampling-based approach for the spectral methods is a polynomial-regression-type approach with gPC expansion (e.g., least-squares regression [54,96,97], least angle regression [12], compressive sampling [32,50]).

Compared with sampling-based methods, the stochastic Galerkin method can lead to smaller errors for a fixed basis dimension [37,110,111]. In general, however, the stochastic Galerkin method suffers from two main problems. First, the method typically leads to a large set of coupled deterministic equations, for which computations will be expensive for large-scale applications. The solution function lies on a tensor product space of a spatial domain (a physical space) and a “stochastic domain” (a parameter space), and, after discretization, the number of coupled deterministic equations to be solved is the product of the numbers of basis polynomials in the spatial domain and the stochastic domain (i.e., degrees of freedom in each

domain). When the solution is sought in a high-dimensional space (i.e., dimension of the discrete physical space or the number of basis polynomials on the parameter space is large), the computation of the solution can be very expensive. Secondly, the method may not produce numerical solutions that minimize any measure of the solution error if the underlying PDE operator is not symmetric positive-definite. For many practical problems such as flow problems, PDE operators are not self-adjoint.

In an effort to alleviate the first difficulty, sparse structures of system matrices that can be obtained from the stochastic Galerkin method have been studied. To compute solutions of those systems, efficient iterative algorithms such as *Krylov subspace methods* [33, 34, 40, 56, 80, 81] and *multigrid methods* [27, 35, 64, 94] are applied. Matrix-vector products are essential matrix operations in those iterative solvers and those products can be performed very efficiently by exploiting the sparsity structures of system matrices. In combination with specially designed preconditioners [81, 84, 101, 105], those iterative solvers have been adjusted and successfully applied to many stochastic PDEs. As the size of the problems become larger, however, the computational costs of the iterative solvers increase rapidly, which makes use of those iterative solvers for high-dimensional problem less attractive.

The second issue of the stochastic Galerkin method has not been explored much. In many applications, however, quantities such as solution error or solution residual can be considered as more important metrics for measuring performance of solution methods. Numerical experiments in [73] demonstrated that, for certain classes of stochastic PDEs, the stochastic Galerkin method fails to generate a solution that minimizes a certain norm of solution error. Weighted projection methods

(i.e., *Petrov–Galerkin* projection techniques) have been proposed to resolve the issue. The weighted projection method successfully minimizes the solution error although the proposed methods require problem-specific projection bases.

In this thesis, we have developed efficient and optimal numerical methods to address and overcome the issues raised above. To address the first problem (high costs), we have developed efficient iterative solvers that decouple matrix operations associated with the spatial domain and the stochastic domain, which makes the computational complexity depend on the sum of the numbers of degrees of freedom in the spatial domain and the stochastic domain rather than their product. In particular, we consider a tensor variant of the Krylov subspace methods that operates in such a decoupled manner so that the computational costs and memory requirements can be significantly reduced. In addition, the variant of the Krylov subspace method will be used to compute a low-rank approximate solution, which further reduces the computational costs. The second problem is addressed using an optimal projection method, the *stochastic least-squares Petrov–Galerkin method*, which produces solution coefficients that minimize a certain measure of the solution error. We study the behavior of the stochastic Galerkin solution in several error measures and propose an optimization framework that provides an optimal projection basis to minimize a certain measure of the solution error.

1.1 Outline of Thesis

An outline of the thesis is as follows. We begin in Chapter 2 by introducing the stochastic Galerkin method and deriving the stochastic Galerkin system that arises from stochastic diffusion equations. Then we briefly review existing iterative solution methods for a large coupled deterministic system arising from the stochastic Galerkin method.

In Chapter 3, we discuss the use of a low-rank tensor variant of the Krylov subspace method in the stochastic Galerkin setting. For the efficient computation, we propose a two-level rank reduction scheme, which identifies an important subspace in the stochastic domain and compresses tensors of high rank on-the-fly during the iterations. The proposed reduction scheme is a coarse-grid method in that the important subspace can be identified inexpensively in a coarse spatial grid setting. The efficiency of the proposed method is illustrated by numerical experiments on benchmark elliptic linear stochastic PDE problems.

In Chapter 4, we develop a low-rank tensor variant of Newton–Krylov subspace methods for stochastic Navier–Stokes problems in the stochastic Galerkin setting. We base our development on a deterministic variant of a “linearization” scheme and solve a linear system at each nonlinear iteration step using the low-rank Krylov subspace method. We test our method under various settings of the Navier–Stokes equations and compare results with the conventional full-rank method.

In Chapter 5, we propose a new projection framework, stochastic Least-Squares Petrov–Galerkin (LSPG) method, which provides an optimal projection method.

The proposed method is optimal in the sense that it produces the solution that minimizes a weighted ℓ^2 -norm of the residual over all solutions in a given finite-dimensional subspace. With extensive numerical experiments, we show that the weighted LSPG methods outperforms other spectral methods in minimizing corresponding target weighted norms.

In Chapter 6, we draw some conclusions.

Chapter 2: Background: The stochastic Galerkin method

In this chapter, we begin with a brief introduction of the stochastic Galerkin method with stochastic diffusion equations as a model problem. The stochastic Galerkin discretization procedure is discussed only with the stochastic diffusion problem, an extension of the stochastic Galerkin formulation to other linear elliptic PDEs with uncertain input is straightforward.

2.1 Overview of the stochastic Galerkin method

Consider the steady-state stochastic diffusion equation with homogeneous Dirichlet boundary conditions,

$$\begin{cases} -\nabla \cdot (a(x, \omega) \nabla u(x, \omega)) = f(x, \omega) & \text{in } D \times \Omega, \\ u(x, \omega) = 0 & \text{on } \partial D \times \Omega, \end{cases} \quad (2.1)$$

where the diffusion coefficient $a(x, \omega)$ is a random field and ω is an elementary event in a probability space (Ω, \mathcal{F}, P) . Here, Ω is a sample space, \mathcal{F} and P are a σ -algebra on Ω and a probability measure on Ω , respectively. The gradient operator ∇ only acts on the physical domain D . We begin by introducing a weak formulation of a deterministic problem of (2.1), which arises from sampling an elementary event $\omega^{(k)}$

from the probability space Ω : Find $u(x, \omega^{(k)}) \in H_0^1(D)$ such that

$$\int_D a(x, \omega^{(k)}) \nabla u(x, \omega^{(k)}) \cdot \nabla v(x) dx = \int_D f v(x), \quad \forall v(x) \in H_0^1(D). \quad (2.2)$$

The stochastic Galerkin method seeks a solution satisfying an “extended” weak formulation of (2.1): Find $u(x, \xi)$ in $V = H_0^1(D) \otimes L^2(\Omega)$ such that

$$\left\langle \int_D a(x, \omega) \nabla u(x, \omega) \cdot \nabla v(x, \omega) dx \right\rangle = \left\langle \int_D f v(x, \omega) \right\rangle, \quad \forall v(x, \omega) \in V \quad (2.3)$$

where $\langle \cdot \rangle$ refers to expected value with respect to the probability measure on $L_2(\Omega)$ and V is equipped with the gradient norm

$$\|v\|_V^2 = \int_{\Omega} \int_D a(x, \omega) |\nabla v(x, \omega)|^2 dx dP(\omega). \quad (2.4)$$

If $a(x, \omega)$ is bounded and uniformly positive,

$$0 < a_{\min} \leq a(x, \omega) \leq a_{\max} < +\infty, \quad \text{a.e. in } D \times \Omega, \quad (2.5)$$

then the Lax-Milgram lemma can be applied to establish existence and uniqueness of a solution $u(x, \omega) \in V$ of the variational problem (2.3). The gradient norm is also called an energy norm [3]. It has been shown that the solution error in the energy norm is minimized by the stochastic Galerkin solution [3, 73] as in this example, the underlying PDE operator is self-adjoint and coercive.

For the uncertain diffusivity $a(x, \omega)$, we consider a spectral representation of

the random field using gPC expansion,

$$a(x, \xi(\omega)) = \sum_{i=0}^{\infty} a_i(x) \psi_i(\xi(\omega)), \quad (2.6)$$

where $\xi(\omega) = \{\xi_1(\omega), \dots, \xi_M(\omega)\}$ is an M -dimensional random variable with joint probability density function $\rho(\xi)$. We assume that the random variables are independent and identically distributed and the stochastic domain is denoted by $\Gamma = \prod_{i=1}^M \Gamma_i$ (i.e., the joint image of ξ) where $\xi_i : \Omega \rightarrow \Gamma_i$. Here, $\{\psi_i(\xi)\}$ is an orthogonal gPC basis, for which the details will be introduced in Section 2.2. In the sequel, we denote the random diffusivity by $a(x, \xi)$ as the random diffusivity is parameterized with a set of random variables ξ .

For simplifying a derivation of the stochastic Galerkin system, we consider a special case of the random field expansion (2.6) where the expansion consists of polynomials with degree ≤ 1 . Such random field can be simulated by using Karhunen-Loève expansion [67] or considering a piecewise constant random field. Note that the derivation of the stochastic Galerkin system with a general random field expansion (2.6) is a straightforward extension of the derivation described in the following. For the discussion in this chapter, we consider a truncated Karhunen-Loève expansion [67],

$$a(x, \omega) = a_0 + \sigma \sum_{i=1}^M \sqrt{\lambda_i} a_i(x) \xi_i(\omega), \quad (2.7)$$

where a_0 and σ^2 are the mean and variance of the random field, respectively, and

(λ_i, a_i) is an eigenpair of the covariance kernel of the random field, $C(x, y)$. That is, eigenpairs consist of the solutions of the eigenproblem of the integral operator $C: L^2(D) \rightarrow L^2(D)$,

$$(Cu)(x) = \int_D c(x, y)u(y)dy, \quad (Ca_m)(x) = \lambda_m a_m(x),$$

If the random field $a(x, \xi)$ is parameterized by a finite number of random variables, then the solution $u(x, \omega)$ can be described by this same set of random variables by Doob-Dynkin's Lemma [86] (i.e., $u(x, \omega) \approx u(x, \xi_1, \dots, \xi_M)$).

2.2 Discretization

The discrete stochastic Galerkin method employs a standard approximation in the spatial domain and a polynomial approximation in the probability domain [1, 3, 46]. The stochastic Galerkin method seeks a finite-dimensional solution $u_{hp}(x, \xi) \in W^h = X_h \otimes S_M$ such that

$$\left\langle \int_D a(x, \xi) \nabla u_{hp}(x, \xi) \cdot \nabla v(x, \xi) dx \right\rangle_\rho = \left\langle \int_D f v(x, \xi) \right\rangle_\rho \quad v(x, \xi) \in W^h \quad (2.8)$$

where X_h and S_M are finite-dimensional subspaces of $H_0^1(D)$ and $L_\rho^2(\Gamma)$,

$$X_h = \text{span}\{\phi_r(x)\}_{r=1}^{n_x} \subset H_0^1(D), \quad (2.9)$$

$$S_M = \text{span}\{\psi_s(\xi)\}_{s=1}^{n_\xi} \subset L_\rho^2(\Gamma), \quad (2.10)$$

and

$$u_{hp}(x, \xi) = \sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{r,s} \phi_r(x) \psi_s(\xi). \quad (2.11)$$

Here, $\{\phi_r\}$ is a set of standard finite element basis functions and $\{\psi_s\}$ is a set of basis functions for the generalized polynomial chaos (gPC) expansion [112] consisting of products of orthonormal univariate polynomials: $\psi_s(\xi) = \psi_{\alpha(s)}(\xi) = \prod_{i=1}^M \pi_{\alpha_i(s)}(\xi_i)$ where $\{\pi_{\alpha_i(s)}(\xi_i)\}_{i=1}^M$ is a set of univariate polynomials and $\alpha(s) = (\alpha_1(s), \dots, \alpha_M(s)) \in \mathbb{N}_0^M$ is a multi-index, where α_i represents the degree of a polynomial in ξ_i . The univariate polynomials $\{\pi_{\alpha_i(s)}(\xi_i)\}_{i=1}^M$ are orthonormal with respect to underlying probability density functions $\{\rho_i(\xi_i)\}_{i=1}^M$,

$$\int_{\Gamma_i} \pi_k(\xi_i) \pi_l(\xi_i) \rho(\xi_i) d\xi_i = \kappa_i \delta_{kl}, \quad k, l \in \mathbb{N}_0, i = 1, \dots, M$$

where $\delta_{kl} = 1$ if $k = l$ and 0 otherwise. Due to the orthonormality of the univariate polynomials $\{\pi_{\alpha_i(s)}(\xi_i)\}_{i=1}^M$ and the independence among the random variables, the stochastic basis functions $\{\psi_s\}$ are orthonormal with respect to the joint probability density function $\rho(\xi) = \prod_{i=1}^M \rho_i(\xi_i)$,

$$\int_{\Gamma} \psi_k(\xi) \psi_l(\xi) \rho(\xi) d\xi = \prod_{i=1}^M \int_{\Gamma_i} \pi_{\alpha_i(k)}(\xi_i) \pi_{\alpha_i(l)}(\xi_i) \rho(\xi_i) d\xi_i = \prod_{i=1}^M \delta_{\alpha_i(k) \alpha_i(l)}.$$

If ρ is the density function corresponding to M -variate uniform distribution, ψ_s is a product of M univariate Legendre polynomials. Table 2.1 lists different types of probability measures (and probability density functions) and the types of gPC basis polynomials associated with those probability measures.

Table 2.1: Probability distribution and the type of gPC basis.

Probability distribution	pdf	gPC basis polynomial	Support
Normal	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{\xi^2}{2})$	Hermite	$[-\infty, \infty]$
Uniform	$\frac{1}{2}$	Legendre	$[0, 1]$
Exponential	$\exp(-\xi)$	Laguerre	$[0, \infty]$
Gamma(α)	$\frac{\xi^\alpha \exp(-\xi)}{\Gamma(\alpha+1)}$	Generalized Laguerre	$[0, \infty]$
Beta(α, β)	$\frac{(1-\xi)^\alpha (1-\xi)^\beta}{2^{\alpha+\beta+1} B(\alpha+1, \beta)}$	Jacobi	$[0, \infty]$

Once the type of the gPC basis polynomials is chosen, the finite-dimensional polynomial space, $S_M = \text{span}\{\psi_s(\xi)\}_{s=1}^{n_\xi}$, can be constructed. The most naive approach in constructing S_M is called ‘‘Tensor Product (TP) space,’’ for which the multi-index set can be defined as

$$\Lambda_{M,p}^{\text{TP}} = \{\alpha(s) \in \mathbb{N}_0^M : \max\{\alpha_1(s), \dots, \alpha_M(s)\} \leq p\}. \quad (2.12)$$

Although the TP space is easy to construct, the cardinality of the set $\Lambda_{M,p}^{\text{TP}}$ is M^p , which increases exponentially as the maximum polynomial degree p increases. Instead, in this study, we set Λ_M to be the Total Degree (TD) space $\Lambda_{M,p}^{\text{TD}}$, given by

$$\Lambda_{M,p}^{\text{TD}} = \{\alpha(s) \in \mathbb{N}_0^M : \|\alpha(s)\|_1 \leq p\} \quad (2.13)$$

where \mathbb{N}_0^M is the set of non-negative integers, $\|\alpha(s)\|_1 = \sum_{k=1}^M \alpha_k(s)$, and p defines the maximal degree of $\{\psi_i\}_{i=1}^{n_\xi}$. Then, the number of gPC basis functions is $n_\xi = \dim(\Lambda_{M,p}) = \frac{(M+p)!}{M!p!}$. The TD space has been known to be very effective in approximating the solutions of many stochastic PDEs. In particular, if the stochastic diffusion equations with the random field of the form (2.7) is considered, the TD

space is known to provide the best N-term approximation for the solutions [23, 24]. In [7, 8], the decay of Legendre coefficients for the solutions of the elliptic stochastic PDEs is studied and the TD space has been shown to be quasi-optimal. Thus, in this thesis, we use the TD space as the finite-dimensional approximation space in the stochastic domain. An example of the TD space with $M = 2$ and $p = 3$ is

$$\Lambda_{2,3}^{\text{TD}} = \{(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (3, 0)\},$$

which is lexicographically ordered, and the cardinality of the space is $n_\xi = \frac{(2+3)!}{2!3!} = 10$.

If the coefficients of (2.11) are ordered by grouping spatial indices together as

$$u_{11}, u_{21}, \dots, u_{n_x 1}, u_{12}, u_{22}, \dots, u_{n_x 2}, u_{13}, \dots, u_{n_x n_\xi}, \quad (2.14)$$

then, it follows from (2.8) and (2.11) that the Galerkin system

$$Au = f \quad (2.15)$$

can be represented using Kronecker-product notation [81],

$$\left(G_0 \otimes K_0 + \sum_{l=1}^M G_l \otimes K_l \right) u = g_0 \otimes f_0, \quad (2.16)$$

where the Kronecker product between two matrices $G \in \mathbb{R}^{n_\xi \times n_\xi}$ and $K \in \mathbb{R}^{n_x \times n_x}$ is

defined as follows:

$$G \otimes K = \begin{bmatrix} [G]_{11}K & \dots & [G]_{1n_\xi}K \\ \vdots & & \vdots \\ [G]_{n_\xi 1}K & \dots & [G]_{n_\xi n_\xi}K \end{bmatrix},$$

K_i refers to the i th weighted stiffness matrix defined via

$$\begin{aligned} [K_0]_{ij} &= \int_D a_0 \nabla \phi_i(x) \nabla \phi_j(x) dx, \\ [K_l]_{ij} &= \sigma \sqrt{\lambda_l} \int_D a_l(x) \nabla \phi_i(x) \nabla \phi_j(x) dx, \quad l = 1, \dots, M, \end{aligned} \tag{2.17}$$

G_i refers to the i th “stochastic” matrices defined via

$$\begin{aligned} [G_0]_{ij} &= \langle \psi_i(\xi) \psi_j(\xi) \rangle_\rho, \\ [G_l]_{ij} &= \langle \xi_l \psi_i(\xi) \psi_j(\xi) \rangle_\rho \quad l = 1, \dots, M, \end{aligned} \tag{2.18}$$

and the vectors f_0 and g_0 are defined via

$$\begin{aligned} [f_0]_i &= \int_D f \phi_i(x) dx, \\ [g_0]_i &= \langle \psi_i(\xi) \rangle_\rho. \end{aligned} \tag{2.19}$$

Note that $\{G_l\}_{l=1}^M$ of (2.18) are highly sparse because of the orthogonality properties of the stochastic basis functions [41].

The global Galerkin system shown in (2.16) is of order $n_x n_\xi$, which becomes very large if the solution is sought on a fine spatial grid (i.e., large n_x) and a high-dimensional stochastic space (i.e., large M or p and, consequently, large n_ξ). The

Kronecker-product structure, however, leads to a block-sparse matrix, where the block nonzero structure of the matrix follows the nonzero structure of the stochastic matrices $\{G_i\}_{i=0}^M$. Figure 2.1 depicts the block nonzero structure of the Galerkin matrix of order $n_x n_\xi$ where $n_\xi = 56$ by setting $M = 5$ and $p = 3$ (i.e., $56 = \frac{(5+3)!}{5!3!}$), and each square in the figure represent a weighted stiffness matrix of order n_x . With this block sparse structure, it is natural to consider development of sparse linear solvers for use with the stochastic Galerkin methods, which will be discussed in the next section.

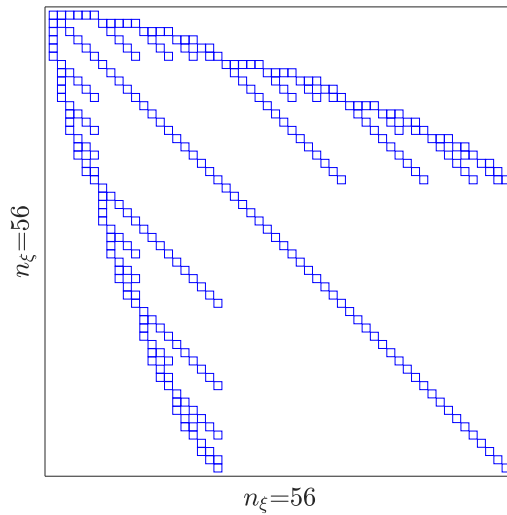


Figure 2.1: Block nonzero structure of the Galerkin matrix.

2.3 Iterative solvers for stochastic Galerkin systems

As for deterministic PDE problems, use of *Krylov subspace* methods has been very successful for stochastic PDE problems. Here, we review briefly review Krylov subspace methods and some notable results concerning Krylov subspace methods

in the context of the stochastic Galerkin method. A Krylov subspace method seeks an approximate solution of a linear system $Ax = b$ on an affine subspace $x_0 + \mathcal{K}_m$, where \mathcal{K}_m is the m -dimensional Krylov subspace

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}.$$

Here, x_0 denotes a possibly arbitrary initial iterate for an approximate solution and $r_0 = b - Ax_0$ denotes the initial residual. An approximate solution x_m can be found by employing an orthogonal projection of the residual $r_m = b - Ax_m$ onto another m -dimensional subspace \mathcal{L}_m ($r_m \perp \mathcal{L}_m$). There are two well-known projection techniques: the Galerkin projection technique with $\mathcal{L}_m = \mathcal{K}_m$ and the Petrov–Galerkin projection technique $\mathcal{L}_m = A\mathcal{K}_m$. Such projection techniques give rise to effective Krylov subspace methods. The Galerkin projection technique characterizes the Conjugate Gradient (CG) method [51] for a symmetric positive definite A . The Petrov–Galerkin projection technique characterizes the minimum residual method (MINRES) [78] for a nonsingular symmetric indefinite A and the generalized minimum residual method (GMRES) [92] for a general nonsingular A .

An initial attempt to solve the global Galerkin system (2.16) using Krylov subspace solution methods can be found in [80], followed by more advanced studies on iterative methods for the Galerkin system [33,34,56]. In those studies, an efficient structure-aware matrix-vector product exploiting the block sparse structure has been studied. For faster convergence of the iterative methods, a preconditioned system

is considered,

$$M^{-1}A = M^{-1}f \tag{2.20}$$

where M is the preconditioner. Or, alternatively,

$$AM^{-1}\tilde{u} = f, \quad u = M^{-1}\tilde{u}. \tag{2.21}$$

A Krylov subspace method constructs an m -dimensional preconditioned Krylov subspace $\text{span}\{r_0, M^{-1}Ar_0, \dots, M^{-1}A^{m-1}r_0\}$ for a left preconditioned system (2.20) and $\text{span}\{r_0, AM^{-1}r_0, \dots, A^{m-1}M^{-1}r_0\}$ for a right preconditioned system (2.21). The use of this preconditioner M in Krylov subspace methods requires an application of the action of its inverse M^{-1} , or approximating it.

In those early studies, a preconditioned conjugate gradient method [51] with a simple block-diagonal preconditioning strategy, which incorporates the mean component of the random field is widely used, i.e.,

$$M = G_0 \otimes K_0. \tag{2.22}$$

For the efficient application of the preconditioner, an incomplete Cholesky factor as a preconditioner has been considered [45, 80].

In more recent work [81], a preconditioned CG method with a black-box Algebraic Multigrid (AMG) [90] preconditioner was considered, where the action of K_0^{-1} is replaced by the V-cycle of the black-box AMG. In [81], it has been shown that an eigenvalue bound of the preconditioned Galerkin system associated with the normal

and the uniform random variables is independent of the spatial discretization parameter and p for bounded random variables such as ones with uniform distribution, but it depends on the variance of the random field σ^2 in (2.7). Also, it has been shown that the black-box AMG preconditioner is robust with respect to the spatial discretization parameter and requires less memory than the factorization method for fine spatial meshes.

So far, the preconditioning strategy only takes into account the matrix associated with the mean coefficient of the random field (i.e., $G_0 \otimes K_0$). The mean-based preconditioner may not be effective if the variance of the random field becomes large compared to the mean. To resolve this issue, a preconditioner proposed in [105] incorporates the entire information in the global Galerkin matrix (2.16). Inspired by the work of [66], the new preconditioner is constructed by solving a minimization problem

$$\min \|A - G \otimes K_0\|_F \quad (2.23)$$

where $\|\cdot\|_F$ is a Frobenius norm, A is the global Galerkin matrix, K_0 is the mean stiffness matrix, and $G \in \mathbb{R}^{n_\xi \times n_\xi}$ to be solved. The minimization can be solved as

$$G = \sum_{i=0}^M \frac{\text{trace}(K_i^T K_0)}{\text{trace}(K_0^T K_0)} G_i, \quad (2.24)$$

where

$$\text{trace}(A) = \sum_{i=1}^{n_x n_\xi} [A]_{ii}$$

is the trace operator, which sums of diagonal entries of $A \in \mathbb{R}^{n_x n_\xi \times n_x n_\xi}$. Combined

with this preconditioner, the conjugate gradient method performs efficiently even for the large variance stochastic diffusion coefficients.

There are other effective preconditioning strategies [36, 89, 100, 101] including a preconditioner based on matrix splitting technique (e.g., Jacobi, Gauss-Seidel) applied to the stochastic matrices [89], and a hierarchical Gauss-Seidel preconditioner and a Schur complement preconditioner and its efficient variant proposed by [100, 101], which exploit recursive hierarchical structure of the global Galerkin matrix. We also note that there are several attempts to solve the stochastic Galerkin system using the multigrid solver. Initiated by [64], the practical application and the theoretical aspects of the multigrid method for the stochastic diffusion equations have been studied in [35, 94], and further extended in [27, 89].

Another successful approach shown in [40] considers a stochastic variant of the mixed variational formulation [16, 87] to discretize the stochastic diffusion equations, which results in a symmetric and indefinite system matrix. To solve the saddle point problem, a preconditioned MINRES method [78] is considered. For a preconditioner, a mean-based Schur complement of the indefinite system is computed, where action of inverse required to apply the preconditioner is replaced with the application of AMG V-cycle. Further studies on a preconditioner for the saddle point system in the stochastic Galerkin mixed variational formulation have been conducted in [84].

When the lognormal diffusion coefficient $a(x, \xi) = \exp(g(x, \xi))$, where $g(x, \xi)$ is a Gaussian random field, is considered, the coefficient $a(x, \xi)$ is typically approximated as a finite-term gPC expansion. After discretization, the resulting systems are block dense [69, 106], which makes matrix operations required by Krylov subspace

methods expensive. To resolve this issue, in [106], a “log-transformed” reformulation [115] of the problem as a convection-diffusion problem is considered. By multiplying $\exp(-g(x, \xi))$ on both sides of $-\nabla \cdot (a(x, \xi) \nabla u(x, \xi)) = f$ and algebraically manipulating the equation, one can obtain a convection-diffusion equation

$$\Delta u(x, \xi) - \nabla g(x, \xi) \cdot \nabla u(x, \xi) = f(x) \exp(-g(x, \xi)). \quad (2.25)$$

The stochastic Galerkin discretization can be applied to the convection-diffusion equations, which results in a nonsymmetric system of equations. To compute solutions of the nonsymmetric system, the generalized minimum residual method [92] is used. For preconditioning, two types of a mean-based preconditioner have been used: one constructed from a diffusion term only, and one constructed from the diffusion term and a convection term associated with the mean coefficients of the random field. A mixed variational formulation of the log-transformed equations and an associated iterative solution method are studied in [107].

Those solution methods have explored various formulations of the stochastic diffusion equations and applied iterative solvers to the resulting Galerkin system, which is preconditioned by various preconditioning strategies. With the numerical experiments with benchmark problems, those methods have been shown to be efficient and effective for moderate dimensional problems. However, the computational complexity $O(n_x n_\xi)$ grows rapidly as the problem posed on a finer spatial grid or the number of random variables parameterizing the problem increases.

Chapter 3: Low-rank approximation method for linear PDEs

3.1 Introduction

In this chapter, we present a low-rank approximation method for the stochastic linear elliptic boundary value problem: Find a random function, $u(x, \xi) : \bar{D} \times \Gamma \rightarrow \mathbb{R}$ that satisfies

$$\mathcal{L}(a(x, \xi))(u(x, \xi)) = f(x) \quad \text{in } D \times \Gamma, \quad (3.1)$$

where \mathcal{L} is a linear elliptic operator and $a(x, \xi)$ is a positive random field parameterized by a set of random variables $\xi = \{\xi_1, \dots, \xi_M\}$. The problem is posed on a bounded domain $D \subset \mathbb{R}^2$ with appropriate boundary conditions.

After the stochastic Galerkin method [1, 3, 46], which, after discretization described in Section 2.2, leads to a large coupled deterministic system (2.16) for which computations will be expensive for large-scale applications. When the coefficient $a(x, \xi)$ has an affine structure depending on a finite number of random variables, the system matrix A can be represented by a sum of Kronecker products of smaller matrices. Matrix operations such as matrix-vector products that take advantage of the tensor format can be performed efficiently, which makes the use of iterative solvers attractive. In this study, we develop a new efficient iterative solver for

systems represented in the Kronecker-product structure.

In recent years, many authors started to explore the Kronecker-product structure of such problems and developed iterative algorithms that exploit the structure to reduce computational efforts [6, 59–61, 70, 83, 93]. In particular, thorough use of tensor Krylov subspace methods, which operate in tensor format, have been studied. Variants of this approach have been developed for the Richardson iteration [61, 70], the conjugate gradient method [61], the BiCGstab method [61], the minimum residual method [103], and the general minimum residual (GMRES) method [6]. In addition, it has been shown that the solution of (3.1) in the stochastic Galerkin setting can be approximated by a tensor of low rank, which further reduces computational effort [4, 5]. If Krylov subspace methods are used to compute such a solution, however, it may happen that approximate solutions or other auxiliary terms obtained during the course of an iteration do not have low rank, and rank-reduction schemes are required to keep costs under control.

In this study, we will explore a variant of the generalized minimum residual (GMRES) method combined with a rank-reduction strategy that exploits specific features of the stochastic Galerkin formulation. The strategy we propose is a two-level scheme that first identifies a low-dimensional subspace, obtained from a coarse-grid spatial discretization, on which a low-rank coarse-grid tensor solution is computed. This solution can be used to estimate the rank of the tensor solution for the desired fine-grid solution. This information is used to define a strategy for rank reduction to be used with iteration on the fine grid space. We show that this strategy enhances the efficiency of preconditioned GMRES for computing the

solution.

The proposed method can be viewed as a dimension-reduction method as it identifies a dominant subspace and computes an approximate solution in that subspace. Other approaches developed for dimension reduction for the solutions of stochastic PDEs include reduced basis methods [52, 85], which construct dominant subspace associated with parameterized models using greedy search methods, and active subspace methods [25], which detect a subspace of strong variability for a scalar-valued multivariate functions using gradient computations. Another model reduction approach developed in [31] identifies a dominant subspace based on the covariance structure of the solution on the coarse grid and uses the subspace for the fine-grid computation. The approach developed here uses inexpensive low-rank approximation technique to construct the desired subspace on coarse-grid computations. Then the identified subspace is used to truncate tensors of high ranks in the iteration process to construct a solution on a finer spatial discretization.

An outline of the chapter is as follows. In section 3.2, we review the stochastic Galerkin method and present the Kronecker-product structure of Galerkin systems. In section 3.3, we present a preconditioned projection method for computing approximate solutions in low-rank tensor format. In section 3.4, we review the conventional approaches and propose a coarse-grid rank-reduction scheme, which is the main contribution of this work. In section 3.5, we illustrate the effectiveness of the low-rank projection method combined with the proposed truncation scheme by numerical experiments on benchmark problems. In section 3.6, we discuss the impact of truncation on solution statistics. Finally, in section 3.7, we draw some

conclusions.

3.2 Stochastic Galerkin formulation in tensor notation

Recall the stochastic Galerkin discretization discussed in Chapter 2, where we consider the steady-state stochastic diffusion equation with homogeneous Dirichlet boundary conditions shown in (2.1) with the diffusion coefficient $a(x, \xi)$ parameterized by using a truncated Karhunen-Loève expansion (2.7). Here, ξ is an M -dimensional random variable with joint probability density function $\rho(\xi)$. We let $\Gamma = \prod_{i=1}^M \Gamma_i$ denote the joint image of ξ , which we refer to as the *stochastic domain*. The expected value of a random variable $v(\xi)$ on Γ is then $\langle v(\xi) \rangle_\rho = \int_\Gamma v(\xi) \rho(\xi) d\xi$.

The stochastic Galerkin method [1, 3, 46] seeks a finite-dimensional solution $u_{hp}(x, \xi) = \sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{rs} \phi_r(x) \psi_s(\xi)$ as shown in (2.11). We consider set the Total Degree (TD) space $\Lambda_{M,p}^{\text{TD}}$: $\Lambda_{M,p}^{\text{TD}} = \{\alpha(s) \in \mathbb{N}_0^M : \|\alpha(s)\|_1 \leq p\}$ (2.13). Consequently, the number of gPC basis functions is $n_\xi = \dim(\Lambda_{M,p}) = \frac{(M+p)!}{M!p!}$. Ordering the coefficients of (2.11) based on lexicographical order as shown in (2.14) gives the linear system $Au = f$ of (2.16) represented in tensor product notation [81],

$$\left(G_0 \otimes K_0 + \sum_{l=1}^M G_l \otimes K_l \right) u = g_0 \otimes f_0 \quad (3.2)$$

where $\{K_l\}_{l=0}^M$, $\{G_l\}_{l=0}^M$, f_0 , and g_0 are defined in (2.17)–(2.19).

We will make use of an isomorphism between $\mathbb{R}^{n_x n_\xi}$ and $\mathbb{R}^{n_x \times n_\xi}$ determined

by the operators $\text{vec}(\cdot)$ and $\text{mat}(\cdot)$: $u = \text{vec}(U)$, $U = \text{mat}(u)$ where

$$u = [u_1^T, \dots, u_{n_\xi}^T]^T \in \mathbb{R}^{n_x n_\xi} \quad (3.3)$$

$$U = [u_1, \dots, u_{n_\xi}] \in \mathbb{R}^{n_x \times n_\xi} \quad (3.4)$$

with each u_i of length n_x . In particular, (3.2) is equivalent to its “matricized” form

$$\sum_{l=0}^M K_l U G_l^T = f_0 g_0^T, \quad (3.5)$$

and u and U can be used interchangeably to represent a solution of the Galerkin system. A solution u can be represented by a sum of vectors of Kronecker structure, or equivalently $U = \text{mat}(u)$ can be represented by a sum of rank-one matrices,

$$u = \sum_{k=1}^{\kappa_u} z_k \otimes y_k \quad (3.6)$$

$$\Leftrightarrow U = \sum_{k=1}^{\kappa_u} y_k z_k^T = Y_{\kappa_u} Z_{\kappa_u}^T \quad (3.7)$$

where $y_i \in \mathbb{R}^{n_x}$, $z_i \in \mathbb{R}^{n_\xi}$, and $Y_{\kappa_u} = [y_1, \dots, y_{\kappa_u}] \in \mathbb{R}^{n_x \times \kappa_u}$ and $Z_{\kappa_u} = [z_1, \dots, z_{\kappa_u}] \in \mathbb{R}^{n_\xi \times \kappa_u}$. A tensor of the form (3.6) is often referred to as having a *canonical decomposition* [21] (e.g., $x = \sum_{i=1}^{\kappa_x} \otimes_{j=1}^d x_i^j$ where $x \in \mathbb{R}^{n_1 \dots n_d}$, $x_i^j \in \mathbb{R}^{n_j}$ for $i = 1, \dots, \kappa_x$, $j = 1, \dots, d$, and d refers to the dimension of the tensor). The tensor rank κ_u is defined as the smallest number of terms needed to represent u . In this study, the dimension of the tensor u is two and the tensor rank κ_u of the tensor u coincides with the rank of the matrix U . Thus, in the sequel, we also use κ_u to refer to the

rank of u . With this notation, the stochastic Galerkin solution $u_{hp}(x, \xi)$ can be represented as

$$u_{hp}(x, \xi) = \Phi(x)^T Y_{\kappa_u} Z_{\kappa_u}^T \Psi(\xi) = (Y_{\kappa_u}^T \Phi(x))^T (Z_{\kappa_u}^T \Psi(\xi)) \quad (3.8)$$

where $\Phi : D \rightarrow \mathbb{R}^{n_x}$ is given by $\Phi(x) = [\phi_1(x), \dots, \phi_{n_x}(x)]^T$ and $\Psi : \Gamma \rightarrow \mathbb{R}^{n_\xi}$ is given by $\Psi(\xi) = [\psi_1(\xi), \dots, \psi_{n_\xi}(\xi)]^T$. As shown in [104], (3.8) corresponds to a separated representation [11],

$$u_{hp}(x, \xi) = \sum_{i=1}^{\kappa_u} \hat{y}_i(x) \hat{z}_i(\xi), \quad (3.9)$$

where $\hat{y}_i(x) = (\Phi(x))^T y_i$ and $\hat{z}_i(\xi) = (\Psi(\xi))^T z_i$. We will use this representation to construct a new rank-reduction operator. In the discrete model (3.8), the rank of the solution is typically $\kappa_u = \min(n_x, n_\xi)$.

In [10, 48], it was shown that the solution to (3.2) can be approximated well by a quantity \tilde{u} of rank $\kappa_{\tilde{u}} \ll \min(n_x, n_\xi)$ if the system matrix and the right-hand side has Kronecker-product structure. Thus, we seek a low-rank approximation to the solution \tilde{u} to (3.2) for which

$$A\tilde{u} = \left(\sum_{l=0}^M G_l \otimes K_l \right) \left(\sum_{k=1}^{\kappa_{\tilde{u}}} \tilde{z}_k \otimes \tilde{y}_k \right) \approx g_0 \otimes f_0. \quad (3.10)$$

3.2.1 Basic operations in tensor notation

We point out here a feature of the basic operations required by Krylov subspace methods in the setting we are considering, where the operators and data of interest have tensor format. The m th step of such methods results in the *Krylov subspace*, $\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}$, which is generated using matrix-vector products and addition/subtraction of vectors.

The matrix-vector product in (3.10) can be represented as a sum of rank-one tensors by exploiting the properties of the Kronecker product,

$$Au = \sum_{l=0}^M \sum_{k=1}^{\kappa_u} G_l z_k \otimes K_l y_k = \sum_{i=1}^{(M+1)\kappa_u} \hat{z}_i \otimes \hat{y}_i. \quad (3.11)$$

The latter expression in (3.11) suggests that in tensor notation, the matrix-vector product typically results in a vector with a higher rank. Similarly, the addition of two vectors u and v of rank κ_u and κ_v in tensor notation gives

$$u + v = \sum_{i=1}^{\kappa_u} z_i \otimes y_i + \sum_{j=1}^{\kappa_v} \hat{z}_j \otimes \hat{y}_j = \sum_{i=1}^{\kappa_u + \kappa_v} z_i \otimes y_i, \quad (3.12)$$

where $y_{i+\kappa_u} = \hat{y}_i$ and $z_{i+\kappa_u} = \hat{z}_i$, $i = 1, \dots, \kappa_v$, so that the resulting sum may have rank as large as $\kappa_u + \kappa_v$. Thus, although the goal is to find an approximate solution to (3.2) of low rank, two of the fundamental operations used in Krylov subspace methods tend to increase the rank of the quantities produced. Following [6], we will address this point in the next section.

3.3 A preconditioned projection method in tensor format

As is well known, the generalized minimum residual method (GMRES) [92] constructs an approximate solution $u_m \in u_0 + \mathcal{K}_m(A, v_1)$ where u_0 is an initial vector with residual $r_0 = f - Au_0$, $v_1 = r_0 / \|r_0\|_2$, and \mathcal{K}_m is the Krylov space. This is done by generating $V_m = [v_1, \dots, v_m]$, where $\{v_j\}_{j=1}^m$ is an orthogonal basis for \mathcal{K}_m , and then computing u_m whose residual r_m is orthogonal to $W_m = AV_m$. The method is shown in Algorithm 1. In this section, we discuss a variant of this method based on low-rank projection, where advantage is taken of the Kronecker format of the matrix A and the fact that we seek an approximation of u with low-rank structure.

Algorithm 1 GMRES method without restarting [91]

```

1: set the initial solution  $u_0$ 
2:  $r_0 := f - Au_0$ 
3:  $\tilde{v}_1 := r_0$ 
4:  $v_1 := \tilde{v}_1 / \|\tilde{v}_1\|$ 
5: for  $j = 1, \dots, m$  do
6:    $w_j := Av_j$ 
7:   solve  $(V_j^T V_j)\alpha = V_j^T w_j$ 
8:    $\tilde{v}_{j+1} := w_j - \sum_{i=1}^j \alpha_i v_i$ 
9:    $v_{j+1} := \tilde{v}_{j+1} / \|\tilde{v}_{j+1}\|$ 
10: end for
11: solve  $(W_m^T AV_m)y = W_m^T r_0$ 
12:  $u_m := u_0 + V_m y$ 

```

3.3.1 Low-rank projection method with restarting

As we observed in Section 3.2, matrix-vector products and vector sums in tensor structure tend to increase the rank of the resulting objects. Thus, although we seek a solution of low rank, straightforward use of the GMRES method may lead to

approximate solutions of higher rank than the desired solutions. This complication can be addressed using *truncation operators* [6,60,61,70,93], whereby vectors of high rank are replaced by ones of low rank. The truncation is inserted into the GMRES algorithm and is interleaved with the basic operations such as matrix-vector product and addition so that the ranks of the vectors used in the algorithm are kept low.

Algorithm 2 Restarted low-rank projection method in tensor format

```

1: set the initial solution  $\tilde{u}_0$ 
2: for  $k = 0, 1, \dots$  do
3:    $r_k := f - A\tilde{u}_k$ 
4:   if  $\|r_k\|/\|f\| < \epsilon$  then
5:     return  $\tilde{u}_k$ 
6:   end if
7:    $\tilde{v}_1 := \mathcal{T}_\kappa(r_k)$ 
8:    $v_1 := \tilde{v}_1/\|\tilde{v}_1\|$ 
9:   for  $j = 1, \dots, m$  do
10:     $w_j := Av_j$ 
11:    solve  $(V_j^T V_j)\alpha = V_j^T w_j$ 
12:     $\tilde{v}_{j+1} := \mathcal{T}_\kappa\left(w_j - \sum_{i=1}^j \alpha_i v_i\right)$ 
13:     $v_{j+1} := \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|$ 
14:   end for
15:   solve  $(W_m^T AV_m)\beta = W_m^T r_k$ 
16:    $\tilde{u}_{k+1} := \mathcal{T}_\kappa(\tilde{u}_k + V_m\beta)$ 
17: end for

```

Algorithm 2 summarizes the restarted low-rank projection method in tensor format [6]. As in the standard Arnoldi iteration used by GMRES, a new vector is constructed by applying the linear operator A to the previous basis vector v_j and orthogonalizing the new basis vector w_j with respect to the previous basis vectors $\{v_i\}_{i=1}^j$. The resulting vector is truncated to a vector \tilde{v}_{j+1} of low rank and normalized to v_{j+1} , which is then added to the set of basis vectors. The truncation operator \mathcal{T}_κ truncates a tensor of higher rank to one of rank κ . Thus, all the

basis vectors $\{v_i\}_{i=1}^m$ are of the same rank, κ . The basis vectors determine the subspace $\mathcal{K}_m = \text{span}\{v_1, \dots, v_m\}$, but because of truncation the basis vectors are not orthogonal and \mathcal{K}_m is not a Krylov subspace. However, it is still possible to project the residual onto the subspace $\mathcal{W}_m = \text{span}\{w_1, \dots, w_m\}$ to find out whether the residual can be decreased by forming a new iterate $\tilde{u}_k + V_m\beta$. Note that all the vectors used in the entire iteration process are stored as the product of two matrices in the form like that shown in (3.7). The ranks of these vectors will be discussed below.

3.3.2 Preconditioned low-rank projection method

To speed the convergence of the projection method, we consider a right-preconditioned system:

$$AM^{-1}\hat{u} = f, \quad \hat{u} = Mu. \quad (3.13)$$

For the stochastic diffusion problem, we consider $M = G_0 \otimes \tilde{K}_0 \approx G_0 \otimes K_0$ as the preconditioner, a mean-based preconditioner [81]. For the practical application of the preconditioner, we employ algebraic multigrid methods [90], where the action of K_0^{-1} is replaced by \tilde{K}_0^{-1} , an application of a single V-cycle of an algebraic multigrid method. The multigrid algorithm used point damped Jacobi smoothing with damping parameter .5 and two presmoothing and two postsmoothing steps, together with bilinear interpolation for grid transfer (as implemented in [98]). The preconditioned

matrix-vector product is then

$$AM^{-1}\hat{u} = \sum_{l=0}^M \sum_{k=1}^{\kappa_{\hat{u}}} G_l \hat{z}_k \otimes K_l \tilde{K}_0^{-1} \hat{y}_k, \quad \hat{u} = Mu = \sum_{i=1}^{\kappa_{\hat{u}}} \hat{z}_i \otimes \hat{y}_i.$$

Note that G_0^{-1} is the identity matrix because of the orthonormality of the stochastic basis functions. With right preconditioning and this preconditioner, the strategy for handling tensor rank is largely unaffected by preconditioning.

3.4 Truncation methods

As discussed in Section 3.3.1, in the low-rank projection method, truncation of tensors is essential for the efficient computation of approximate solutions. In this section, we discuss the conventional approach for truncation and we introduce a new coarse-grid truncation method based on a coarse-grid solution.

3.4.1 Truncation based on singular values

Given a matricized vector $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ' , a standard approach for truncation [6, 70] is to compute the singular value decomposition (SVD) of U and compress U into an approximation of desired rank $\kappa \ll \kappa'$. This can be done efficiently by computing QR factorizations of $Y_{\kappa'}$ and $Z_{\kappa'}$:

$$Y_{\kappa'} = Q_Y R_Y \in \mathbb{R}^{n_x \times \kappa'}, \quad Z_{\kappa'} = Q_Z R_Z \in \mathbb{R}^{n_\xi \times \kappa'}.$$

Then, one can compute the SVD of $R_Y R_Z^T$:

$$R_Y R_Z^T = \hat{U}_{\kappa'} \hat{\Sigma}_{\kappa'} \hat{V}_{\kappa'}^T = \sum_{k=1}^{\kappa'} \hat{\sigma}_k \hat{u}_k \hat{v}_k^T$$

and truncate the sum with κ terms to produce

$$\tilde{Y}_\kappa = Q_Y \hat{U}_\kappa \hat{\Sigma}_\kappa \in \mathbb{R}^{n_x \times \kappa}, \quad \tilde{Z}_\kappa = Q_Z \hat{V}_\kappa \in \mathbb{R}^{n_\xi \times \kappa}.$$

The truncated approximation of U is then $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$. The computational complexity of the truncation is $O((n_x + n_\xi + \kappa)(\kappa')^2)$ [47], which grows quadratically with respect to κ' . In the next section, we introduce a new truncation method that avoids this computation.

3.4.2 Truncation based on coarse-grid rank-reduction

We now propose a coarse-grid rank-reduction strategy. We obtain insight into the rank structure of the solution using a coarse spatial grid computation. Then, we define a truncation operator based on the information obtained from this coarse-grid computation.

Let $u^c(x, \xi)$ represent a solution obtained on a coarse spatial grid (i.e., n_x is small). As in (3.8), $u^c(x, \xi)$ can be represented as

$$u^c(x, \xi) = (\Phi^c(x))^T U^c \Psi(\xi) = ((Y^c)^T \Phi^c(x))^T ((Z^c)^T \Psi(\xi)). \quad (3.14)$$

Here, we propose to use Z^c to define a truncation operator for use in the projection method to compute a solution for the problem on a finer grid. That is, the truncation operator is defined such that, given a matricized vector $U = Y_{\kappa'} Z_{\kappa'}^T$ of rank κ' ,

$$\mathcal{T}_\kappa(U) \equiv (Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c) (Z_\kappa^c)^T = \tilde{U} \quad (3.15)$$

where the resulting quantity $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$ is of rank κ ,

$$\tilde{Y}_\kappa = Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c \in \mathbb{R}^{n_x \times \kappa}, \quad \tilde{Z}_\kappa = Z_\kappa^c \in \mathbb{R}^{n_\xi \times \kappa}.$$

The desired rank κ is determined such that the relative residual $\|f^c - A^c u^{c,\kappa}\|_2 / \|f^c\|_2$ is smaller than a certain tolerance ϵ^c where $u^{c,\kappa}$ is a κ -term approximation of u^c . This truncation operation requires two matrix-matrix products, and the computational complexity of truncating a vector from κ' to κ is $O(\kappa' \kappa (n_x + n_\xi))$. Note that with the proposed truncation strategy, the fine-grid computation is equivalent to applying GMRES to $\sum_{i=0}^M K_i U G_i Z_\kappa^c (Z_\kappa^c)^T = f_0 g_0^T$.

For efficient coarse-grid computation, we use the Proper Generalized Decomposition (PGD) method developed in [76, 104], which computes a separated representation of a coarse-grid solution:

$$u^{c,\kappa}(x, \xi) = \sum_{i=1}^{\kappa} \tilde{y}_i(x) \tilde{z}_i(\xi). \quad (3.16)$$

With the stochastic Galerkin discretization, each function can be represented as

$$\tilde{y}_i(x) = \sum_{k=1}^{n_x} \tilde{y}_k^{(i)} \phi_k^c(x), \quad \tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi} \tilde{z}_l^{(i)} \psi_l(\xi).$$

As a result, as in (3.8),

$$u^{c,\kappa}(x, \xi) = \left((\tilde{Y}_\kappa^c)^T \Phi^c(x) \right)^T \left((\tilde{Z}_\kappa^c)^T \Psi(\xi) \right)$$

where $\tilde{Y}_\kappa^c = [\tilde{y}^{(1)}, \dots, \tilde{y}^{(\kappa)}] \in \mathbb{R}^{n_x \times \kappa}$ and $\tilde{Z}_\kappa^c = [\tilde{z}^{(1)}, \dots, \tilde{z}^{(\kappa)}] \in \mathbb{R}^{n_\xi \times \kappa}$ are coefficient matrices such that the i th elements of $\tilde{y}^{(j)}$ and $\tilde{z}^{(j)}$ are $\tilde{y}_i^{(j)}$ and $\tilde{z}_i^{(j)}$, respectively.

Now, the discrete solution U^c in (3.14) is approximated by $U^{c,\kappa} = \tilde{Y}_\kappa^c (\tilde{Z}_\kappa^c)^T$, and we can obtain Z_κ^c by computing the SVD of $U^{c,\kappa} = \hat{U} \hat{\Sigma} \hat{V}^T$, and, as a result, $Z_\kappa^c = \hat{V}$.

We briefly explain how the PGD method computes a κ -term approximation in the next section.

3.4.3 Proper Generalized Decomposition method

The PGD method is a successive rank-1 approximation method. That is, the method incrementally identifies the function pairs $(\tilde{y}_i(x), \tilde{z}_i(\xi))$ of (3.16) one at a time. Once i such pairs have been computed, the next pair $(\tilde{y}_{i+1}, \tilde{z}_{i+1})$ is sought in $X_h \times S_M$ by imposing Galerkin orthogonality with respect to the tangent manifold of the set of rank-one elements at $\tilde{y}_{i+1} \tilde{z}_{i+1}$, which is $\{\tilde{y}_{i+1} \zeta + v \tilde{z}_{i+1}; v \in X_h, \zeta \in S_M\}$:

find $\tilde{y}_{i+1}\tilde{z}_{i+1}$ such that $\forall(v, \zeta) \in X_h \times S_M$

$$\left\langle \int_D a(x, \xi) \nabla(u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla(\tilde{y}_{i+1}\zeta + v\tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f(\tilde{y}_{i+1}\zeta + v\tilde{z}_{i+1}) \right\rangle. \quad (3.17)$$

It follows from (3.17) that each component of a pair $(\tilde{y}_{i+1}, \tilde{z}_{i+1})$ can be computed by solving two coupled problems: a deterministic problem (3.18) and a stochastic problem (3.19). The deterministic problem is as follows: given \tilde{z}_{i+1} , find $\tilde{y}_{i+1} \in X_h$ such that

$$\left\langle \int_D a(x, \xi) \nabla(u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla(\phi_j^c \tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f \phi_j^c \tilde{z}_{i+1} \right\rangle, \quad j = 1, \dots, n_x^c. \quad (3.18)$$

The first basis function \tilde{z}_1 can be chosen arbitrarily at the beginning of the PGD method. The finite element discretization of u_{i+1} yields a linear system of order n_x^c . Analogously, the stochastic problem starts with \tilde{y}_{i+1} and finds $\tilde{z}_{i+1} \in S_M$ such that

$$\left\langle \int_D a(x, \xi) \nabla(u^{c,i} + \tilde{y}_{i+1}\tilde{z}_{i+1}) \cdot \nabla(\tilde{y}_{i+1}\psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_{i+1}\psi_j \right\rangle, \quad j = 1, \dots, n_\xi. \quad (3.19)$$

Since \tilde{z}_{i+1} is approximated by the gPC, n_ξ unknowns have to be determined by solving a linear system of order n_ξ .

Solutions of these sets of κ systems of order n_x^c and κ systems of order n_ξ produce the κ -term approximation to the solution. The PGD method seeks solution

pairs until the relative residual of the computed solution satisfies a given tolerance,

$$\|f^c - A^c u^{c,\kappa}\|_2 / \|f^c\|_2 < \epsilon^c. \quad (3.20)$$

The accuracy of the κ -term approximation can also be improved by solving a set of κ coupled equations: given $\{\tilde{y}_i\}_{i=1}^\kappa$, find $\{\tilde{z}_i\}_{i=1}^\kappa$ such that

$$\left\langle \int_D a(x, \xi) \nabla(u^{(\kappa)}) \cdot \nabla(\tilde{y}_i \psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_i \psi_j \right\rangle, \quad i = 1, \dots, \kappa, j = 1, \dots, n_\xi. \quad (3.21)$$

This update requires the solution of a linear system of order κn_ξ . For the stochastic diffusion problems, the update problem is solved once at the end of the PGD method. Note that the update problem could also be formulated for finding the deterministic parts $\{u_i\}_{i=1}^\kappa$ if $n_x \ll n_\xi$, which requires a solution of a linear system of order κn_x .

With the proposed truncation strategy, Algorithm 3 summarizes the entire procedure to compute a solution on a finer grid.

Algorithm 3 Preconditioned low-rank projection method with the coarse-grid rank-reduction

- 1: Compute $u^{c,\kappa}$ that satisfies $\frac{\|f^c - A^c u^{c,\kappa}\|_2}{\|f^c\|_2} < \epsilon^c$ using the PGD method
 - 2: Compute Z_κ^c such that $U^{c,\kappa} = Y_\kappa^c (Z_\kappa^c)^T$ and define $\mathcal{T}_\kappa(U) \equiv (U Z_\kappa^c) (Z_\kappa^c)^T$
 - 3: Run Algorithm 2 with $\mathcal{L} = AM^{-1}$, f , and \mathcal{T}_κ
-

3.5 Numerical experiments

In this section, we present the results of numerical experiments in which the proposed iterative solver is applied to some benchmark problems. The implementa-

tion of the spatial discretization is based on the Incompressible Flow and Iterative Solver Software (IFISS) package [98]. Example problems are posed on a square domain and ℓ is the spatial discretization parameter (i.e., $n_x = (2^\ell + 1)^2$).

For $a(x, \xi)$ in (2.7), we consider independent random variables $\{\xi_i\}_{i=1}^M$ that are uniformly distributed over $[-\sqrt{3}, \sqrt{3}]$, $a_0 = 1$ and unless otherwise specified, $\sigma = 0.05$. As the covariance kernel, we use

$$C(x, y) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{\gamma} - \frac{|x_2 - y_2|}{\gamma}\right) \quad (3.22)$$

where γ is the correlation length. The number of terms M in the truncated expansion (2.7) is determined such that 95% of the total variance is captured by M terms (i.e., $(\sum_{i=1}^M \lambda_i) / (\sum_{i=1}^{n_x} \lambda_i) > 0.95$). We use bilinear Q_1 elements to generate the finite element basis and Legendre polynomials as the stochastic basis functions because the underlying random variables have a uniform distribution. The default setting of the maximal polynomial degree p is 3.

3.5.1 Stochastic diffusion problem

We consider the steady-state stochastic diffusion equation in (2.8) on a domain $D = [0, 1] \times [0, 1]$ with forcing term $f(x) = 1$ and homogeneous Dirichlet boundary conditions, $u(x, \omega) = 0$ on $\partial D \times \Gamma$.

Coarse spatial grid computation. We compute κ -term approximations using the PGD method on a coarser spatial grid. Here ℓ^c is the refinement level for the coarse grid and n_x^c is the number of degrees of freedom in the corresponding

spatial domain excluding boundary nodes. We discuss choices of coarse spatial grid in Section 3.5.3. Table 3.1 shows the rank κ of solutions that satisfy the tolerance ϵ^c for varying correlation lengths γ and M and the computation time t_c . In PGD, the linear systems arising from (3.18), (3.19), and (3.21) are solved using MATLAB’s backslash operator.

Table 3.1: Rank (κ) of coarse-grid solutions satisfying ϵ^c of (3.20), and CPU time (t_c) for coarse-grid computation using the PGD method, for varying γ and M .

	$\epsilon^c = 10^{-5}$				$\epsilon^c = 10^{-6}$			
γ	4	3	2.5	2	4	3	2.5	2
M, n_ξ	5, 56	7, 120	10, 286	15, 816	5, 56	7, 120	10, 286	15, 816
$n_x^c(\ell^c)$	225(4)	225(4)	961(5)	961(5)	225(4)	225(4)	961(5)	961(5)
Rank(κ)	25	40	65	115	35	65	100	210
CPU time(t_c)	2.49	3.47	8.35	45.08	2.93	5.04	14.83	162.71

Fine spatial grid computation. With the truncation operator \mathcal{T}_κ (3.15) obtained from the coarse-grid solution (i.e., Z_κ^c), we solve the same stochastic diffusion problems on finer spatial grids $\ell = \{7, 8, 9\}$. For the fine-grid low-rank solutions, we use the rank κ obtained from the coarse-grid solutions. For example, the third column of Table 3.2 shows the time required to find solutions of rank 25 satisfying the relative residual tolerance 10^{-5} when the number of terms in (2.7) is $M = 5$. In Algorithm 2, we set $m = 8$ (like restarted GMRES(8)). In examining performance, we identify the number of cycles, k , performed for the outer for-loop in Algorithm 2; this means that the number of matrix-vector products (i.e., the number of times line 10 is executed) is mk . Tables 3.2 and 3.3 show the number of cycles, k , and the computation time in seconds needed to compute approximate solutions with

$\epsilon = 10^{-5}$ and 10^{-6} , respectively, (see line 4 of Algorithm 2). Here, t is the total time and t_f excludes the time to compute the coarse-grid solution, t_c . The fine-grid computation time, t_f , consists of algorithm execution time and preconditioner set-up time, t_{setup} .

Table 3.2: CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ using the preconditioned low-rank projection method. Numbers of GMRES cycles are shown in parentheses.

$n_x(\ell)$		$M=5$	$M=7$	$M=10$	$M=15$	t_{setup}
129 ² (7)	t_f	4.12 (1)	7.22 (1)	18.79 (1)	86.29 (1)	1.76
	t	8.35	12.43	28.88	132.15	
257 ² (8)	t_f	12.55 (1)	24.70 (1)	74.71 (1)	330.45 (1)	10.16
	t	25.17	38.37	93.20	385.59	
513 ² (9)	t_f	92.83 (1)	102.42 (1)	353.07 (1)	2717.03 (1)	92.41
	t	147.17	197.87	453.71	2854.62	

Table 3.3: CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-6}$ using the preconditioned low-rank projection method. Numbers of GMRES cycles are shown in parentheses.

$n_x(\ell)$		$M=5$	$M=7$	$M=10$	$M=15$	t_{setup}
129 ² (7)	t_f	5.40 (1)	12.50 (1)	35.09 (1)	233.54 (1)	1.79
	t	10.14	19.32	51.69	398.06	
257 ² (8)	t_f	17.23 (1)	46.07 (1)	137.19 (1)	1004.40 (1)	10.53
	t	30.55	61.41	162.90	1177.68	
513 ² (9)	t_f	70.37 (1)	217.12 (1)	1225.77 (1)	OoM	92.81
	t	166.24	315.18	1333.63	OoM	

The execution times show “textbook” behavior, i.e., they grow linearly with the size of the spatial grid.¹ Note that the computational cost for the coarse-grid computation becomes negligible as the size of the problem becomes higher. If the

¹An exception to this statement is when both M and n_x are large. For these cases, the problem does not fit into physical memory and memory swap-in/out time dominates the execution time.

required memory for running Algorithm 2 exceeds the resources of our computing environment, solutions could not be computed and we denote these cases by OoM for “Out-of-Memory”. Table 3.4 shows the number of degrees of freedom of the fine spatial-grid problems for varying stochastic dimensions, M .

In these experiments and in all those described below, we used $\epsilon^c = \epsilon$ (the stopping tolerance specified in line 4 of Algorithm 2), and for this choice, the solver always satisfied the stopping criterion. We also tested both larger ϵ^c and smaller ϵ^c . For $\epsilon^c > \epsilon$, the solver sometimes failed to satisfy the stopping criterion. For $\epsilon^c < \epsilon$, the solver was robust but consistently more expensive.

Table 3.4: Number of degrees of freedom of the fine-grid discretizations with $p = 3$, for varying spatial-grid refinement level, ℓ , and number of random variables, M .

ℓ	$M=5$	$M=7$	$M=10$	$M=15$
7	931,896	1,996,920	4,759,326	13,579,056
8	3,698,744	7,925,880	18,890,014	53,895,984
9	14,737,464	31,580,280	75,266,334	214,745,904

Example problems with varying σ and p . We examine the rank structure of the numerical solutions of the stochastic diffusion problems and assess the performance of the proposed solution algorithm for different values of maximal degree of stochastic polynomial, p in (2.13), and variance σ^2 of the random field $a(x, \xi)$. As in the previous numerical experiments, we first identify the rank structure and define the truncation operator from coarse-grid computation. Then, we solve the same problems on a finer grid by using the proposed low-rank projection method with the coarse-grid rank-reduction scheme.

Table 3.5 shows the computation time needed to compute approximate solutions of the stochastic diffusion problems with $M = 7$ for varying maximal polynomial degree p . The required ranks of the approximate solutions are not affected by the number of terms in the polynomial expansion. However, the computation time is increased for the polynomial expansion with higher maximal polynomial degree because the size of $\{G_i\}_{i=0}^M$ and the size of the stochastic part of the solution gets larger as the number of terms in the gPC is increased.

Table 3.5: CPU times t to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection method for varying maximal polynomial degree p (stochastic dofs, n_ξ , in the parenthesis).

$n_x(\ell)$	$\epsilon^c = \epsilon = 10^{-5} (\kappa = 40)$			$\epsilon^c = \epsilon = 10^{-6} (\kappa = 65)$		
	$p = 3$ (120)	$p = 4$ (330)	$p = 5$ (792)	$p = 3$ (120)	$p = 4$ (330)	$p = 5$ (792)
$129^2(7)$	12.43	15.55	21.56	19.32	23.42	38.49
$257^2(8)$	38.37	44.27	56.79	61.41	69.17	91.10
$513^2(9)$	197.87	217.38	252.39	315.18	322.86	383.89

Table 3.6 shows the computation time t needed to compute approximate solutions of the stochastic diffusion problems that satisfy the tolerance 10^{-5} and 10^{-6} for varying variance, σ^2 . In general, the example problem with a larger variance requires a higher rank to satisfy the stopping tolerance, which, therefore, requires more computational effort.

Comparison to a truncation operator based on singular values. We compare the performance of the proposed solver to the preconditioned low-rank projection method combined with the conventional truncation operator from [61]. Table 3.7 shows the computation time required to compute approximate solutions

Table 3.6: CPU times t and rank κ to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection method for varying σ .

σ	n_x	$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
		$M=5$	$M=7$	$M=10$	$M=15$	$M=5$	$M=7$	$M=10$	$M=15$
0.01		$\kappa = 15$	$\kappa = 20$	$\kappa = 35$	$\kappa = 55$	$\kappa = 20$	$\kappa = 30$	$\kappa = 50$	$\kappa = 85$
	129^2	7.28	8.65	15.01	45.69	7.87	10.81	20.76	83.07
	257^2	21.47	26.08	47.21	135.75	23.30	31.94	66.92	240.98
	513^2	130.93	150.85	236.34	922.87	137.98	173.03	333.70	1893.89
0.05		$\kappa = 25$	$\kappa = 40$	$\kappa = 65$	$\kappa = 115$	$\kappa = 35$	$\kappa = 65$	$\kappa = 100$	$\kappa = 210$
	129^2	8.35	12.43	28.88	132.15	10.14	19.32	51.69	398.06
	257^2	25.17	38.37	93.20	385.59	30.55	61.41	162.90	1177.68
	513^2	147.17	197.87	453.71	2854.62	166.24	315.18	1333.63	OoM
0.1		$\kappa = 35$	$\kappa = 60$	$\kappa = 100$	$\kappa = 180$	$\kappa = 50$	$\kappa = 85$	$\kappa = 145$	-
	129^2	9.78	17.24	50.70	297.35	8.79	28.37	113.53	OoM
	257^2	29.98	54.94	157.76	866.41	41.69	94.48	356.50	OoM
	513^2	164.48	273.33	1324.47	OoM	208.15	515.29	2902.95	OoM

using the conventional and new truncation strategies. The total computation time, t , of the low-rank projection method with the coarse-grid rank reduction includes both coarse-grid, t_c , and fine-grid computations, t_f . The low-rank projection method with the SVD-based truncation operator, which is implemented based on [6], does not require a coarse-grid computation and can start with any arbitrary initial guess for rank, κ . For these computations, we used the values of rank identified in the coarse-grid computations, which are illustrated in Table 3.1, for the initial rank.

Table 3.7: CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection (LRP) methods with the coarse-grid rank-reduction and the singular value based truncation on the level 8 spatial grid (i.e., $n_x = 257^2$).

	Solver		$M=5$	$M=7$	$M=10$	$M=15$	$M=20$
$\epsilon = 10^{-5}$	LRP-SVD	t_{SVD}	55.04	108.11	284.27	1280.65	5691.19
	LRP-Coarse	t	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	LRP-SVD	t_{SVD}	76.03	198.20	564.12	5131.32	OoM
	LRP-Coarse	t	30.55	61.41	162.90	1177.68	OoM

Table 3.8: CPU times to compute low-rank solutions of the diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the PGD method and the preconditioned low-rank projection methods on the level 8 spatial grid (i.e., $n_x = 257^2$).

	Solver		$M=5$	$M=7$	$M=10$	$M=15$	$M=20$
$\epsilon = 10^{-5}$	PGD	κ	25	45	65	125	195
		t	43.78	109.72	228.73	940.69	3066.87
	LRP-Coarse	κ	25	40	65	115	180
		t	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	PGD	κ	40	70	110	225	OoM
		t	74.43	214.82	533.10	2713.70	OoM
	LRP-Coarse	κ	35	65	100	210	OoM
		t	30.55	61.41	162.90	1177.68	OoM

PGD as a solver on a finer spatial grid. The PGD method could be applied directly to the fine-grid problems. We assess the performance of the PGD method for computing fine-grid solutions in Table 3.8, which shows the rank and computation time for computing approximate solutions that satisfy the tolerance 10^{-5} and 10^{-6} using PGD on a finer spatial grid. For the low-rank projection method, we record total computation time, t , which includes coarse-grid computation, t_c , AMG preconditioner set-up, t_{setup} , and fine-grid computation time, t_f .

We compare the rank and the computation time for computing solutions using the PGD method and the proposed projection method. The proposed low-rank projection method runs faster and requires somewhat smaller ranks than the PGD method.

Remark. We also tested the techniques compared in Tables 3.7 and 3.8 for different values of σ , $\sigma = 0.01$ and 0.1 , with similar results. Indeed, the performance of LRP-Coarse is more favorable for the larger value $\sigma = 0.1$.

3.5.2 Stochastic convection-diffusion problem

For a second benchmark problem, we consider the steady-state convection-diffusion equation defined on $D = [-1, 1] \times [-1, 1]$ with non-homogeneous Dirichlet boundary conditions, constant vertical wind $\vec{w} = (0, 1)$, and $f = 0$,

$$\begin{cases} \nu \nabla \cdot (a(x, \xi) \nabla u(x, \xi)) + \vec{w} \cdot \nabla u(x, \xi) = f(x, \xi) & \text{in } D \times \Gamma, \\ u(x, \xi) = g_D(x) & \text{on } \partial D \times \Gamma, \end{cases} \quad (3.23)$$

where $g_D(x)$ is determined by

$$g_D(x) = \begin{cases} g_D(x, -1) = x, & g_D(x, 1) = 0, \\ g_D(-1, y) = -1, & g_D(1, y) = 1, \end{cases} \quad (3.24)$$

where the latter two approximations hold except near $y = 1$, and ν is the viscosity parameter. We consider the convection-dominated case (i.e., $\nu < 1$) and employ the streamline-diffusion method for stabilization [17]. Here, we define the element

Peclet number

$$\mathcal{P}_k = \frac{\|\vec{w}_k\|_2 h_k}{2\nu} \quad (3.25)$$

where $\|\vec{w}_k\|_2$ is the ℓ_2 norm of the wind at the element centroid and h_k is a measure of the element length in the direction of the wind. Note that the solution has an *exponential boundary layer* near $y = 1$ where the value of the solution dramatically changes essentially from -1 to 0 on the left and $+1$ to 0 on the right [39]. Figure 3.1 illustrates the mean of solutions $\langle u(x, \xi) \rangle_\rho$ computed on the level 6 spatial grid and corresponding contour plots for varying viscosity parameter, ν .

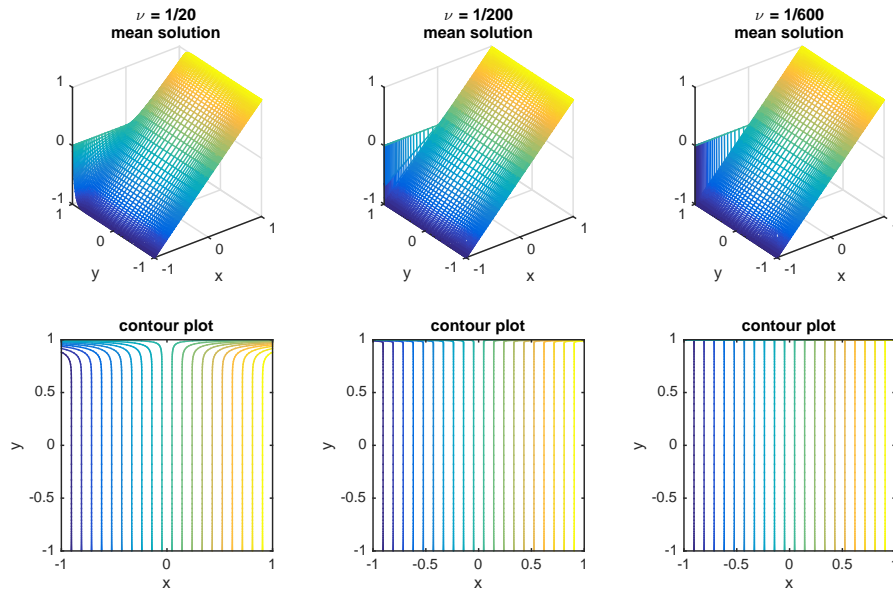


Figure 3.1: Mean solutions and contour plots on the level 6 spatial grid for varying ν .

Given $a(x, \xi)$ in (2.7), we again discretize (3.23) using the finite element method and the gPC expansion. The result is a linear system in tensor product

notation

$$\left(G_0 \otimes \nu K_0 + \sum_{l=1}^M G_l \otimes \nu K_l + G_0 \otimes N + G_0 \otimes S \right) u = g_0 \otimes f_0 \quad (3.26)$$

where the convection term N and the streamline-diffusion term S are given by

$$\begin{aligned} [N]_{ij} &= \int_D \vec{w} \cdot \nabla \phi_i(x) \phi_j(x) dx, \\ [S]_{ij} &= \sum_{k=1}^{n_e} \delta_k \int_D (\vec{w} \cdot \nabla \phi_i) (\vec{w} \cdot \nabla \phi_j) dx, \end{aligned}$$

n_e is the number of elements in the finite element discretization, and

$$\delta_k = \begin{cases} \frac{h_k}{2\|\vec{w}\|_2} \left(1 - \frac{1}{\mathcal{P}_k} \right) & \text{if } \mathcal{P}_k > 1 \\ 0 & \text{if } \mathcal{P}_k \leq 1 \end{cases}. \quad (3.27)$$

As the preconditioner, we choose $M \approx G_0 \otimes (K_0 + N + S)$ where the action of $(K_0 + N + S)^{-1}$ is replaced by application of a single V-cycle of an AMG method. In the PGD method, the non-homogeneous Dirichlet boundary condition is handled by introducing an extended affine space [76]: $u^c \approx u_{bc} + u^{c,\kappa}$ where u_{bc} is the boundary nodal functions such as $u_{bc} = \sum_{k \in \partial D} u_k^{(bc)} \phi_k(x)$. For the stochastic convection-diffusion problems, the update problems (3.21) need to be solved more often to compute an approximate solution of a desired accuracy with fewer terms.

Numerical results. To cope with the existence of the exponential boundary layer in the solution, we use vertically stretched spatial grids. We examine the performance of the low-rank projection method for varying viscosity parameter ν ,

and we set $m = 10$ for Algorithm 2. Table 3.9 and 3.10 show κ computed by the PGD method, coarse-grid computation time t_c , and fine grid computation time t_f to compute approximate solutions on fine spatial grids $\ell = \{7, 8, 9\}$ satisfying 10^{-5} and 10^{-6} , respectively. Underlined numbers in the spatial grid level indicates cases where streamline diffusion is not needed.

Table 3.9: CPU times to compute low-rank solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ using the preconditioned low-rank projection methods for varying ν . Numbers of GMRES cycles are shown in parentheses.

ν	ℓ		$M = 5$	$M = 7$	$M = 10$	$M = 15$	t_{setup}
$\frac{1}{20}$	4	κ	25	35	55	65*	
		t_c	2.56	4.83	26.34	58.92*	
	<u>7</u>	t_f	5.73 (1)	9.47 (1)	24.86 (1)	72.29 (1)	6.14
	<u>8</u>	t_f	20.52 (1)	36.66 (1)	98.72 (1)	248.31 (1)	30.57
	<u>9</u>	t_f	84.55 (1)	152.69 (1)	592.63 (1)	1953.52 (2)	338.28
$\frac{1}{100}$	4	κ	20	25	45	55*	
		t_c	2.94	3.12	16.28	47.24*	
	7	t_f	5.06 (1)	7.28 (1)	18.90 (1)	60.66 (1)	6.34
	8	t_f	16.87 (1)	26.36 (1)	74.26 (1)	202.29 (1)	35.52
	<u>9</u>	t_f	121.98 (2)	201.62 (2)	745.92 (2)	3079.24 (2)	341.41
$\frac{1}{200}$	5	κ	20	25	45	50	
		t_c	2.91	4.79	16.54	46.85	
	7	t_f	5.16 (1)	7.21 (1)	16.57 (1)	53.97 (1)	6.35
	8	t_f	17.57 (1)	25.05 (1)	63.56 (1)	175.30 (1)	35.89
	9	t_f	123.73 (2)	200.10 (2)	605.50 (2)	2568.41 (2)	344.87
$\frac{1}{400}$	5	κ	20	20	35	45 [†]	
		t_c	2.94	3.79	12.49	82.06 [†]	
	7	t_f	8.61 (2)	9.84 (2)	26.97 (2)	85.01 (2)	6.09
	8	t_f	31.55 (2)	37.74 (2)	111.31 (2)	298.49 (2)	34.93
	9	t_f	133.45 (2)	158.01 (2)	512.88 (2)	2080.60 (2)	342.12
$\frac{1}{600}$	6	κ	20	20	35	45	
		t_c	9.79	13.20	34.47	94.79	
	7	t_f	8.27 (2)	10.07 (2)	26.91 (2)	82.30 (2)	6.14
	8	t_f	31.94 (2)	39.84 (2)	109.25 (2)	295.25 (2)	33.25
	9	t_f	343.80 (2)	163.90 (2)	506.42 (2)	1977.83 (2)	342.98

When the viscosity parameter is small (i.e., $\nu = 1/600$), the coarse-grid com-

putation requires the κ -term approximation on a relatively fine spatial grid (i.e., $\ell = 6$). The exponential boundary layer gets narrower as the viscosity parameter gets smaller, which requires the use of a finer spatial grid for the coarse-grid computation. If the coarse-grid computation is performed on coarser spatial grids, it fails to identify the rank structure of solutions and to yield a proper truncation operator. Analogously, when the number of terms, M , in the KL expansion (2.7) is large, the coarse-grid computation has to be done on a relatively fine spatial grid because the KL expansion contains more spatially oscillatory terms. In the last columns of Table 3.9 and 3.10, * and † indicate that the coarse-grid solutions are computed on the level 5 and the level 6 spatial grid, respectively.

Table 3.10: CPU times to compute approximate solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-6}$ using the preconditioned low-rank projection methods for varying ν . Numbers of GMRES cycles are shown in parentheses.

ν	ℓ		$M = 5$	$M = 7$	$M = 10$	$M = 15$	t_{setup}
$\frac{1}{20}$	4	κ	35	50	75	105*	
		t_c	3.31	9.17	60.51	194.33*	
	<u>7</u>	t_f	13.92 (2)	27.47 (2)	80.78 (2)	275.96 (2)	6.14
	<u>8</u>	t_f	52.45 (2)	106.11 (2)	311.59 (2)	1042.40 (2)	30.57
	<u>9</u>	t_f	220.67 (2)	534.61 (2)	2694.26 (2)	8101.20 (2)	338.28
$\frac{1}{100}$	4	κ	30	40	65	95*	
		t_c	2.83	6.25	38.39	155.83*	
	7	t_f	12.34 (2)	21.28 (2)	65.02 (2)	239.91 (2)	6.34
	8	t_f	46.67 (2)	85.66 (2)	255.79 (2)	895.81 (2)	35.52
	<u>9</u>	t_f	273.45 (3)	549.82 (3)	3069.96 (3)	10963.03 (3)	341.41
$\frac{1}{200}$	5	κ	25	40	60	85	
		t_c	3.46	8.57	38.35	122.49	
	7	t_f	10.52 (2)	21.43 (2)	56.36 (2)	204.09 (2)	6.35
	8	t_f	39.39 (2)	84.14 (2)	219.36 (2)	732.88 (2)	35.89
	9	t_f	226.83 (3)	547.62 (3)	2627.98 (3)	9284.60 (3)	344.87
$\frac{1}{400}$	5	κ	25	35	55	75 [†]	
		t_c	3.49	6.63	30.50	151.46 [†]	
	7	t_f	10.44 (2)	17.96 (2)	50.96 (2)	161.58 (2)	6.09
	8	t_f	40.02 (2)	70.82 (2)	204.71 (2)	610.23 (2)	34.93
	9	t_f	239.04 (3)	441.73 (3)	2106.30 (3)	7817.82 (3)	342.12
$\frac{1}{600}$	6	κ	30	35	45	65	
		t_c	17.99	22.03	47.44	140.01	
	7	t_f	17.74 (3)	26.56 (3)	56.25 (3)	281.27 (3)	6.14
	8	t_f	48.39 (2)	74.40 (2)	153.35 (2)	506.84 (2)	33.25
	9	t_f	281.27 (3)	462.52 (3)	1184.74 (3)	6261.34 (3)	342.98

Comparison to a truncation operator based on singular values. We again compare the performance of the proposed solver to the preconditioned low-rank projection method combined with the conventional truncation operator, the SVD-based truncation operator. Table 3.11 shows the computation time required to compute approximate solutions using the conventional and the new truncation strategy. When the low-rank projection method with SVD-based truncation oper-

ator is used, initial values for rank κ in Algorithm 2 are obtained from coarse-grid computations of the proposed rank reduction strategy.

Table 3.11: CPU times to compute low-rank solutions of the convection-diffusion equation for $\epsilon^c = \epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection (LRP) methods with the coarse-grid rank-reduction and the singular value based truncation on the level 8 spatial grid (i.e., $n_x = 257^2$).

	Viscosity (ν)	Solver		$M = 5$	$M = 7$	$M = 10$	$M = 15$	
$\epsilon = 10^{-5}$	1/20	LRP-SVD	t_{SVD}	68.45	100.83	201.34	438.25	
		LRP-Coarse	t	54.06	72.08	154.79	338.21	
	1/100	LRP-SVD	t_{SVD}	93.91	121.89	295.27	655.71	
		LRP-Coarse	t	55.28	64.36	125.88	285.94	
	1/200	LRP-SVD	t_{SVD}	90.70	122.56	251.60	574.68	
		LRP-Coarse	t	55.42	66.08	115.68	258.97	
	1/400	LRP-SVD	t_{SVD}	91.11	107.47	221.32	475.60	
		LRP-Coarse	t	69.01	76.63	158.07	416.36	
	1/600	LRP-SVD	t_{SVD}	90.33	103.44	218.35	484.08	
		LRP-Coarse	t	75.26	86.48	176.93	422.85	
	$\epsilon = 10^{-6}$	1/20	LRP-SVD	t_{SVD}	132.08	234.15	570.56	1748.43
			LRP-Coarse	t	86.74	145.86	401.83	1267.71
1/100		LRP-SVD	t_{SVD}	121.88	196.66	471.11	1479.80	
		LRP-Coarse	t	84.97	126.77	329.52	1088.05	
1/200		LRP-SVD	t_{SVD}	106.79	188.76	416.52	1203.78	
		LRP-Coarse	t	77.79	128.96	293.30	892.18	
1/400		LRP-SVD	t_{SVD}	107.12	168.01	380.01	1015.88	
		LRP-Coarse	t	78.04	112.55	269.48	797.50	
1/600		LRP-SVD	t_{SVD}	122.44	231.07	421.76	1208.88	
		LRP-Coarse	t	97.00	129.87	234.00	670.90	

3.5.3 Choices of coarse spatial grid

Finally, we discuss criteria for choosing the coarse grid used to generate truncation operators. The basic idea is that the coarse grid needs to be fine enough

so that important features of the problem are represented. This quality is problem dependent, and we outline what is needed for the two types of problems we examined.

First consider the diffusion equation of Section 5.1. The issue is the oscillatory nature of components of the random field $a(x, \xi)$. In the KL expansion (2.7), the eigenpairs, $\{(\lambda_i, a_i(x))\}_{i=1}^M$, can be obtained by solving the following integral equation,

$$\int_D C(x, y)a_i(y)dy = \lambda_i a_i(x), \quad i = 1, \dots, M \quad (3.28)$$

where $C(x, y)$ is the covariance kernel (3.22). Since the kernel is separable, the eigenfunctions of the integral problem (3.28) can be represented as $a_i(x) = a_k^1(x_1)a_j^2(x_2)$, where $\{a_k^1\}_{k=1}^\infty$ and $\{a_j^2\}_{j=1}^\infty$ are the eigenfunctions of the one-dimensional integral problem (i.e., $\int_D \exp(-|x_l - y_l|/\gamma)a_k^l(y_l)dy_l = \lambda_k^l a_k^l(x_l)$, $l = 1, 2$). The eigenvalues, $\{\lambda_i\}_{i=1}^M$, are in decreasing order and λ_i is the i th largest value of products $\lambda_k^1 \lambda_j^2$ for $k, j = 1, 2, \dots$. Analytic expressions for the 1D eigenfunctions are given in [46] as, for $l = 1, 2$,

$$\begin{aligned} a_k^l(x) &= \cos(\theta_k x) \Big/ \sqrt{\frac{1}{2} + \frac{\sin \theta_k}{2\theta_k}} && \text{for even } k, \\ a_k^{l*}(x) &= \sin(\theta_k^* x) \Big/ \sqrt{\frac{1}{2} - \frac{\sin \theta_k}{2\theta_k}} && \text{for odd } k, \end{aligned} \quad (3.29)$$

where θ_k and θ_k^* are the solutions of

$$\frac{1}{c} - \theta \tan\left(\frac{\theta}{2}\right) = 0 \quad \text{and} \quad \theta^* + \frac{1}{c} \tan\left(\frac{\theta^*}{2}\right) = 0,$$

Table 3.12: Largest values of θ_k or θ_k^* of eigenfunctions (3.29) in the KL expansion, required grid refinement level ℓ^c , half wavelength π/θ , and element size $h^c = 2^{-\ell^c}$ for different values of M .

M	3	5	7	10	15	20
$\max(\theta_k, \theta_k^*)$	3.25	6.36	9.49	12.63	18.90	25.19
wavelength/2	.97	.49	.33	.25	.17	.12
$\ell^c (h^c)$	3 ($\frac{1}{8}$)	4 ($\frac{1}{16}$)	4 ($\frac{1}{16}$)	5 ($\frac{1}{32}$)	5 ($\frac{1}{32}$)	6 ($\frac{1}{64}$)

respectively, when the 1D integral problem is posed on $[-\frac{1}{2}, \frac{1}{2}]$. As i in the KL expansion (2.7) increases, the eigenfunctions $a_i(x)$ become more oscillatory over the spatial domain (i.e., θ_k or θ_k^* become larger), so that finer coarse spatial grids are required to capture the oscillatory features of the KL expansion. Table 3.12 shows the largest value of $\{\theta_k, \theta_k^*\}$ of the eigenfunctions in the KL expansion, the half-wavelength of the functions from (3.29) and our choice of coarse spatial grid refinement levels, ℓ^c , for different values of M . With these coarse grids, there are approximately eight grid points per half wave, enough to capture the qualitative character of the wave.

We turn now to the convection-diffusion equation of Section 5.2. This problem has the same diffusion coefficient (2.7) as the diffusion problem, but in addition its solution has an exponential boundary layer. In particular, for small ν , the width of the layer is smaller than the finest interval needed to represent the eigenfunctions in (2.7), and in this case the coarse grid must be finer than that needed for the diffusion problem (whose solution is smooth). In Figure 3.2, the top plot illustrates the mean solutions $\langle u(x, \xi) \rangle_\rho$ of the weak formulation of (3.23) at $x = 1$, which are computed on two coarse spatial grids $\ell = \{4, 5\}$ using PGD and a fine spatial grid $\ell = 8$

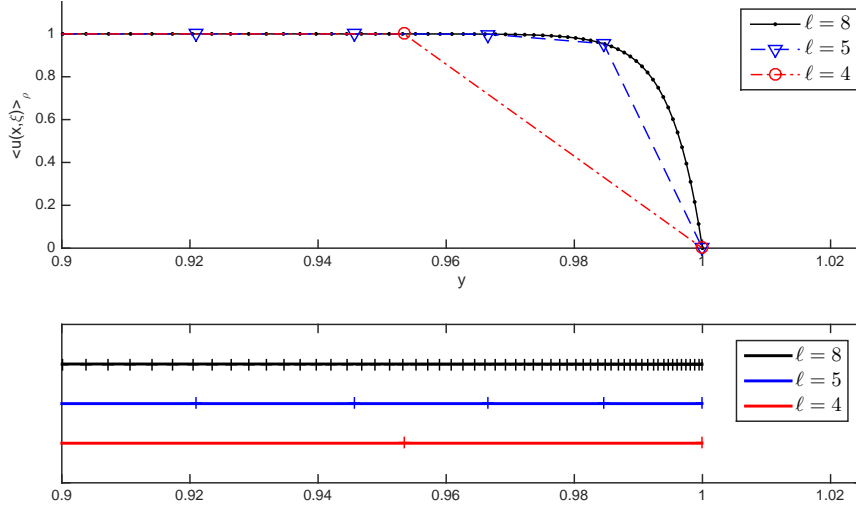


Figure 3.2: Mean solutions $\langle u(x, \xi) \rangle_\rho$ at $x = 1$ and $y = [0.9, 1]$ illustrating the exponential boundary layer for varying spatial grid refinement level, $\ell = \{4, 5, 8\}$, (top) and lengths in y -direction of first few elements from $y = 1$ (bottom).

using the proposed method, with the viscosity parameter, $\nu = \frac{1}{200}$, and $M = 10$ random variables. The bottom plot shows the lengths of the first few elements in the y -direction near $y = 1$ for these refinement levels. If the level-4 spatial grid is used for the coarse grid computation (i.e., $\ell = 4$, red line in Figure 3.2), the width of exponential boundary layer is much narrower than the length of the smallest element and the coarse-grid solution gives a poor representation of the boundary layer. When this coarse grid is used to construct the truncation operator, the proposed scheme fails to compute an accurate approximate solution on a fine spatial grid (i.e., $\ell = 8$, black line in Figure 3.2). On the other hand, the level-5 spatial grid (i.e., $\ell = 5$, blue line in Figure 3.2) is fine enough for the coarse-grid solution to represent the character of the exponential layer, and with this coarse-grid, the resulting proposed scheme efficiently computes an accurate fine-grid solution.

Although this discussion shows that some a priori knowledge of the problem is needed to identify the coarse grid operator, in general this information is not difficult to come by. In particular, we are assuming that the expansion (2.5) is known, and it is straightforward to identify the resolution needed to represent its components, for example by examining one-dimensional cross-sections of them. If as for the second problem some knowledge of the solution is needed, this can be obtained cheaply from the solution of a deterministic problem derived from the mean of the diffusion coefficient; indeed, for the convection-diffusion problem, the boundary layer for the deterministic solution has essentially the same character as that of the stochastic solution whose mean is shown in Figure 3.2.

3.6 Statistical Computations

In this section, we explore the impact of truncation on statistical quantities associated with the solutions. In particular, we examine the mean and the variance of the solution $u_{hp}(x, \xi)$, which are defined as

$$\mu = E[u_{hp}], \quad \sigma_u^2 = E[(u_{hp} - \mu)^2], \quad (3.30)$$

where $E[\cdot] = \int_{\Gamma} \cdot \rho(\xi) d\xi$ refers to the expectation. Let $u_{hp}^{(\text{full})}$ refer to the discrete solution (of form (2.11)) obtained from a full-rank solution of (3.2) (i.e., with no truncation), and let $u_{hp}^{(\text{low})}$ refer to that obtained using Algorithm 3. We will examine the accuracy of $u_{hp}^{(\text{low})}$ by comparing its mean and variance to those of a reference

solution $u_{hp}^{(\text{ref})}$ as follows:

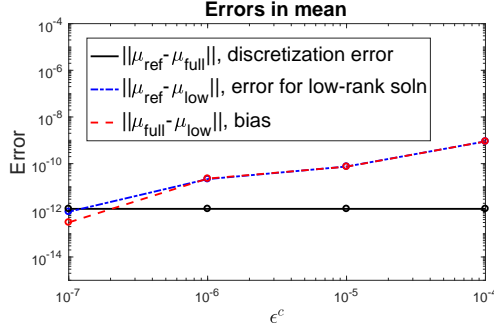
$$\eta_\mu \equiv \|\mu_{\text{ref}} - \mu_{\text{low}}\|_2 \leq \|\mu_{\text{ref}} - \mu_{\text{full}}\|_2 + \|\mu_{\text{full}} - \mu_{\text{low}}\|_2, \quad (3.31)$$

$$\eta_\sigma \equiv \|\sigma_{u,\text{ref}}^2 - \sigma_{u,\text{low}}^2\|_2 \leq \|\sigma_{u,\text{ref}}^2 - \sigma_{u,\text{full}}^2\|_2 + \|\sigma_{u,\text{full}}^2 - \sigma_{u,\text{low}}^2\|_2, \quad (3.32)$$

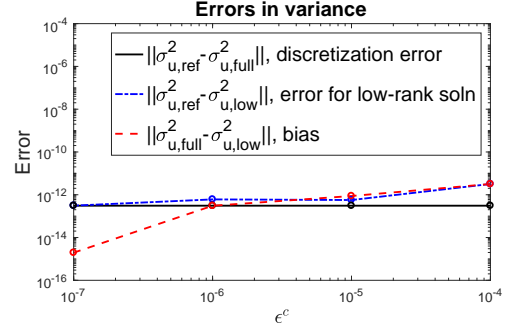
where the norm in (3.31)–(3.32) is the ℓ_2 -norm (e.g., $\|\mu\|_2 = (\int_D \mu(x)^2 dx)^{\frac{1}{2}}$). For these tests, $u_{hp}^{(\text{full})}$ and $u_{hp}^{(\text{low})}$ were computed using a fixed discretization on a spatial grid ($\ell = 7$) and polynomial degree $p = 3$ for the stochastic discretization, and $u_{hp}^{(\text{ref})}$ was computed using the larger polynomial degree $p = 5$.² Thus, for the means in (3.31), $\mu_{\text{ref}} - \mu_{\text{full}}$ represents an approximate to the discretization error and $\mu_{\text{full}} - \mu_{\text{low}}$ is the error caused by the low-rank approximation, which we refer to as the bias. Note that the mean and the variance of the stochastic Galerkin solution (2.11) can be computed easily by exploiting the orthonormality of the basis functions (i.e., for $u(\xi) = \sum_{i=1}^n u_i \psi_i(\xi)$, $\mu = u_1 E[\psi_1] = u_1$ and $\sigma_u^2 = \sum_{i=2}^n u_i^2 E[\psi_i^2] = \sum_{i=2}^n u_i^2$).

Figure 3.3 shows the results for various tolerances ϵ^c and two examples of the diffusion problem (2.1) (with $M = 5$ and $M = 7$ in (2.7)) and one example of the convection-diffusion problem (3.23) with $M = 5$. In all cases, it can be seen that the error for the low-rank solution is somewhat larger than the discretization error for large ϵ^c (and this is caused by the bias), but the bias is significantly smaller than the tolerance ϵ^c . The bias is negligible for $\epsilon^c = 10^{-7}$.

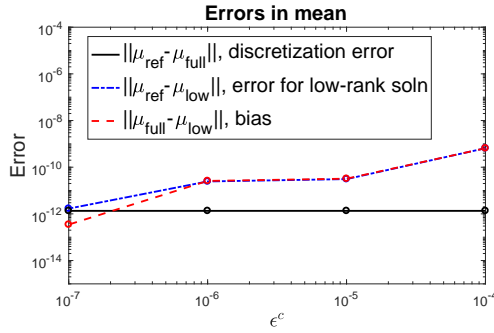
²We also computed a more accurate reference solution with $p = 7$ for the moderate-dimensional problem (i.e., the diffusion problem (2.1) with $M = 5$) and found the results to be virtually identical.



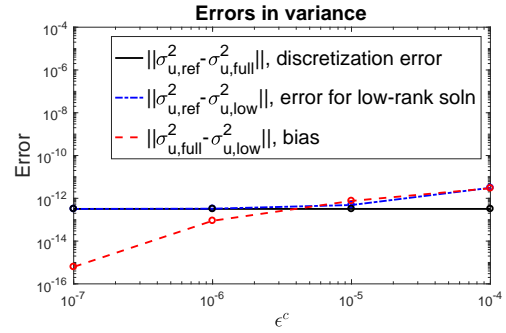
(a) Diffusion with $M = 5$



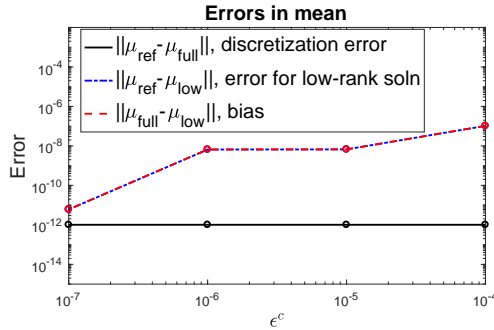
(b) Diffusion with $M = 5$



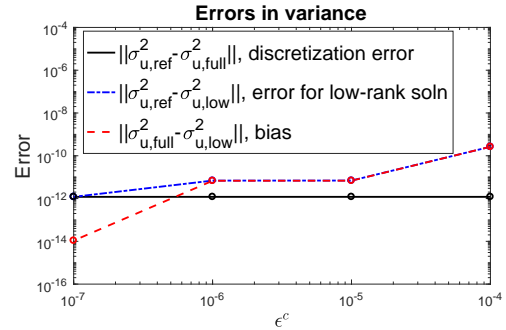
(c) Diffusion with $M = 7$



(d) Diffusion with $M = 7$



(e) Convection-Diffusion with $M = 5$



(f) Convection-Diffusion with $M = 5$

Figure 3.3: Errors in the mean and the variance of the low-rank approximate solutions shown in (3.31) and (3.32) for the stochastic diffusion problem (a)-(d) and the stochastic convection-diffusion problem (e)-(f).

3.7 Conclusion

We have studied iterative solvers for low-rank solutions of stochastic Galerkin systems of stochastic partial differential equations. In particular, we have explored

low-rank projection methods in tensor format for linear systems of Kronecker-product structure. For the computational efficiency of the projection methods, basis vectors and iterates in the projection methods are forced to have low rank, which is achieved by a coarse-grid rank-reduction strategy. We have examined the performance of this strategy with two benchmark problems: stochastic diffusion problems and stochastic convection-diffusion problems. For both problem classes, the rank structure of the solution can be identified by an inexpensive coarse-grid computation, and with the resulting coarse-grid rank-reduction strategy, the low-rank projection method is more efficient than methods for which the truncation operator is based on singular values.

Chapter 4: Low-rank approximation method for parameterized Navier–Stokes equations

4.1 Introduction

In this chapter, we present a low-rank approximation method for the steady-state Navier–Stokes equations with uncertain viscosity. Such uncertainty may arise from measurement error or uncertain ratios of multiple phases in porous media. The uncertain viscosity can be modeled as a positive random field parameterized by a set of random variables [82,99,104] and, consequently, the solution of the stochastic Navier–Stokes equations also can be modeled as a random vector field depending on the parameters associated with the viscosity (i.e., a function of the same set of random variables). As a solution method, we consider the stochastic Galerkin method combined with the generalized polynomial chaos (gPC) expansion, which provides a spectral approximation of the solution function. The stochastic Galerkin method results in a coupled algebraic system of equations, for which computational costs may be high when the global system becomes large.

One way to address this issue is thorough use of tensor *Krylov subspace* methods, which operate in tensor format and reduce the costs of matrix operations by

exploiting a Kronecker-product structure of system matrices. Variants of this approach have been developed for the Richardson iteration [61, 70], the conjugate gradient method [61], the BiCGstab method [61], the minimum residual method [103], and the general minimum residual (GMRES) method [6]. Efficiencies are also obtained from the fact that solutions can often be well approximated by low-rank objects. These ideas have been shown to reduce costs for solving steady [65, 70] and unsteady stochastic diffusion equations [10].

In this study, we adapt the low-rank approximation scheme to a solver for the systems of nonlinear equations obtained from the stochastic Galerkin discretization of the stochastic Navier–Stokes equations. In particular, we consider a low-rank variant of linearization schemes based on Picard and Newton iteration, where the solution of the nonlinear system is computed by solving a sequence of linearized systems using a low-rank variant of the GMRES method (lrGMRES) [6] in combination with inexact nonlinear iteration [30].

We base our development of the stochastic Galerkin formulation of the stochastic Navier–Stokes equations on ideas from [82, 99]. In particular, we consider a random viscosity affinely dependent on a set of random variables as suggested in [82] (and in [99], which considers a gPC approximation of the lognormally distributed viscosity). The stochastic Galerkin formulation of the stochastic Navier–Stokes equations is also considered in [9], which studies an optimal control problem constrained by the stochastic Navier–Stokes problem and computes an approximate solution using a low-rank tensor-train decomposition [77]. Related work [104] extends a Proper Generalized Decomposition method [75] for the stochastic Navier–Stokes equations,

where a low-rank approximate solution is built from successively computing rank-one approximations. See the book [63] for an overview and other spectral approximation approaches for models of computational fluid dynamics.

An outline of the chapter is as follows. In section 4.2, we review the stochastic Navier–Stokes equations and their discrete Galerkin formulations. In section 4.3, we present an iterative low-rank approximation method for solutions of the discretized stochastic Navier–Stokes problems. In section 4.4, we introduce an efficient variant of the inexact Newton method, which solves linear systems arising in nonlinear iteration using low-rank format. We follow a hybrid approach, which employs several steps of Picard iteration followed by Newton iteration. In section 4.5, we examine the performance of the proposed method on a set of benchmark problems that model the flow over an obstacle. Finally, in section 4.6, we draw some conclusions.

4.2 Stochastic Navier–Stokes equations

Consider the stochastic Navier–Stokes equations: Find velocity $\vec{u}(x, \xi)$ and pressure $p(x, \xi)$ such that

$$\begin{aligned}
 -\nu(x, \xi)\nabla^2\vec{u}(x, \xi) + (\vec{u}(x, \xi) \cdot \nabla)\vec{u}(x, \xi) + \nabla p(x, \xi) &= \vec{f}(x, \xi), \\
 \nabla \cdot \vec{u}(x, \xi) &= 0,
 \end{aligned}
 \tag{4.1}$$

in $D \times \Gamma$, with a boundary conditions

$$\begin{aligned} \vec{u}(x, \xi) &= \vec{g}(x, \xi), & \text{on } \partial D_{\text{Dir}}, \\ \nu(x, \xi) \nabla \vec{u}(x, \xi) \cdot \vec{n} - p(x, \xi) \vec{n}(x, \xi) &= \vec{0}, & \text{on } \partial D_{\text{Neu}}, \end{aligned}$$

where $\partial D = \partial D_{\text{Dir}} \cup \partial D_{\text{Neu}}$. The stochasticity of the equation (4.1) stems from the random viscosity $\nu(x, \xi)$, which is modeled as a positive random field parameterized by a set of independent, identically distributed random variables $\xi = \{\xi_1, \dots, \xi_{n_\nu}\}$. The random variables comprising ξ are defined on a probability space (Ω, \mathcal{F}, P) such that $\xi : \Omega \rightarrow \Gamma \subset \mathbb{R}^{n_\nu}$, where Ω is a sample space, \mathcal{F} is a σ -algebra on Ω , and P is a probability measure on Ω . The joint probability density function of ξ is denoted by $\rho(\xi)$ and the expected value of a random function $v(\xi)$ on Γ is then $\langle v \rangle_\rho = \mathbb{E}[v] \equiv \int_\Gamma v(\xi) \rho(\xi) d\xi$.

For the random viscosity, we consider a random field that has affine dependence on the random variables ξ ,

$$\nu(x, \xi) \equiv \nu_0 + \sigma_\nu \sum_{k=1}^{n_\nu} \nu_k(x) \xi_k, \quad (4.2)$$

where $\{\nu_0, \sigma_\nu^2\}$ are the mean and the variance of the random field $\nu(x, \xi)$. We will also refer to the coefficient of variation (CoV), the relative size of the standard deviation with respect to the mean,

$$CoV \equiv \frac{\sigma_\nu}{\nu_0}. \quad (4.3)$$

The random viscosity leads to the random Reynolds number

$$\text{Re}(\xi) \equiv \frac{UL}{\nu(\xi)}, \quad (4.4)$$

where U is the characteristic velocity and L is the characteristic length. We denote the Reynolds number associated with the mean viscosity by $\text{Re}_0 = \frac{UL}{\nu_0}$. In this study, we ensure that the viscosity (4.2) has positive values by controlling CoV and only consider small enough Re_0 so that the flow problem has a unique solution.

4.2.1 Stochastic Galerkin method

In the stochastic Galerkin method, a mixed variational formulation of (4.1) can be obtained by employing Galerkin orthogonality: Find $(\vec{u}, p) \in (V_E, Q_D) \otimes L^2(\Gamma)$ such that

$$\left\langle \int_D \nu \nabla \vec{u} : \nabla \vec{v} + (\vec{u} \cdot \nabla \vec{u}) \vec{v} - p(\nabla \cdot \vec{v}) \right\rangle_\rho = \left\langle \int_D \vec{f} \cdot \vec{v} \right\rangle_\rho, \quad \forall \vec{v} \in V_D \otimes L^2(\Gamma), \quad (4.5)$$

$$\left\langle \int_D q(\nabla \cdot \vec{u}) \right\rangle_\rho = 0, \quad \forall q \in Q_D \otimes L^2(\Gamma). \quad (4.6)$$

The velocity solution and test spaces are $V_E = \{\vec{u} \in \mathcal{H}^1(D)^2 | \vec{u} = \vec{g} \text{ on } \partial D_{\text{Dir}}\}$ and $V_D = \{\vec{v} \in \mathcal{H}^1(D)^2 | \vec{v} = \vec{0} \text{ on } \partial D_{\text{Dir}}\}$, where $\mathcal{H}^1(D)$ refers to the Sobolev space of functions with derivatives in $L^2(D)$, for the pressure solution, $Q_D = L^2(D)$, and $L^2(\Gamma)$ is a Hilbert space equipped with an inner product

$$\langle u, v \rangle_\rho \equiv \int_\Gamma u(\xi)v(\xi)\rho(\xi)d\xi.$$

The solution of the variational formulation (4.5)–(4.6) satisfies

$$\mathcal{R}(\vec{u}, p; \vec{v}, q) = 0, \quad \forall \vec{v} \in V_D \otimes L^2(\Gamma), \forall q \in Q_D \otimes L^2(\Gamma), \quad (4.7)$$

where $\mathcal{R}(\vec{u}, p; \vec{v}, q)$ is a nonlinear residual

$$\mathcal{R}(\vec{u}, p; \vec{v}, q) \equiv \begin{bmatrix} \langle \int_D \vec{f} \cdot \vec{v} - \nu \nabla \vec{u} : \nabla \vec{v} + (\vec{u} \cdot \nabla \vec{u}) \vec{v} - \int_D p(\nabla \cdot \vec{v}) \rangle_\rho \\ \langle - \int_D q(\nabla \cdot \vec{u}) \rangle_\rho \end{bmatrix}. \quad (4.8)$$

To compute the solution of the nonlinear equation (4.7), we employ linearization techniques based on either Picard iteration or Newton iteration [39]. Replacing (\vec{u}, p) of (4.5)–(4.6) with $(\vec{u} + \delta \vec{u}, p + \delta p)$ and neglecting the quadratic term $c(\delta \vec{u}; \delta \vec{u}, \vec{v})$, where $c(\vec{z}; \vec{u}, \vec{v}) \equiv \int_D (\vec{z} \cdot \nabla \vec{u}) \cdot \vec{v}$, gives

$$\begin{bmatrix} \langle \int_D \nu \nabla \delta \vec{u} : \nabla \vec{v} + c(\delta \vec{u}; \vec{u}, \vec{v}) + c(\vec{u}; \delta \vec{u}, \vec{v}) - \int_D \delta p(\nabla \cdot \vec{v}) \rangle_\rho \\ \langle \int_D q(\nabla \cdot \delta \vec{u}) \rangle_\rho \end{bmatrix} = \mathcal{R}(\vec{u}, p; \vec{v}, q). \quad (4.9)$$

In Newton iteration, the $(n+1)$ st iterate (\vec{u}^{n+1}, p^{n+1}) is computed by taking $\vec{u} = \vec{u}^n$, $p = p^n$ in (4.9), solving (4.9) for $(\delta \vec{u}^n, \delta p^n)$, and updating

$$\vec{u}^{n+1} := \vec{u}^n + \delta \vec{u}^n, \quad p^{n+1} := p^n + \delta p^n.$$

In Picard iteration, the term $c(\delta \vec{u}; \vec{u}, \vec{v})$ is omitted from the linearized form (4.9).

4.2.2 Discrete stochastic Galerkin system

To obtain a discrete system, the velocity $\vec{u}(x, \xi)$ and the pressure $p(x, \xi)$ are approximated by a generalized polynomial chaos expansion [112]:

$$\vec{u}(x, \xi) \equiv \sum_{i=1}^{n_\xi} \vec{u}_i(x) \psi_i(\xi), \quad p(x, \xi) \equiv \sum_{i=1}^{n_\xi} p_i(x) \psi_i(\xi), \quad (4.10)$$

where $\{\psi_i(\xi)\}_{i=1}^{n_\xi}$ is a set of n_ν -variate orthogonal polynomials (i.e., $\langle \psi_i \psi_j \rangle_\rho = 0$ if $i \neq j$). This set of orthogonal polynomials gives rise to a finite-dimensional approximation space $S = \text{span}(\{\psi_i(\xi)\}_{i=1}^{n_\xi}) \subset L^2(\Gamma)$. For spatial discretization, a div-stable mixed finite element method [39] is considered, the Taylor-Hood element consisting of biquadratic velocities and bilinear pressure. Basis sets for the velocity space V_E^h and the pressure space Q_D^h are denoted by $\left\{ \begin{bmatrix} \phi_i(x) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \phi_i(x) \end{bmatrix} \right\}_{i=1}^{n_u}$ and $\{\varphi_i(x)\}_{i=1}^{n_p}$, respectively. Then the fully discrete version of (4.10) can be written as

$$\vec{u}(x, \xi) = \begin{bmatrix} \vec{u}^x(x, \xi) \\ \vec{u}^y(x, \xi) \end{bmatrix} \equiv \begin{bmatrix} \sum_{i=1}^{n_\xi} \sum_{j=1}^{n_u} u_{ij}^x \phi_j(x) \psi_i(\xi) \\ \sum_{i=1}^{n_\xi} \sum_{j=1}^{n_u} u_{ij}^y \phi_j(x) \psi_i(\xi) \end{bmatrix}, \quad p(x, \xi) \equiv \sum_{i=1}^{n_\xi} \sum_{j=1}^{n_p} p_{ij} \varphi_j(x) \psi_i(\xi). \quad (4.11)$$

Let us introduce a vector notation for the coefficients, $\bar{u}_i^x \equiv [u_{i1}^x, \dots, u_{in_u}^x]^T \in \mathbb{R}^{n_u}$, $\bar{u}_i^y \equiv [u_{i1}^y, \dots, u_{in_u}^y]^T \in \mathbb{R}^{n_u}$, and $\bar{p}_i \equiv [p_{i1}, \dots, p_{in_p}]^T \in \mathbb{R}^{n_p}$ for $i = 1, \dots, n_\xi$, which, for each gPC index i , groups the horizontal velocity coefficients together followed by the vertical velocity coefficients, and then by the pressure coefficients, giving a

vector

$$\bar{u}_i = [(\bar{u}_i^x)^T, (\bar{u}_i^y)^T, p_i^T]^T. \quad (4.12)$$

Taking $\nu(x, \xi)$ from (4.2) and replacing $\bar{u}(x, \xi)$, $p(x, \xi)$ in (4.9) with their discrete approximations (4.11) yields a system of linear equations of order $(2n_u + n_p)n_\xi$. The coefficient matrix has a Kronecker-product structure,

$$J \equiv G_1 \otimes \mathcal{F}_1 + \sum_{l=2}^{n_\xi} G_l \otimes \mathcal{F}_l, \quad (4.13)$$

where G_l refers to the l th “stochastic matrix”

$$[G_l]_{ij} = \langle \psi_l \psi_i \psi_j \rangle_\rho, \quad l = 1, \dots, n_\xi$$

with $\psi_1(\xi) = 1$, $\psi_i(\xi) = \xi_{i-1}$ for $i = 2, \dots, n_\nu + 1$ and

$$\mathcal{F}_1 \equiv \begin{bmatrix} F_1 & B^T \\ B & 0 \end{bmatrix}, \quad \mathcal{F}_l \equiv \begin{bmatrix} F_l & 0 \\ 0 & 0 \end{bmatrix}, \quad l = 2, \dots, n_\xi$$

with $F_l \equiv A_l + N_l + W_l$ for the Newton iteration and $F_l \equiv A_l + N_l$ for the Picard iteration. We refer to the matrix of (4.13) derived from the Newton iteration as the *Jacobian* matrix, and that derived from the Picard iteration as the *Oseen* matrix, denoted by J_N and J_P , respectively. Here, A_l is the l th symmetric matrix defined

as

$$[A_l]_{ij} \equiv \int_D \nu_{l-1}(x) (\nabla \phi_i : \nabla \phi_j), \quad l = 1, \dots, n_\nu + 1, \quad (4.14)$$

$N_l = N(\vec{u}_l(x))$ and $W_l = W(\vec{u}_l(x))$ are, respectively, the l th vector-convection matrix and the l th Newton derivative matrix with $\vec{u}_l^n(x)$ from the l th term of (4.10),

$$\begin{aligned} [N_l]_{ij} &= [N(\vec{u}_l(x))]_{ij} \equiv \int_D (\vec{u}_l(x) \cdot \nabla \phi_j(x)) \cdot \phi_i(x), \quad l = 1, \dots, n_\xi, \\ [W_l]_{ij} &= [W(\vec{u}_l(x))]_{ij} \equiv \int_D (\phi_j(x) \cdot \nabla \vec{u}_l(x)) \cdot \phi_i(x), \quad l = 1, \dots, n_\xi, \end{aligned}$$

and B is the divergence matrix,

$$[B]_{ij} \equiv \int_D \varphi_j(\nabla \cdot \phi_i). \quad (4.15)$$

If the number of gPC polynomial terms in (4.11) is larger than the number of terms in (4.2) (i.e., $n_\xi > n_\nu + 1$), we simply set $\{A_l\}_{l=n_\nu+2}^{n_\xi}$ as matrices containing only zeros so that $\mathcal{F}_l = N_l + W_l$ for $l = n_\nu + 2, \dots, n_\xi$.

A discrete version of (4.8) can be derived in a similar way,

$$\bar{r} := \bar{y} - \left(G_1 \otimes \mathcal{P}_1 + \sum_{l=2}^{n_\xi} G_l \otimes \mathcal{P}_l \right) \bar{u} \quad (4.16)$$

where $\bar{u} := [\bar{u}_1^T \dots \bar{u}_{n_\xi}^T]^T \in \mathbb{R}^{(2n_u+n_p)n_\xi}$ with \bar{u}_i as in (4.12), \bar{y} is the right-hand side

determined from the forcing function and Dirichlet boundary data, and

$$\mathcal{P}_1 \equiv \begin{bmatrix} A_1 + N_1 & B^T \\ B & 0 \end{bmatrix}, \quad \mathcal{P}_l \equiv \begin{bmatrix} A_l + N_l & 0 \\ 0 & 0 \end{bmatrix} \quad l = 2, \dots, n_\xi.$$

The system of linear equations arising at the n th nonlinear iteration is

$$J^n \delta \bar{u}^n = -\bar{r}^n, \tag{4.17}$$

where the matrix J^n from (4.13) and the residual \bar{r}^n from (4.16) each evaluated at the n th iterate \bar{u}^n , and the update $\delta \bar{u}^n$ is computed by solving (4.17). The order of the system $(2n_u + n_p)n_\xi$ grows fast as the number of random variables used to parameterize the random viscosity increases. Even for a moderate-dimensional stochastic Navier–Stokes problem, solving a sequence of linear systems of order $(2n_u + n_p)n_\xi$ can be computationally prohibitive. To address this issue, we present an efficient variant of Newton–Krylov methods in the following sections.

4.3 Low-rank Newton–Krylov method

In this section, we outline the formalism in which the solutions to (4.16) and (4.17) can be efficiently approximated by low-rank objects while not losing much accuracy and we show how solvers are adjusted within this formalism.

Before presenting these ideas, we describe the nonlinear iteration. We consider a hybrid strategy. An initial approximation for the nonlinear solution is computed

by solving the parameterized Stokes equations,

$$\begin{aligned} -\nu(x, \xi) \nabla^2 \vec{u}(x, \xi) + \nabla p(x, \xi) &= \vec{f}(x, \xi), \\ \nabla \cdot \vec{u}(x, \xi) &= 0. \end{aligned}$$

The discrete Stokes operator, which is obtained from the stochastic Galerkin discretization as shown in Section 4.2.2, is

$$\left(G_1 \otimes \mathcal{S}_1 + \sum_{l=2}^{n_\nu+1} G_l \otimes \mathcal{S}_l \right) \bar{u}_{\text{st}} = b_{\text{st}}, \quad (4.18)$$

where

$$\mathcal{S}_1 = \begin{bmatrix} A_1 & B^T \\ B & 0 \end{bmatrix}, \quad \mathcal{S}_l = \begin{bmatrix} A_l & 0 \\ 0 & 0 \end{bmatrix}, \quad l = 2, \dots, n_\nu + 1,$$

with $\{A_l\}_{l=1}^{n_\nu+1}$ defined in (4.14) and B defined in (4.15). After this initial computation, updates to the solution are computed by first solving m_p Picard systems with coefficient matrix J_P and then using Newton's method with coefficient matrix J_N to compute the solution.

Algorithm 4 Solution methods

- 1: compute an approximate solution of $A_{\text{st}} \bar{u}_{\text{st}} = b_{\text{st}}$ in (4.18)
 - 2: set an initial guess for the Navier–Stokes problem $\bar{u}^0 := \bar{u}_{\text{st}}$
 - 3: **for** $k = 0, \dots, m_p - 1$ **do**
 - 4: solve $J_P^k \delta \bar{u}^k = -\bar{r}^k$
 - 5: update $\bar{u}^{k+1} := \bar{u}^k + \delta \bar{u}^k$
 - 6: **end for**
 - 7: **while** $k < m_n$ and $\|\bar{r}^k\|_2 > \epsilon_{\text{nl}} \|\bar{r}^0\|_2$ **do**
 - 8: solve $J_N^k \delta \bar{u}^k = -\bar{r}^k$
 - 9: update $\bar{u}^{k+1} := \bar{u}^k + \delta \bar{u}^k$
 - 10: **end while**
-

4.3.1 Approximation in low rank

We now develop a low-rank variant of Algorithm 4. Let us begin by introducing some concepts to define the rank of computed quantities. Let $X = [\bar{x}_1, \dots, \bar{x}_{n_2}] \in \mathbb{R}^{n_1 \times n_2}$ and $\bar{x} = [\bar{x}_1^T, \dots, \bar{x}_{n_2}^T]^T \in \mathbb{R}^{n_1 n_2}$, where $\bar{x}_i \in \mathbb{R}^{n_1}$ for $i = 1, \dots, n_2$. That is, \bar{x} can be constructed by rearranging the elements of X , and vice versa. Suppose X has rank α_x . Then two mathematically equivalent expressions for X and \bar{x} are given by

$$X = YZ^T = \sum_{i=1}^{\alpha_{\bar{x}}} \bar{y}_i \bar{z}_i^T \quad \Leftrightarrow \quad \bar{x} = \sum_{i=1}^{\alpha_{\bar{x}}} \bar{z}_i \otimes \bar{y}_i, \quad (4.19)$$

where $Y \equiv [\bar{y}_1, \dots, \bar{y}_{\alpha_{\bar{x}}}] \in \mathbb{R}^{n_1 \times \alpha_{\bar{x}}}$, $Z \equiv [\bar{z}_1, \dots, \bar{z}_{\alpha_{\bar{x}}}] \in \mathbb{R}^{n_2 \times \alpha_{\bar{x}}}$ with $\bar{y}_i \in \mathbb{R}^{n_1}$, $\bar{z}_i \in \mathbb{R}^{n_2}$ for $i = 1, \dots, \alpha_{\bar{x}}$. The representation of X and its rank is standard matrix notation; we also use α_x to refer to the rank of the corresponding vector \bar{x} .

With this definition of rank, our goal is to inexpensively find a low-rank approximate solution \bar{u}^k satisfying $\|\bar{r}^k\|_2 \leq \epsilon_{\text{nl}} \|\bar{r}^0\|_2$ for small enough ϵ_{nl} . To achieve this goal, we approximate updates $\{\delta \bar{u}^k\}$ in low-rank using a low-rank variant of GMRES method, which exploits the Kronecker product structure in the system matrix as in (4.13) and (4.18). In the following section, we present the solutions \bar{u} (and $\delta \bar{u}$) in the formats of (4.19) together with matrix and vector operations that are essential for developing the low-rank GMRES method.

4.3.2 Solution coefficients in Kronecker-product form

We seek separate low-rank approximations of the horizontal and vertical velocity solutions and the pressure solution. With the representation shown in (4.19), the solution coefficient vector $\bar{u} \in \mathbb{R}^{(2n_u+n_p)n_\xi}$, which consists of the coefficients of the velocity solution and the pressure solution (4.11), has an equivalent representation $U \in \mathbb{R}^{(2n_u+n_p) \times n_\xi}$. The matricized solution coefficients $U = [U^x, U^y, P]^T$ where $U^x = [\bar{u}_1^x, \dots, \bar{u}_{n_\xi}^x]$, $U^y = [\bar{u}_1^y, \dots, \bar{u}_{n_\xi}^y] \in \mathbb{R}^{n_u \times n_\xi}$ and the pressure solution $P = [\bar{p}_1, \dots, \bar{p}_{n_\xi}] \in \mathbb{R}^{n_p \times n_\xi}$. The components admit the following representations:

$$U^x = \sum_{i=1}^{\alpha_{\bar{u}^x}} \bar{v}_i^x (\bar{w}_i^x)^T = V^x (W^x)^T \Leftrightarrow \bar{u}^x = \sum_{i=1}^{\alpha_{\bar{u}^x}} \bar{w}_i^x \otimes \bar{v}_i^x, \quad (4.20)$$

$$U^y = \sum_{i=1}^{\alpha_{\bar{u}^y}} \bar{v}_i^y (\bar{w}_i^y)^T = V^y (W^y)^T \Leftrightarrow \bar{u}^y = \sum_{i=1}^{\alpha_{\bar{u}^y}} \bar{w}_i^y \otimes \bar{v}_i^y, \quad (4.21)$$

$$P = \sum_{i=1}^{\alpha_{\bar{p}}} \bar{v}_i^p (\bar{w}_i^p)^T = V^p (W^p)^T \Leftrightarrow \bar{p} = \sum_{i=1}^{\alpha_{\bar{p}}} \bar{w}_i^p \otimes \bar{v}_i^p, \quad (4.22)$$

where $V^x = [\bar{v}_1^x \dots \bar{v}_{\alpha_{\bar{u}^x}}^x]$, $W^x = [\bar{w}_1^x \dots \bar{w}_{\alpha_{\bar{u}^x}}^x]$, $\alpha_{\bar{u}^x}$ is the rank of \bar{u}^x and U^x , and the same interpretation can be applied to \bar{u}^y and \bar{p} .

4.3.2.1 Matrix operations

In this section, we introduce essential matrix operations used by the low-rank GMRES methods, using the representations shown in (4.20)–(4.22). First, consider the matrix-vector product with the Jacobian system matrix (4.13) and

vectors (4.20)–(4.22),

$$J^n \bar{u}^n = \left(\sum_{l=1}^{n_\xi} G_l \otimes \mathcal{F}_l^n \right) \bar{u}^n, \quad (4.23)$$

where

$$\mathcal{F}_l^n = \begin{bmatrix} A_l^{xx} + N_l^n + W_l^{xx,n} & W_l^{xy,n} & B^{xT} \\ W_l^{yx,n} & A_l^{yy} + N_l^n + W_l^{yy,n} & B^{yT} \\ B^x & B^y & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{F}_l^{xx,n} & \mathcal{F}_l^{xy,n} & B^{xT} \\ \mathcal{F}_l^{yx,n} & \mathcal{F}_l^{yy,n} & B^{yT} \\ B^x & B^y & 0 \end{bmatrix}$$

with $\mathcal{F}_l^{xx,n}, \mathcal{F}_l^{xy,n}, \mathcal{F}_l^{yx,n}, \mathcal{F}_l^{yy,n} \in \mathbb{R}^{n_u \times n_u}$ and $B^x, B^y \in \mathbb{R}^{n_p \times n_u}$. The expression (4.23)

has the equivalent matricized form $\sum_{l=1}^{n_\xi} \mathcal{F}_l^n U^n G_l^T$ where the l th-term is evaluated

as

$$\mathcal{F}_l^n U^n G_l^T = \begin{bmatrix} \mathcal{F}_l^{xx,n} V^{x,n} (G_l W^{x,n})^T + \mathcal{F}_l^{xy,n} V^{y,n} (G_l W^{y,n})^T + B^{xT} V^{p,n} (G_l W^{p,n})^T \\ \mathcal{F}_l^{yx,n} V^{x,n} (G_l W^{x,n})^T + \mathcal{F}_l^{yy,n} V^{y,n} (G_l W^{y,n})^T + B^{yT} V^{p,n} (G_l W^{p,n})^T \\ B^x V^{x,n} (G_l W^{x,n})^T + B^y V^{y,n} (G_l W^{y,n})^T \end{bmatrix}. \quad (4.24)$$

Equivalently, in the Kronecker-product structure, the matrix-vector product (4.24)

updates each set of solution coefficients as follows:

$$\sum_{l=1}^{n_\xi} (G_l \otimes \mathcal{F}_l^{xx,n}) \bar{u}^{x,n} + (G_l \otimes \mathcal{F}_l^{xy,n}) \bar{u}^{y,n} + (G_l \otimes B^{xT}) \bar{p}^n, \quad (x\text{-velocity}), \quad (4.25)$$

$$\sum_{l=1}^{n_\xi} (G_l \otimes \mathcal{F}_l^{yx,n}) \bar{u}^{x,n} + (G_l \otimes \mathcal{F}_l^{yy,n}) \bar{u}^{y,n} + (G_l \otimes B^{yT}) \bar{p}^n, \quad (y\text{-velocity}) \quad (4.26)$$

$$\sum_{l=1}^{n_\xi} (G_l \otimes B^x) \bar{u}^{x,n} + (G_l \otimes B^y) \bar{u}^{y,n}, \quad (\text{pressure}) \quad (4.27)$$

where each matrix-vector product can be performed by exploiting the Kronecker-product structure, for example,

$$\sum_{l=1}^{n_\xi} (G_l \otimes \mathcal{F}_l^{xx,n}) \bar{u}^{x,n} = \sum_{l=1}^{n_\xi} G_l \otimes \mathcal{F}_l^{xx,n} \sum_{i=1}^{\alpha_{\bar{u}^x}} w_i^x \otimes v_i^x = \sum_{l=1}^{n_\xi} \sum_{i=1}^{\alpha_{\bar{u}^x}} G_l w_i^x \otimes \mathcal{F}_l^{xx,n} v_i^x. \quad (4.28)$$

The matrix-vector product shown in (4.25)–(4.27) requires $O(2n_u + n_p + n_\xi)$ flops, whereas (4.23) requires $O((2n_u + n_p)n_\xi)$ flops. Thus, as the problem size grows, the additive form of the latter count grows much less rapidly than the multiplicative form for (4.23).

The addition of two vectors \bar{u}^x and \bar{u}^y can also be efficiently performed in the Kronecker-product structure,

$$\bar{u}^x + \bar{u}^y = \sum_{i=1}^{\alpha_{\bar{u}^x}} w_i^x \otimes v_i^x + \sum_{i=1}^{\alpha_{\bar{u}^y}} w_i^y \otimes v_i^y = \sum_{i=1}^{\alpha_{\bar{u}^x} + \alpha_{\bar{u}^y}} \hat{v}_i + \hat{w}_i, \quad (4.29)$$

where $\hat{v}_i = v_i^x$, $\hat{w}_i = w_i^x$ for $i = 1, \dots, \alpha_{\bar{u}^x}$, and $\hat{v}_i = v_i^y$, $\hat{w}_i = w_i^y$ for $i = \alpha_{\bar{u}^x} + 1, \dots, \alpha_{\bar{u}^x} + \alpha_{\bar{u}^y}$.

$1, \dots, \alpha_{\bar{u}^x} + \alpha_{\bar{u}^y}$.

Inner products can be performed with similar efficiencies. Consider two vectors \bar{x}_1 and \bar{x}_2 , whose matricized representations are

$$X_1 = \begin{bmatrix} Y_{11}Z_{11}^T \\ Y_{12}Z_{12}^T \\ Y_{13}Z_{13}^T \end{bmatrix}, \quad X_2 = \begin{bmatrix} Y_{21}Z_{21}^T \\ Y_{22}Z_{22}^T \\ Y_{23}Z_{23}^T \end{bmatrix}. \quad (4.30)$$

Then the Euclidean inner product between x_1 and x_2 can be evaluated as

$$\bar{x}_1^T \bar{x}_2 = \text{trace}((Y_{11}Z_{11}^T)^T Y_{21}Z_{21}^T) + \text{trace}((Y_{12}Z_{12}^T)^T Y_{22}Z_{22}^T) + \text{trace}((Y_{13}Z_{13}^T)^T Y_{23}Z_{23}^T),$$

where $\text{trace}(X)$ is defined as a sum of the diagonal entries of the matrix X .

Although the matrix-vector product and the sum, as described in (4.28) and (4.29), can be performed efficiently, the results of (4.28) and (4.29) are represented by $n_\xi \alpha_{\bar{u}^x}$ terms and $\alpha_{\bar{u}^x} + \alpha_{\bar{u}^y}$ terms, respectively, which typically causes the ranks of the computed quantities to be higher than the inputs for the computations and potentially undermines the efficiency of the solution method. To resolve this issue, a truncation operator will be used to modify the result of matrix-vector products and sums and force the ranks of quantities used to be small.

4.3.2.2 Truncation of $U^{x,n}$, $U^{y,n}$ and P^n

We now explain the details of the truncation. Consider the velocity and the pressure represented in a matrix form as in (4.20)–(4.22). The best α -rank ap-

proximation of a matrix can be found by using the singular value decomposition (SVD) [61, 70]. Here, we define a truncation operator for a given matrix $U = VW^T$ whose rank is α_U ,

$$\mathcal{T}_{\epsilon_{\text{trunc}}} : U \rightarrow \tilde{U},$$

where the rank of U is larger than the rank of \tilde{U} (i.e., $\alpha_U \gg \alpha_{\tilde{U}}$). The truncation operator $\mathcal{T}_{\epsilon_{\text{trunc}}}$ compresses U to \tilde{U} such that $\|\tilde{U} - U\|_F \leq \epsilon_{\text{trunc}}\|U\|_F$ where $\|\cdot\|_F$ is the Frobenius norm. To achieve this goal, the singular value decomposition of U can be computed (i.e., $U = \hat{V}D\tilde{W}^T$ where $D = \text{diag}(d_1, \dots, d_n)$ is the diagonal matrix of singular values). Letting $\{\hat{v}_i\}$ and $\{\tilde{w}_i\}$ denote the singular vectors, the approximation is $\tilde{U} = \sum_{i=1}^{\alpha_{\tilde{U}}} \tilde{v}_i \tilde{w}_i^T$ with $\tilde{v}_i = d_i \hat{v}_i$ and the truncation rank $\alpha_{\tilde{U}}$ is determined by the condition

$$\sqrt{d_{\alpha_{\tilde{U}}+1}^2 + \dots + d_n^2} \leq \epsilon_{\text{trunc}} \sqrt{d_1^2 + \dots + d_n^2}. \quad (4.31)$$

4.3.3 Low-rank GMRES method

We describe the low-rank GMRES method (lrGMRES) with a generic linear system $Ax = b$. The method follows the standard Arnoldi iteration used by GMRES [92]: construct a set of basis vectors $\{v_i\}_{i=1}^{m_{\text{gm}}}$ by applying the linear operator A to basis vectors, i.e., $w_j = Av_j$ for $j = 1, \dots, m_{\text{gm}}$, and orthogonalizing the resulting vector w_j with respect to previously generated basis vectors $\{v_i\}_{i=1}^{j-1}$. In the low-rank GMRES method, iterates, basis vectors $\{v_i\}$ and intermediate quantities $\{w_i\}$ are

represented in terms of the factors of their matricized representations (so that X in (4.19) would be represented using Y and Z without explicit construction of X), and matrix operations such as matrix-vector products are performed as described in Section 4.3.2.1. As pointed out in Section 4.3.2.1, these matrix operations typically tend to increase the rank of the resulting quantity, and this is resolved by interleaving the truncation operator \mathcal{T} with the matrix operations. The low-rank GMRES method computes a new iterate by solving

$$\min_{\beta \in \mathbb{R}^{m_{\text{gm}}}} \|b - A(x_0 + V_{m_{\text{gm}}}\bar{\beta})\|_2, \quad (4.32)$$

and constructing a new iterate $x_1 = x_0 + V_{m_{\text{gm}}}\bar{\beta}$ where x_0 is an initial guess. Due to truncation, the basis vectors $\{v_i\}$ are not orthogonal and $\text{span}(V_{m_{\text{gm}}})$, where $V_{m_{\text{gm}}} = [v_1 \dots v_{m_{\text{gm}}}]$, is not a Krylov subspace, so that (4.32) must be solved explicitly rather than exploiting Hessenberg structure as in standard GMRES. Algorithm 5 summarizes the lrGMRES. We will use this method to solve the linear system of (4.17).

4.3.4 Preconditioning

We also use preconditioning to speed convergence of the low-rank GMRES method. For this, we consider a right-preconditioned system

$$J^n(M^n)^{-1}\tilde{u}^n = \tilde{r}^n,$$

Algorithm 5 Restarted low-rank GMRES method in tensor format

```

1: set the initial solution  $\bar{u}_{\text{gm}}^0$ 
2: for  $k = 0, 1, \dots$  do
3:    $r_{\text{gm}}^k := f - A\bar{u}_{\text{gm}}^k$ 
4:   if  $\|r_{\text{gm}}^k\|_2/\|f\|_2 < \epsilon_{\text{gmres}}$  or  $\|r_{\text{gm}}^k\|_2 \geq \|r_{\text{gm}}^{k-1}\|_2$  then
5:     return  $\bar{u}_{\text{gm}}^k$ 
6:   end if
7:    $\bar{v}_1 := \mathcal{T}_{\epsilon_{\text{trunc}}}(r_{\text{gm}}^k)$ 
8:    $v_1 := \bar{v}_1/\|\bar{v}_1\|_2$ 
9:   for  $j = 1, \dots, m_{\text{gm}}$  do
10:     $w_j := Av_j$ 
11:    solve  $(V_j^T V_j)\bar{\alpha} = V_j^T w_j$  where  $V_j = [v_1, \dots, v_j]$ 
12:     $\bar{v}_{j+1} := \mathcal{T}_{\epsilon_{\text{trunc}}}\left(w_j - \sum_{i=1}^j \alpha_i v_i\right)$ 
13:     $v_{j+1} := \bar{v}_{j+1}/\|\bar{v}_{j+1}\|_2$ 
14:   end for
15:   solve  $(W_{m_{\text{gm}}}^T AV_{m_{\text{gm}}})\bar{\beta} = W_{m_{\text{gm}}}^T r_{\text{gm}}^k$  where  $W_j = [w_1, \dots, w_j]$ 
16:    $\bar{u}_{\text{gm}}^{k+1} := \mathcal{T}_{\epsilon_{\text{trunc}}}(\bar{u}_{\text{gm}}^k + V_{m_{\text{gm}}}\bar{\beta})$ 
17: end for

```

where M^n is the preconditioner and $M^n \bar{u}^n = \tilde{u}^n$ such that $J^n \bar{u}^n = \bar{r}^n$. We consider an approximate mean-based preconditioner [81], which is derived from the matrix $G_1 \otimes \mathcal{F}_1$ associated with the mean ν_0 of the random viscosity (4.2),

$$M^n = G_1 \otimes \begin{bmatrix} M_A^n & B^T \\ 0 & -M_s^n \end{bmatrix}, \quad (4.33)$$

where

$$M_A^n = \begin{bmatrix} A_1^{xx} + N_1^n & 0 \\ 0 & A_1^{yy} + N_1^n \end{bmatrix}, \quad (\text{Picard iteration}),$$

$$M_A^n = \begin{bmatrix} A_1^{xx} + N_1^n + W_1^{xx,n} & 0 \\ 0 & A_1^{yy} + N_1^n + W_1^{yy,n} \end{bmatrix}, \quad (\text{Newton iteration}).$$

For approximating the action of the inverse, $(M_s^n)^{-1}$, we choose the boundary-adjusted least-squares commutator (LSC) preconditioning scheme [39],

$$M_s^n = BF_1^{-1}B^T \approx (BH^{-1}B^T)(BM_*^{-1}F_1H^{-1}B^T)^{-1}(BM_*^{-1}B^T),$$

where M_* is the diagonal of the velocity mass matrix and $H = D^{-1/2}M_*D^{-1/2}$, where D is a diagonal scaling matrix deemphasizing contributions near the boundary. During the low-rank GMRES iteration, the action of the inverse of the preconditioner (4.33) can be applied to a vector in a manner analogous to (4.25)–(4.27).

4.4 Inexact nonlinear iteration

As outlined in Algorithm 4, we use the hybrid approach, employing a few steps of Picard iteration followed by Newton iteration, and the linear systems are solved using lrGMRES (Algorithm 5). We extend the hybrid approach to an inexact variant based on an inexact Newton algorithm, in which the accuracy of the approximate linear system solution is tied to the accuracy of the nonlinear iterate (see e.g., [57] and references therein). That is, when the nonlinear iterate is far from the solution, the linear systems may not have to be solved accurately. Thus, a sequence of iterates $\bar{u}^{n+1} := \bar{u}^n + \delta\bar{u}^n$ is computed where $\delta\bar{u}^n$ satisfies

$$\|J_N^n \delta\bar{u}^n + \bar{r}^n\|_2 \leq \epsilon_{\text{gmres}}^n \|\bar{r}^n\|_2, \quad (J_P \text{ for Picard iteration}),$$

where the lrGMRES stopping tolerance ($\epsilon_{\text{gmres}}^n$ of Algorithm 5) is given by

$$\epsilon_{\text{gmres}}^n := \rho_{\text{gmres}} \|\bar{r}^n\|_2, \quad (4.34)$$

where $0 < \rho_{\text{gmres}} \leq 1$. With this strategy, the Jacobian system is solved with increased accuracy as the error becomes smaller, leading to savings in the average cost per step and, as we will show, with no degradation in the asymptotic convergence rate of the nonlinear iteration.

In addition, in Algorithms 4 and 5, the truncation operator $\mathcal{T}_{\epsilon_{\text{trunc}}}$ is used for the low-rank approximation of the nonlinear iterate (i.e., truncating \bar{u}^x , \bar{u}^y , and \bar{p} at lines 5 and 9 in Algorithm 4) and updates (i.e., truncating $\delta\bar{u}^x$, $\delta\bar{u}^y$, and $\delta\bar{p}$ in Algorithm 5). As the lrGMRES stopping tolerance is adaptively determined by the criterion (4.34), we also choose the value of the truncation tolerances $\epsilon_{\text{trunc,sol}}$ and $\epsilon_{\text{trunc,corr}}^n$, adaptively. For truncating the nonlinear iterate, the truncation tolerance for the iterate $\{\epsilon_{\text{trunc,sol}}^n\}$ is chosen based on the nonlinear iteration stopping tolerance,

$$\epsilon_{\text{trunc,sol}} := \rho_{\text{nl}} \epsilon_{\text{nl}},$$

where $0 < \rho_{\text{nl}} \leq 1$. For truncating the updates (or corrections), the truncation tolerance for the correction $\{\epsilon_{\text{trunc,corr}}^n\}$ is adaptively chosen based on the stopping

tolerance of the linear solver,

$$\epsilon_{\text{trunc,corr}}^n := \rho_{\text{trunc,P}} \epsilon_{\text{gmres}}^n, \quad (\text{for the } n\text{th Picard step}),$$

$$\epsilon_{\text{trunc,corr}}^n := \rho_{\text{trunc,N}} \epsilon_{\text{gmres}}^n, \quad (\text{for the } n\text{th Newton step}),$$

where $0 < \rho_{\text{trunc,P}}, \rho_{\text{trunc,N}} \leq 1$. Thus, for computing n th update $\delta \bar{u}^n$, we set $\epsilon_{\text{trunc}} = \epsilon_{\text{trunc,corr}}^n$ in Algorithm 5.

Algorithm 6 Inexact nonlinear iteration with adaptive tolerances

- 1: set $\epsilon_{\text{trunc,sol}} := \rho_{\text{nl}} \epsilon_{\text{nl}}$
 - 2: compute an approximate solution of $A_{\text{st}} \bar{u}_{\text{st}} = b_{\text{st}}$ using Algorithm 5
 - 3: set an initial guess for the Navier–Stokes problem $\bar{u}^0 := \bar{u}_{\text{st}}$
 - 4: **for** $k = 0, \dots, m_p - 1$ **do**
 - 5: set $\epsilon_{\text{gmres}}^k = \rho_{\text{gmres}} \|\bar{r}^k\|_2$, and $\epsilon_{\text{trunc,corr}}^k = \rho_{\text{trunc,P}} \|\bar{r}^k\|_2$
 - 6: solve $J_{\text{P}}^k \delta \bar{u}^k = -\bar{r}^k$ using Algorithm 5
 - 7: update $\bar{u}^{k+1} := \mathcal{T}_{\epsilon_{\text{trunc,sol}}}(\bar{u}^k + \delta \bar{u}^k)$
 - 8: **end for**
 - 9: **while** $\|\bar{r}^k\|_2 > \epsilon_{\text{nl}} \|\bar{r}^0\|_2$ **do**
 - 10: set $\epsilon_{\text{gmres}}^k = \rho_{\text{gmres}} \|\bar{r}^k\|_2$, and $\epsilon_{\text{trunc,corr}}^k = \rho_{\text{trunc,N}} \|\bar{r}^k\|_2$
 - 11: solve $J_{\text{N}}^k \delta \bar{u}^k = -\bar{r}^k$ using Algorithm 5
 - 12: update $\bar{u}^{k+1} := \mathcal{T}_{\epsilon_{\text{trunc,sol}}}(\bar{u}^k + \delta \bar{u}^k)$
 - 13: **end while**
-

4.5 Numerical results

In this section, we present the results of numerical experiments on a model problem, flow around a square obstacle in a channel, for which the details are depicted in Figure 4.1. The domain has length 12 and height 2, and it contains a square obstacle centered at (2,0) with sides of length .25.

For the numerical experiments, we define the random viscosity (4.2) using the

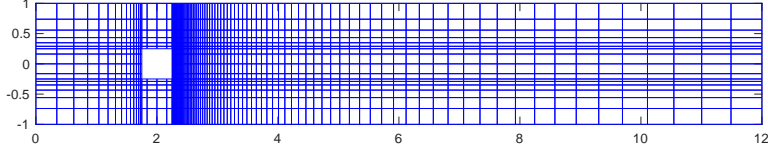


Figure 4.1: Spatial domain and finite element discretization.

Karhunen-Loève (KL) expansion [67],

$$\nu(x, \xi) = \nu_0 + \sigma_\nu \sum_{i=1}^{n_\nu} \sqrt{\lambda_i} \nu_i(x) \xi_i, \quad (4.35)$$

where ν_0 and σ_ν^2 are the mean and the variance of the viscosity of $\nu(x, \xi)$, and $\{(\lambda_i, \nu_i(x))\}_{i=1}^{n_\nu}$ are eigenpairs of the eigenvalue problem associated with the covariance kernel $C(x, y)$ of the random field. We consider two types of covariance kernel: absolute difference exponential (AE) and squared difference exponential (SE), which are defined via

$$C^{\text{AE}}(x, y) = \exp\left(-\sum_{i=1}^2 \frac{|x_i - y_i|}{l_i}\right), \quad C^{\text{SE}}(x, y) = \exp\left(-\sum_{i=1}^2 \frac{(x_i - y_i)^2}{l_i^2}\right), \quad (4.36)$$

where $x = (x_1, x_2)$ and $y = (y_1, y_2)$ are points in the spatial domain, and l_1, l_2 are correlation lengths. We assume that the random variables $\{\xi_i\}_{i=1}^{n_\nu}$ are independent and identically distributed and that ξ_i (for $i = 1, \dots, n_\nu$) follows a uniform distribution over $[-1, 1]$. For the mean of the viscosity, we consider several choices, $\nu_0 = \{\frac{1}{50}, \frac{1}{100}, \frac{1}{150}\}$, which corresponds to $\text{Re}_0 = \{100, 200, 300\}$. In all experiments, we use a finite-term KL-expansion with $n_\nu = 5$. For constructing the

finite-dimensional approximation space $S = \text{span}(\{\psi_i(\xi)\}_{i=1}^{n_\xi})$ in the parameter domain, we use orthogonal polynomials $\{\psi_i(\xi)\}_{i=1}^{n_\xi}$ of total degree 3, which results in $n_\xi = 56$. The orthogonal polynomials associated with uniform random variables are Legendre polynomials, $\psi_i(\xi) = \prod_{j=1}^{n_\nu} \ell_{d_j(i)}(\xi_j)$ where $d(i) = (d_1(i), \dots, d_{n_\nu}(i))$ is a multi-index consisting of non-negative integers and $\ell_{d_j(i)}$ is the $d_j(i)$ th order Legendre polynomial of ξ_j . For the spatial discretization, Taylor–Hood elements are used on a stretched grid, which results in $\{6320, 6320, 1640\}$ degrees of freedom in $\{\vec{u}^x, \vec{u}^y, p\}$, respectively (i.e., $n_u = 6320$ and $n_p = 1640$.) The implementation is based on the Incompressible Flow and Iterative Solver Software (IFISS) package [38, 98].

4.5.1 Low-rank inexact nonlinear iteration

In this section, we compare the results obtained from the low-rank inexact nonlinear iteration with those obtained from other methods, the exact and the inexact nonlinear iteration with full rank solutions, and the Monte Carlo method. Default parameter settings are listed in Table 4.1, where the truncation tolerances only apply to the low-rank method. Unless otherwise specified, the linear system is solved using a restarted version of low-rank GMRES, lrGMRES(20).

We first examine the convergence behavior of the inexact nonlinear iteration for a model problem characterized by $\text{Re}_0 = 100$, $\text{CoV} = 1\%$, and SE covariance kernel in (4.36) with $l_1 = l_2 = 32$. We compute a full-rank solution using the exact nonlinear iteration ($\epsilon_{\text{gmres}}^n = 10^{-12}$ and no truncation) until the nonlinear iterate reaches

Table 4.1: Tolerances and adaptive parameters.

Nonlinear iteration stopping tolerance	$\epsilon_{\text{nl}} = 10^{-5}$
GMRES tolerance (Stokes)	$\epsilon_{\text{gmres}} = 10^{-4}$
GMRES tolerances (Picard and Newton)	$\epsilon_{\text{gmres}}^n = \rho_{\text{gmres}} \ \bar{r}^n\ _2$ ($\rho_{\text{gmres}} = 10^{-5}$)
Truncation tolerance for solutions	$\epsilon_{\text{trunc,sol}} = \rho_{\text{nl}} \epsilon_{\text{nl}}$ ($\rho_{\text{nl}} = 10^{-1}$)
Truncation tolerance for corrections	$\epsilon_{\text{trunc,corr}}^n = \rho_{\text{trunc}} \epsilon_{\text{gmres}}^n$ ($\rho_{\text{trunc}} = 10^{-1}$)

the nonlinear stopping tolerance, $\epsilon_{\text{nl}} = 10^{-8}$. Then we compute another full-rank solution using the inexact nonlinear iteration (i.e., adaptive choice of $\epsilon_{\text{gmres}}^n$ as shown in Table 4.1 and no truncation). Lastly, we compute a low-rank approximate solution using the low-rank inexact nonlinear iteration (i.e., adaptive choices of $\epsilon_{\text{gmres}}^n$ and $\epsilon_{\text{trunc,corr}}^n$ as shown in Table 4.1 and for varying $\epsilon_{\text{trunc,sol}} = \{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$). Figure 4.2 shows the convergence behavior of the three methods. In Figure 4.2(a), the hybrid approach is used, in which the first step corresponds to the Stokes problem (line 2 of Algorithm 6), the 2nd–5th steps correspond to the Picard iteration (line 4–8 of Algorithm 6, and $m_p = 4$), and the 6th–7th steps correspond to the Newton iteration (line 9–13 of Algorithm 6). Figure 4.2(a) confirms that the inexact nonlinear iteration is as effective as the exact nonlinear iteration. The low-rank inexact nonlinear iteration behaves similarly up to the 6th nonlinear step but when the truncation tolerances are large $\epsilon_{\text{trunc,sol}} = \{10^{-5}, 10^{-6}\}$, it fails to produce a nonlinear solution satisfying $\epsilon_{\text{nl}} = 10^{-8}$. Similar results can be seen in Figure 4.2(b), where only the Picard iteration is used. As expected, in that case, the relative residual decreases linearly for all solution methods, but the low-rank inexact nonlinear iteration with the mild truncation tolerances also fails to reach the nonlinear

iteration stopping tolerance.

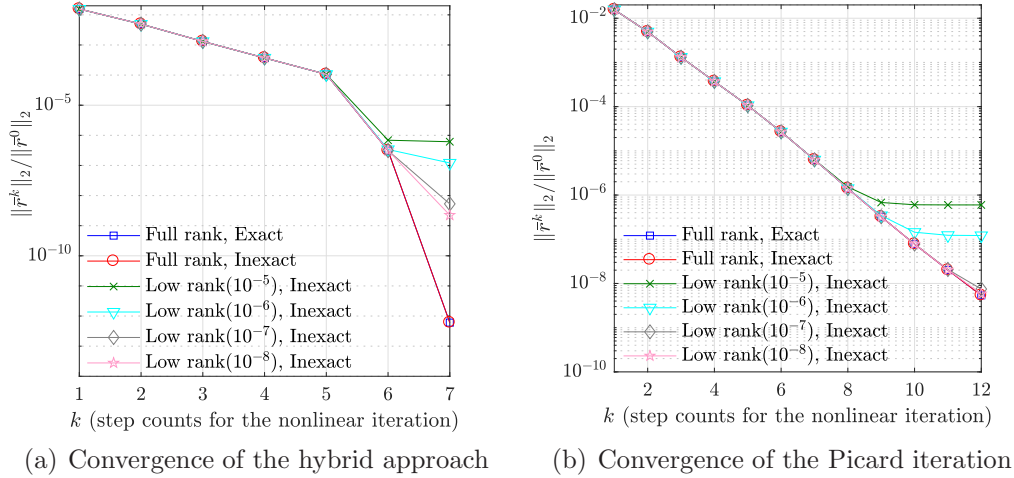


Figure 4.2: Convergence of both exact and inexact nonlinear iterations (full-rank) and the low-rank inexact nonlinear iteration.

Figure 4.3 shows means and variances of the components of the full-rank solution, given by

$$\mu_{u^x} = \mathbb{E}[\vec{u}^x], \quad \mu_{u^y} = \mathbb{E}[\vec{u}^y], \quad \mu_p = \mathbb{E}[p], \quad (4.37)$$

$$\sigma_{u^x}^2 = \mathbb{E}[(\vec{u}^x - \mu_{u^x})^2], \quad \sigma_{u^y}^2 = \mathbb{E}[(\vec{u}^y - \mu_{u^y})^2], \quad \sigma_p^2 = \mathbb{E}[(p - \mu_p)^2]. \quad (4.38)$$

These quantities are easily computed by exploiting the orthogonality of basis functions in the gPC expansion. Figure 4.4 shows the differences in the means and variances of the solutions computed using the full-rank and the low-rank inexact nonlinear iteration. Let us denote the full-rank and low-rank horizontal velocity solutions by $u^{x,\text{full}}$ and $u^{x,\text{lr}}$, with analogous notation for the vertical velocity and

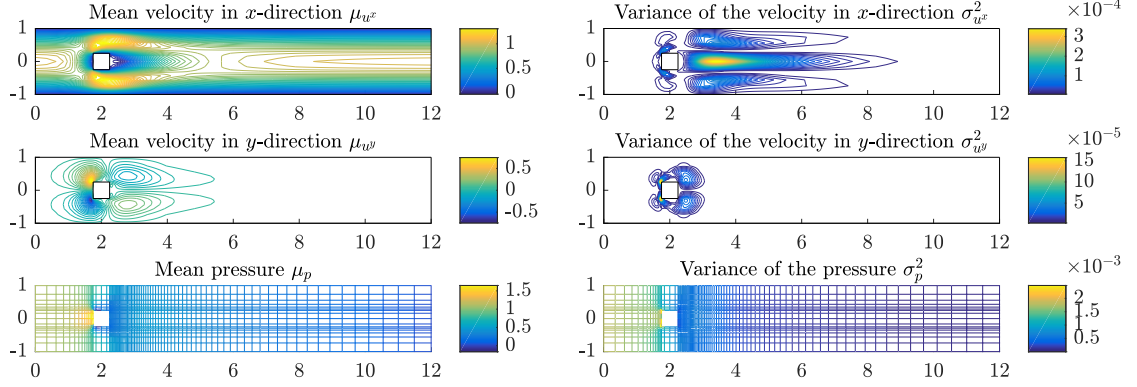


Figure 4.3: Mean and variances of full-rank velocity solutions $\vec{u}^x(x, \xi)$, $\vec{u}^y(x, \xi)$, and pressure solution $p(x, \xi)$ for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$.

the pressure. Thus, the differences in the means and the variances are

$$\begin{aligned} \eta_\mu^x &= \mu_{u^x, \text{full}} - \mu_{u^x, \text{lr}}, & \eta_\mu^y &= \mu_{u^y, \text{full}} - \mu_{u^y, \text{lr}}, & \eta_\mu^p &= \mu_{p^{\text{full}}} - \mu_{p^{\text{lr}}}, \\ \eta_\sigma^x &= \sigma_{u^x, \text{full}}^2 - \sigma_{u^x, \text{lr}}^2, & \eta_\sigma^y &= \sigma_{u^y, \text{full}}^2 - \sigma_{u^y, \text{lr}}^2, & \eta_\sigma^p &= \sigma_{p^{\text{full}}}^2 - \sigma_{p^{\text{lr}}}^2. \end{aligned}$$

Figure 4.4 shows these differences, normalized by graph norms $\|\nabla \vec{\mu}_{u^{\text{full}}}\| + \|\mu_{p^{\text{full}}}\|$ for the means and $\|\nabla \vec{\sigma}_{u^{\text{full}}}\| + \|\sigma_{p^{\text{full}}}\|$ for the variances, where $\|\nabla \vec{u}\| = (\int_D \nabla \vec{u} : \nabla \vec{u} dx)^{\frac{1}{2}}$ and $\|p\| = (\int_D p^2 dx)^{\frac{1}{2}}$. Figure 4.4 shows that the normalized differences in the mean and the variance are of order $10^{-9} \sim 10^{-10}$ and $10^{-10} \sim 10^{-12}$, respectively, i.e., the errors in low-rank solutions are considerably smaller than the magnitude of the truncation tolerances $\epsilon_{\text{trunc, sol}}, \epsilon_{\text{trunc, corr}}$ (see Table 4.1).

4.5.2 Characteristics of the Galerkin solution

In this section, we examine various properties of the Galerkin solutions, with emphasis on comparison of the low-rank and full-rank versions of these solutions and

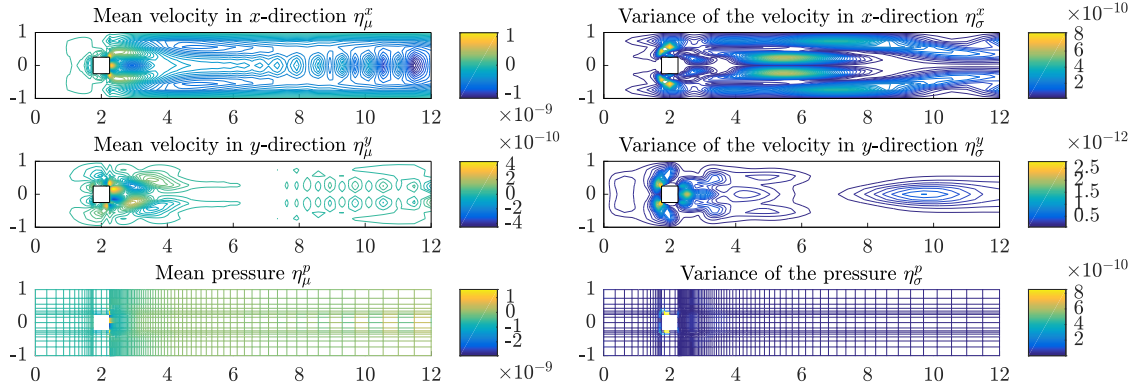


Figure 4.4: Difference in the means and variances of the full-rank and the low-rank solutions for $\text{Re}_0 = 100$, $\text{CoV} = 1$, and $l_1 = l_2 = 32$.

development of an enhanced understanding of the relation between the Galerkin solution and the polynomial chaos basis. We use the same experimental setting studied above (SE covariance kernel, $l_1 = l_2 = 32$, $\text{Re}_0 = 100$ and $\text{CoV} = 1\%$).

We begin by comparing the Galerkin solution with one obtained using Monte Carlo methods. In particular, we estimate a probability density function (pdf) of the velocity solutions $(\vec{u}^x(x, \xi), \vec{u}^y(x, \xi))$ and the pressure solution $(p(x, \xi))$ at a specific point on the spatial domain D . In the Monte Carlo method, we solve $n_{\text{MC}} = 25000$ deterministic systems, $\mathcal{R}(\vec{u}, p, \vec{v}, q; \xi^{(k)}) = 0$ associated with n_{MC} realizations $\{\xi^{(k)}\}_{k=1}^{n_{\text{MC}}}$ in the parameter space. Using the MATLAB function `ksdensity`, the pdfs of $(\vec{u}^x(x, \xi), \vec{u}^y(x, \xi), p(x, \xi))$ are estimated at the spatial point with coordinates $(3.6436, 0)$, where the variance of $\vec{u}^x(x, \xi)$ is large (see Figure 4.3). The results are shown in Figure 4.5. They indicate that the pdf of the Galerkin solution is virtually identical to that of the Monte Carlo solution, and there is essentially no difference between the low-rank and full-rank results.

Next, we explore some characteristics of the Galerkin solution, focusing on the

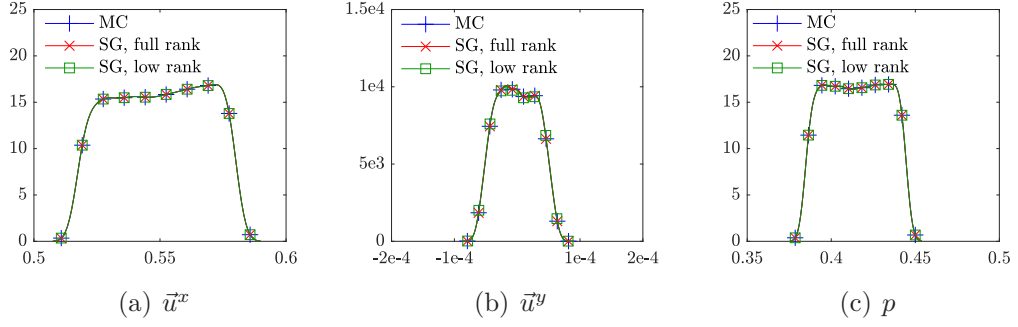


Figure 4.5: Estimated pdfs of the velocities \vec{u}^x , \vec{u}^y , and the pressure p at the point $(3.6436, 0)$.

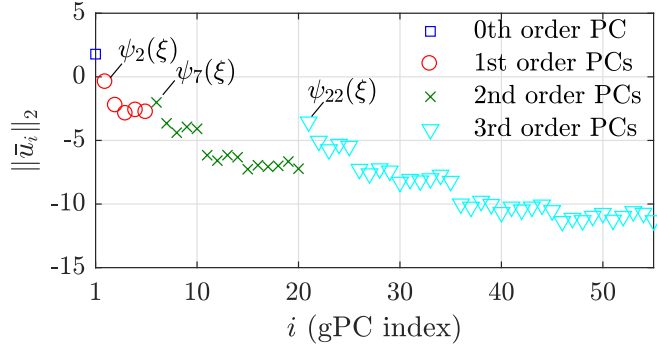


Figure 4.6: Norms of the gPC coefficients $\|\bar{u}_i\|_2$ for $Re_0 = 100$, $CoV = 1$, and $l_1 = l_2 = 32$.

horizontal velocity solution; the observations made here also hold for the other components of the solution. Given the coefficients of the velocity solution in matrixized form, U^x , the discrete velocity solution is then given by

$$\vec{u}^x(x, \xi) = \Phi^T(x)U^x\Psi(\xi),$$

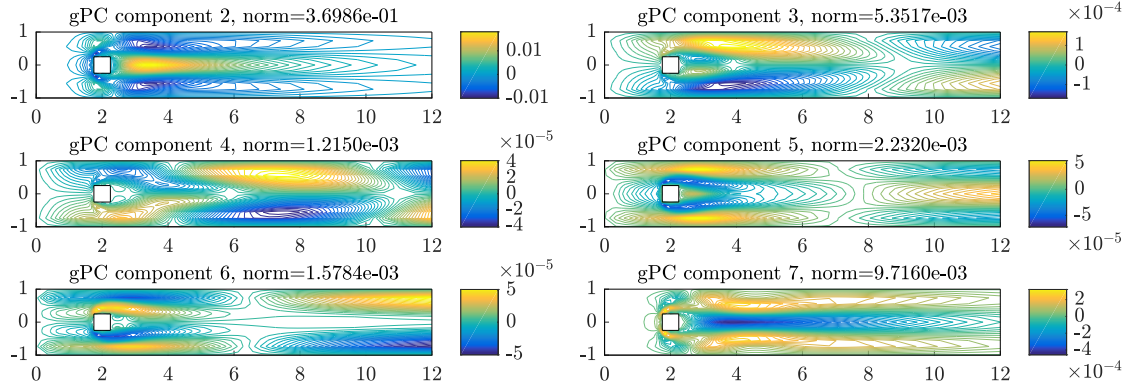
where $\Phi(x) = [\phi_1(x), \dots, \phi_{n_u}(x)]^T$ and $\Psi(\xi) = [\psi_1(\xi), \dots, \psi_{n_\xi}(\xi)]^T$. Consider in particular the component of this expression corresponding to the j th column of U^x ,

$$\left(\sum_{i=1}^{n_u} u_{ij}^x \phi_i(x) \right) \psi_j(\xi)$$

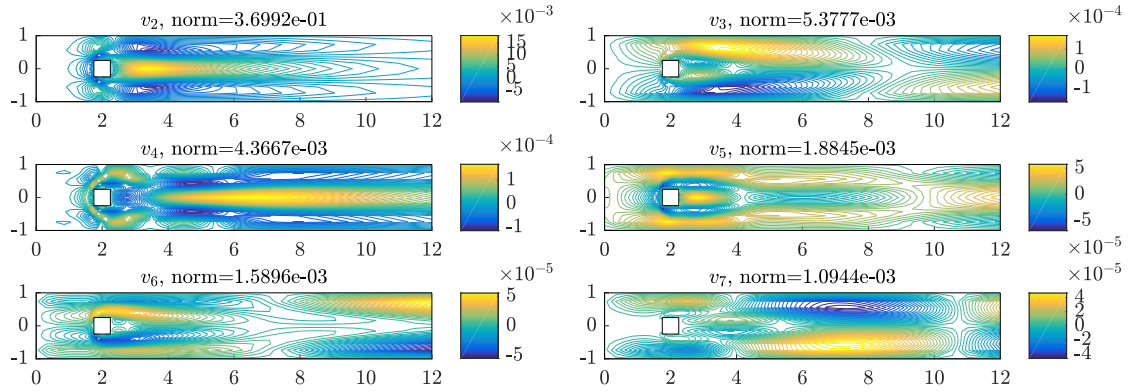
so that this (j th) column $\bar{u}_j^x = [U^x]_j$ corresponds to the coefficient of the j th polynomial basis function ψ_j . Figure 4.6 plots the values of the coefficients $\|\bar{u}_i^x\|_2$. (This data is computed with $\text{Re}_0 = 100$, $\text{CoV} = 1\%$, and SE covariance kernel with $l_1 = l_2 = 32$). Note that the gPC indices $\{j\}$ are in one-to-one correspondence with multi-indices $d(j) = (d_1(j), \dots, d_{n_u}(j))$, where the element of the multi-index indicates the degree of univariate Legendre polynomial. The multi-indices $\{d(i)\}_{i=1}^{n_\xi}$ are ordered in the lexicographical order, for example, the first eight multi-indices are as $d(1) = (0, 0, 0, 0, 0)$, $d(2) = (1, 0, 0, 0, 0)$, $d(3) = (0, 1, 0, 0, 0)$, \dots , $d(6) = (0, 0, 0, 0, 1)$, $d(7) = (2, 0, 0, 0, 0)$, and $d(8) = (1, 1, 0, 0, 0)$. In Figure 4.6, the blue square is associated with the zeroth-order gPC component ($d(1)$), the red circles are associated with the first-order gPC components ($\{d(i)\}_{i=2}^6$), and so on. Let us focus on three gPC components associated only with ξ_1 , $\{\psi_2(\xi) = \ell_1(\xi_1)$, $\psi_7(\xi) = \ell_2(\xi_1)$, $\psi_{22}(\xi) = \ell_3(\xi_1)\}$, where, for $i = 2, 7, 22$, the multi-indices are $d(2) = (1, 0, 0, 0, 0)$, $d(7) = (2, 0, 0, 0, 0)$, and $d(22) = (3, 0, 0, 0, 0)$. The figure shows that the coefficients of gPC components $\{\psi_2(\xi), \psi_7(\xi), \psi_{22}(\xi)\}$ decay more slowly than those of gPC components associated with other random variables $\{\xi_i\}_{i=2}^{n_\nu}$.

We continue the examination of this data in Figure 4.7(a), which shows two-dimensional mesh plots of the 2nd through 7th columns of U^x . These images show that these coefficients are either symmetric with respect to the horizontal axis, or reflectionally symmetric (equal in magnitude but of opposite sign), and (as also revealed in Figure 4.6), they tend to have smaller values as the index j is increased.

We now look more closely at features of the factors of the low-rank approximate solution and compare these with those of the (unfactored) full-rank solu-



(a) Plots of coefficients of gPC components 2–7 of $\vec{u}^x(x, \xi)$



(b) Plots of coefficients v_i of $\theta_i^x(\xi)$ for $i = 2, \dots, 7$

Figure 4.7: Plots of coefficients of gPC components 2–7 of $\vec{u}^x(x, \xi)$ and coefficients v_i of $\theta_i(\xi)$ for $i = 2, \dots, 7$ for $\text{Re}_0 = 100$, $CoV = 1$, and $l_1 = l_2 = 32$.

tion. In the low-rank format, the solution is represented using factors $\vec{u}^x(x, \xi) = (\Phi^T(x)V^x)(\Psi^T(\xi)W^x)^T$. Let us introduce a concise notation of

$$\vec{u}^x(x, \xi) = Z_{\alpha_{\vec{u}^x}}^x(x)^T \Theta_{\alpha_{\vec{u}^x}}^x(\xi) = \sum_{i=1}^{\alpha_{\vec{u}^x}} \zeta_i^x(x) \theta_i^x(\xi)$$

where $Z_{\alpha_{\vec{u}^x}}^x(x) = [\zeta_1^x(x), \dots, \zeta_{\alpha_{\vec{u}^x}}^x(x)]$ and $\Theta_{\alpha_{\vec{u}^x}}^x(\xi) = [\theta_1^x(\xi), \dots, \theta_{\alpha_{\vec{u}^x}}^x(\xi)]$ with $\zeta_i^x(x) = [\Phi^T(x)V^x]_i$ and $\theta_i^x(\xi) = [(\Psi^T(\xi)W^x)]_i$ for $i = 1, \dots, \alpha_{\vec{u}^x}$. Figure 4.7(b) shows the coefficients of the i th random variable $\theta_i(\xi)$. As opposed to the gPC coefficients of the

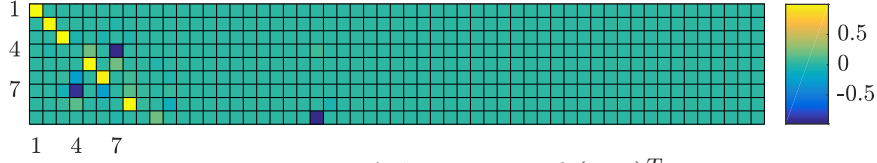


Figure 4.8: A heat map of $(W^x)^T$.

full-rank solution, the norms of the coefficients of $\{\theta_i(\xi)\}$ decrease monotonically as the index i increases. This is a consequence of the fact that the ordering for $\{\theta_i(\xi)\}$ comes from the singular values of U^x . Figure 4.7(b) shows the 2nd-7th columns of V^x . Figures 4.7(a) and 4.7(b) show that the coefficients $\{v_i\}$ of $\{\theta_i(\xi)\}$ are comparable to the coefficients $\{u_i^x\}$ of the gPC components. Each pair of components in the following parentheses is similar to each other: (u_2, v_2) , (u_3, v_3) , $(u_7, -v_4)$, $(u_4, -v_7)$, (u_5, v_5) , and $(u_6, -v_6)$.

While the columns of V^x show the resemblance to the subset of the columns of U^x , W^x tends to act as a permutation matrix. Figure 4.8 shows a “heat map” of $(W^x)^T$, where values of the elements in W^x are represented as colors and the map shows that a very few elements of W_i^x are dominant and a sum of those elements is close to 1. Recall that $\theta_i^x(\xi) = (W_i^x)^T \Psi(\xi)$. Many dominant elements are located in the diagonal of W^x , which results in $\theta_i^x(\xi) \approx \pm \psi_i(\xi)$ (e.g., $i = 1, 2, 3, 5, \dots$). In the case of W_4^x , the most dominant element is the 7th element and has a value close to -1, which results in $\theta_4^x(\xi) \approx -\psi_7(\xi)$. As observed in Figure 4.6, $\psi_7(\xi)$ has a larger contribution than other gPC components and, in the new solution representation, $\theta_4^x(\xi)$, which consists mainly of $\psi_7(\xi)$, appears earlier in the representation.

4.5.3 Computational costs

In this section, we assess the costs of the low-rank inexact nonlinear iteration under various experimental settings: two types of covariance kernels (4.36), varying CoV (4.3), and varying Re_0 . In addition, for various values of these quantities, we investigate the decay of the eigenvalues $\{\lambda_i\}$ used to define the random viscosity (4.35) and their influence on the rank of solutions. All numerical experiments are performed on an Intel 3.1 GHz i7 CPU, 16 GB RAM using MATLAB R2016b and costs are measured in terms of CPU wall time (in seconds). For larger CoV and Re_0 , we found the solver to be more effective using the slightly smaller truncation tolerance $\rho_{\text{trunc}} = 10^{-1.5}$ and used this choice for all experiments described below. (Other adaptive tolerances are those shown as in Table 4.1.) This change had little impact on results for small CoV and Re_0 .

Figure 4.9 shows the 50 largest eigenvalues $\{\lambda_i\}$ of the eigenvalue problems associated with the SE covariance kernel and the AE covariance kernel (4.36) with $l_1 = l_2 = 8$, $CoV = 1\%$, and $Re_0 = 100$. The eigenvalues of the SE covariance kernel decay much more rapidly than those of the AE covariance kernel. Because we choose a fixed number of terms $n_\nu = 5$, the random viscosity with the SE covariance kernel retains a smaller variance.

Figure 4.10(a) shows the computational costs (in seconds) needed for computing the full-rank solutions and the low-rank approximate solutions using the inexact nonlinear iteration for the two covariance kernels and a set of correlation lengths, $l_1 = l_2 = \{1, 2, 4, 8, 16, 32\}$. Figure 4.10(b) shows the ranks of the low-rank approx-

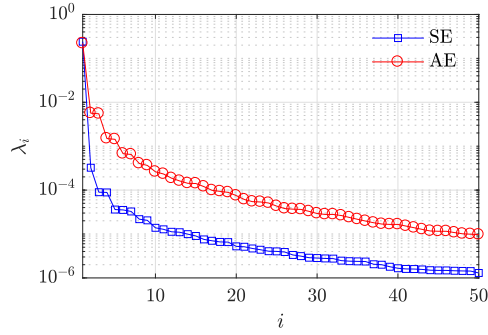
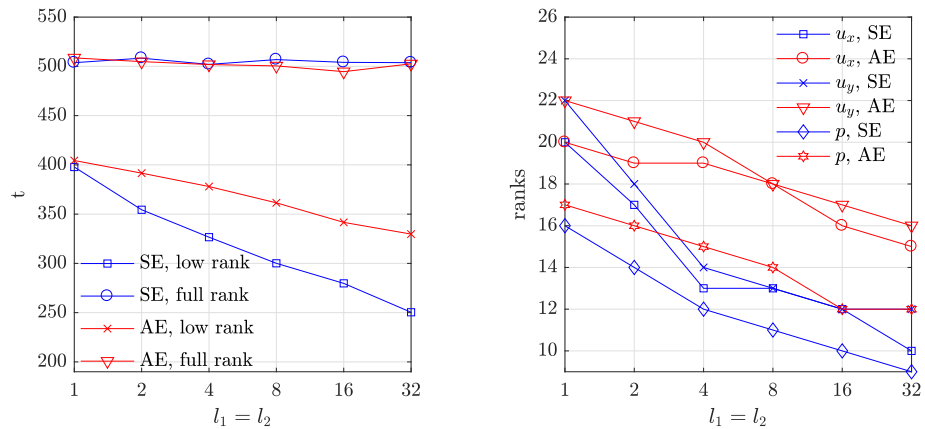


Figure 4.9: Eigenvalue decay of the AE and the SE covariance kernels.

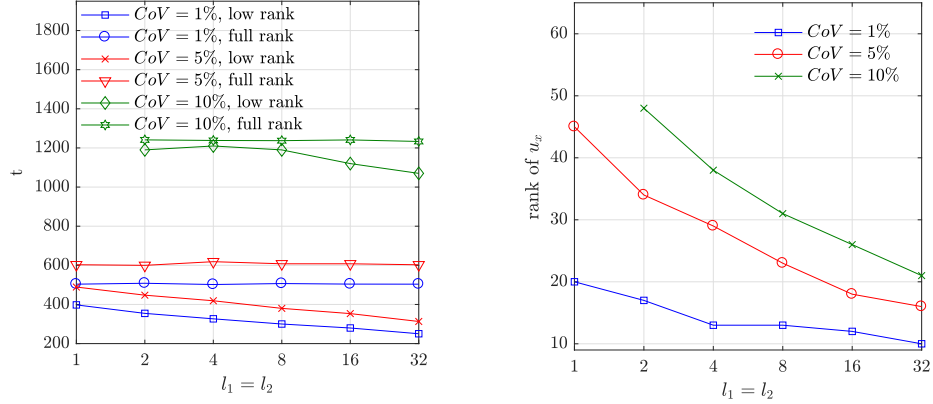
imate solutions that satisfy the nonlinear stopping tolerance $\epsilon_{\text{nl}} = 10^{-5}$. Again, $\text{Re}_0 = 100$ and $\text{CoV} = 1\%$. For this benchmark problem, 4 Picard iterations and 1 Newton iteration are enough to generate a nonlinear iterate satisfying the stopping tolerance ϵ_{nl} . It can be seen from Figure 4.10(a) that in all cases the use of low rank methods reduces computational cost. Moreover, as the correlation length becomes larger, the ranks of the corrections and the nonlinear iterates become smaller. As a result, the low-rank method achieves greater computational savings for the problems with larger correlation length.



(a) Computational cost of full-rank computation and low-rank approximation (b) Ranks of the low-rank approximate solutions

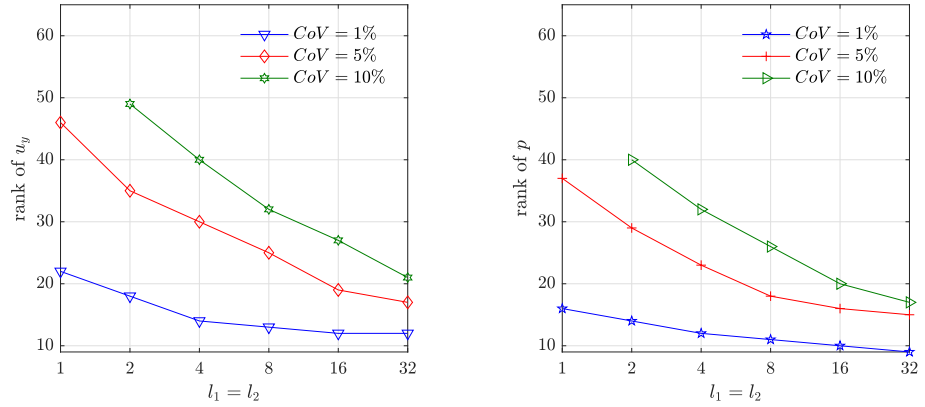
Figure 4.10: Computational costs and ranks for varying correlation lengths with SE and AE covariance kernel.

Next, we examine the performances of the low-rank approximation method for varying CoV , which is defined in (4.3). In this experiment, we fix the value of $Re_0 = 100$ and the variance of the random σ_ν is controlled. We consider the SE covariance kernel.



(a) Computational cost of full-rank computation and low-rank approximation

(b) Ranks of the low-rank approximate solutions u_x



(c) Ranks of the low-rank approximate solutions u_y

(d) Ranks of the low-rank approximate solutions p

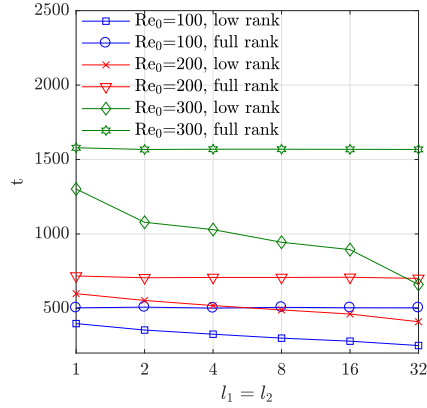
Figure 4.11: Computational costs and ranks for varying correlation lengths and varying CoV with $Re_0 = 100$.

Figure 4.11 shows the performances of the full-rank and the low-rank methods for varying $CoV = \{1\%, 5\%, 10\%\}$. We use Algorithm 6 with 4 Picard steps,

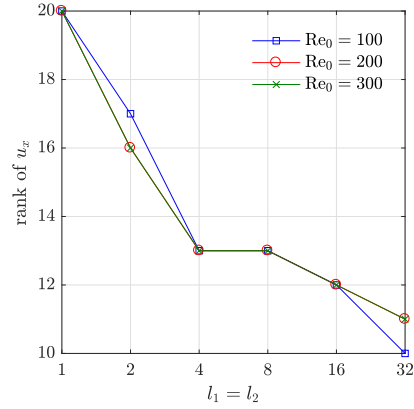
followed by several Newton steps until convergence. For $CoV = \{1\%, 5\%\}$, one Newton step is required for the convergence and, for $CoV = 10\%$, two Newton steps are required. Figure 4.11(a) shows the computational costs. For $CoV = \{1\%, 5\%\}$, the computational benefits of using the low-rank approximation methods are pronounced whereas, for $CoV = 10\%$, the performances of the two approaches are essentially the same for shorter correlation lengths. Indeed, for higher CoV , the ranks of solutions \bar{u} (see Figures 4.11(b)–4.11(d)) as well as updates $\delta\bar{u}^k$ at Newton steps become close to the full rank ($n_\xi = 56$).

Lastly, we study the benchmark problems with varying mean viscosity with SE covariance kernel and $CoV = 1\%$. As the mean viscosity decreases, Re_0 grows, and the nonlinear problem tends to become harder to solve, and for the larger Reynolds numbers $Re_0 = 200$ or 300 , we use more Picard steps (5 or 6, respectively) before switching to Newton’s method.

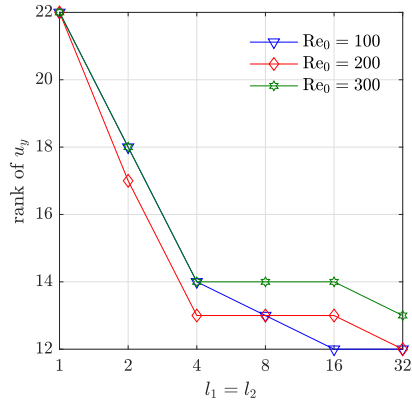
Figure 4.12 shows the performances of the low-rank methods for varying Reynolds number, $Re_0 = \{100, 200, 300\}$. For $Re_0 = 200$, after 5 Picard steps, one Newton step leads to convergence (and 6 Picard steps and one Newton step for $Re_0 = 300$). As the figures 4.12(b)–4.12(d) show, the ranks of the solutions increase slightly as the Reynolds number becomes larger and, thus, for all Re_0 tested here, the low-rank method demonstrates notable computational savings (with $CoV = 1\%$). Note that overall computational costs in Figure 4.12(a) increase as the Reynolds number becomes larger because (1) the number of nonlinear steps required to converge increases as the Reynolds number increases and (2) to solve each linearized systems, typically more lrGMRES cycles are required for the problems with higher



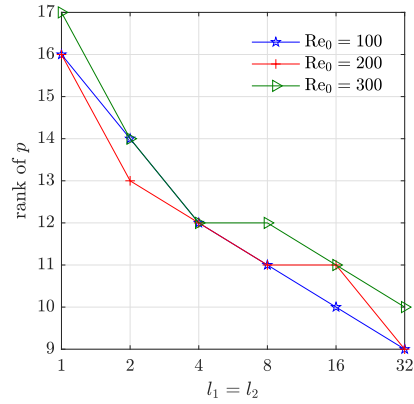
(a) Computational cost of full-rank computation and low-rank approximation



(b) Ranks of the low-rank approximate solutions u_x



(c) Ranks of the low-rank approximate solutions u_y



(d) Ranks of the low-rank approximate solutions p

Figure 4.12: Computational costs and ranks for varying correlation lengths and varying Re_0 .

Reynolds number.

4.6 Conclusion

In this study, we have developed the inexact low-rank nonlinear iteration for the solutions of the Navier–Stoke equations with uncertain viscosity in the stochastic Galerkin context. At each step of the nonlinear iteration, the solution of the linear

system is inexpensively approximated in low rank using the tensor variant of the GMRES method. We examined the effect of the truncation to an accuracy of the low-rank approximate solutions by comparing those solutions to the ones computed using exact, inexact nonlinear iterations in full rank and the Monte Carlo method. Then we explored the efficiency of the proposed method with a set of benchmark problems for various settings of uncertain viscosity. The numerical experiments demonstrated that the low-rank nonlinear iteration achieved significant computational savings for the problems with smaller CoV and larger correlation lengths. The experiments also showed that the mean Reynolds number does not significantly affect the rank of the solution and the low-rank nonlinear iteration achieves computational savings for varying Reynolds number for small CoV and large correlation lengths.

Chapter 5: Stochastic Least-Square Petrov Galerkin method

5.1 Introduction

In this chapter, we consider the issues of optimality associated with the stochastic Galerkin method. The stochastic Galerkin method combined with generalized polynomial chaos (gPC) expansions [112] seeks a polynomial approximation of the numerical solution in the stochastic domain by enforcing a Galerkin orthogonality condition, i.e., the residual of the parameterized linear system is forced to be orthogonal to the span of the stochastic polynomial basis with respect to an inner product associated with an underlying probability measure. The Galerkin projection scheme is popular for its simplicity (i.e., the trial and test bases are the same) and its optimality in terms of minimizing an energy norm of solution errors when the underlying PDE operator is symmetric positive definite. In many applications, however, the stochastic Galerkin method does not exhibit any optimality property [73]. That is, it does not produce solutions that minimize any measure of the solution error. In such cases, the stochastic Galerkin method can lead to poor approximations and non-convergent behavior.

To address this issue, we propose a novel optimal projection technique, which we refer to as the stochastic least-squares Petrov–Galerkin (LSPG) method. Inspired

by the successes of LSPG methods in nonlinear model reduction [18–20], finite element methods [13, 14, 55], and iterative linear solvers (e.g., GMRES, GCR) [91], we propose, as an alternative to enforcing the Galerkin orthogonality condition, to directly minimize the residual of a parameterized linear system over the stochastic domain in a (weighted) ℓ^2 -norm. The stochastic LSPG method produces an optimal solution for a given stochastic subspace and guarantees that the ℓ^2 -norm of the residual monotonically decreases as the stochastic basis is enriched. In addition to producing monotonically convergent approximations as measured in the chosen weighted ℓ^2 -norm, the method can also be adapted to target output quantities of interest (QoI); this can be accomplished by employing a weighted ℓ^2 -norm used for least-squares minimization that coincides with the ℓ^2 -(semi)norm of the error in the chosen QoI.

In addition to proposing the stochastic LSPG method, this study shows that specific choices of weighting functions lead to equivalences between the stochastic LSPG method and both the stochastic Galerkin method and the pseudo-spectral method [109, 110]. We demonstrate the effectiveness of the LSPG method with extensive numerical experiments on various SPDEs. The results show that the proposed LSPG technique significantly outperforms the stochastic Galerkin when the solution error is measured in different weighted ℓ^2 -norms. We also show that the proposed method can effectively minimize the error in target QoIs.

An outline of the chapter is as follows. Section 5.2 formulates parameterized linear algebraic systems and reviews conventional spectral approaches for computing numerical solutions. Section 5.3 develops a residual minimization formulation

based on least-squares methods and its adaptation to the stochastic LSPG method. We also provide proofs of optimality and monotonic convergence behavior of the proposed method. Section 5.4 provides error analysis for stochastic LSPG methods. Section 5.5 demonstrates the efficiency and the effectiveness of the proposed methods by testing them on various benchmark problems. Finally, Section 5.6 outlines some conclusions.

5.2 Spectral methods for parameterized linear systems

We begin by introducing a mathematical formulation of parameterized linear systems and briefly reviewing the stochastic Galerkin and the pseudo-spectral methods, which are spectral methods for approximating the numerical solutions of such systems.

5.2.1 Problem formulation

Consider a parameterized linear system

$$A(\xi)u(\xi) = b(\xi), \quad (5.1)$$

where $A : \Gamma \rightarrow \mathbb{R}^{n_x \times n_x}$, and $u, b : \Gamma \rightarrow \mathbb{R}^{n_x}$. The system is parameterized by a set of stochastic input parameters $\xi(\omega) \equiv \{\xi_1(\omega), \dots, \xi_{n_\xi}(\omega)\}$. Here, $\omega \in \Omega$ is an elementary event in a probability space (Ω, \mathcal{F}, P) and the stochastic domain is denoted by $\Gamma \equiv \prod_{i=1}^{n_\xi} \Gamma_i$ where $\xi_i : \Omega \rightarrow \Gamma_i$. We are interested in computing solutions in finite-dimensional subspaces of $L^2(\Gamma)$ (defined below) using weak formulations of

(5.1) corresponding to Galerkin and Petrov–Galerkin projections.

Let $\rho \equiv \rho(\xi)$ be a density function defining an underlying measure of the stochastic space Γ and

$$\langle g, h \rangle_\rho \equiv \int_\Gamma g(\xi)h(\xi)\rho(\xi)d\xi, \quad (5.2)$$

$$E[g] \equiv \int_\Gamma g(\xi)\rho(\xi)d\xi, \quad (5.3)$$

define an inner product between scalar-valued functions $g(\xi)$ and $h(\xi)$ with respect to $\rho(\xi)$ and the expectation of $g(\xi)$, respectively. The inner product (5.2) also determines the Hilbert space $L^2(\Gamma)$. In addition, the ℓ^2 -norm of a vector-valued function $v(\xi) \in \mathbb{R}^{n_x}$ is defined as

$$\|v\|_2^2 \equiv \sum_{i=1}^{n_x} \int_\Gamma v_i^2(\xi)\rho(\xi)d\xi = E[v^T v]. \quad (5.4)$$

We are interested in computing approximate solutions to (5.1) using spectral methods, that is, finding solutions in an n_ψ -dimensional subspace S_{n_ψ} spanned by a finite set of polynomials $\{\psi_i(\xi)\}_{i=1}^{n_\psi}$ such that $S_{n_\psi} \equiv \text{span}\{\psi_i\}_{i=1}^{n_\psi} \subseteq L^2(\Gamma)$. Then

$$u(\xi) \approx \tilde{u}(\xi) = \sum_{i=1}^{n_\psi} \bar{u}_i \psi_i(\xi) = (\psi^T(\xi) \otimes I_{n_x}) \bar{u}, \quad (5.5)$$

where $\{\bar{u}_i\}_{i=1}^{n_\psi}$ with $\bar{u}_i \in \mathbb{R}^{n_x}$ are unknown coefficient vectors, $\bar{u} \equiv [\bar{u}_1^T \cdots \bar{u}_{n_\psi}^T]^T \in \mathbb{R}^{n_x n_\psi}$ is the vertical concatenation of these coefficient vectors, $\psi \equiv [\psi_1 \cdots \psi_{n_\psi}]^T \in \mathbb{R}^{n_\psi}$ is a concatenation of the polynomial basis, \otimes denotes the Kronecker prod-

uct, and I_{n_x} denotes the identity matrix of dimension n_x . Note that $\tilde{u} \in (S_{n_\psi})^{n_x}$. Typically, the “stochastic” basis $\{\psi_i\}$ consists of products of univariate polynomials: $\psi_i \equiv \psi_{\alpha(i)} \equiv \prod_{k=1}^{n_\xi} \pi_{\alpha_k(i)}(\xi_k)$ where $\{\pi_{\alpha_k(i)}\}_{k=1}^{n_\xi}$ are univariate polynomials, $\alpha(i) = (\alpha_1(i), \dots, \alpha_{n_\xi}(i)) \in \mathbb{N}_0^{n_\xi}$ is a multi-index and α_k represents the degree of a polynomial in ξ_k . The dimension of the stochastic subspace n_ψ depends on the number of random variables n_ξ , the maximum polynomial degree p , and a construction of the polynomial space (e.g., a total-degree space that contains polynomials with total degree up to p , $\sum_{k=1}^{n_\xi} \alpha_k(i) \leq p$). By substituting $u(\xi)$ with $\tilde{u}(\xi)$ in (5.1), the residual can be defined as

$$r(\bar{u}; \xi) := b(\xi) - A(\xi) \sum_{i=1}^{n_\psi} \bar{u}_i \psi_i(\xi) = b(\xi) - (\psi^T(\xi) \otimes A(\xi)) \bar{u}, \quad (5.6)$$

where $\psi^T(\cdot) \otimes A(\cdot) : \Gamma \rightarrow \mathbb{R}^{n_x \times n_\psi n_x}$.

It follows from (5.5) and (5.6) that our goal now is to compute the unknown coefficients $\{\bar{u}_i\}_{i=1}^{n_\psi}$ of the solution expansion. We briefly review two conventional approaches for doing so: the stochastic Galerkin method and the pseudo-spectral method. Typically, the polynomial basis is constructed to be orthogonal in the $\langle \cdot, \cdot \rangle_\rho$ inner product, i.e., $\langle \psi_i, \psi_j \rangle_\rho = \prod_{k=1}^{n_\xi} \langle \pi_{\alpha_k(i)}, \pi_{\alpha_k(j)} \rangle_{\rho_k} = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta.

5.2.2 Stochastic Galerkin method

The stochastic Galerkin method computes the unknown coefficients $\{\bar{u}_i\}_{i=1}^{n_\psi}$ of $\tilde{u}(\xi)$ in (5.5) by imposing orthogonality of the residual (5.6) with respect to the

inner product $\langle \cdot, \cdot \rangle_\rho$ in the subspace S_{n_ψ} . This Galerkin orthogonality condition can be expressed as follows: Find $\bar{u}^{\text{SG}} \in \mathbb{R}^{n_x n_\psi}$ such that

$$\langle r_i(\bar{u}^{\text{SG}}), \psi_j \rangle_\rho = E[r_i(\bar{u}^{\text{SG}})\psi_j] = 0, \quad i = 1, \dots, n_x, \quad j = 1, \dots, n_\psi, \quad (5.7)$$

where $r \equiv [r_1 \ \dots \ r_{n_x}]^T$. The condition (5.7) can be represented in matrix notation as

$$E[\psi \otimes r(\bar{u}^{\text{SG}})] = 0. \quad (5.8)$$

From the definition of the residual (5.6), this gives a system of linear equations

$$E[\psi\psi^T \otimes A]\bar{u}^{\text{SG}} = E[\psi \otimes b], \quad (5.9)$$

of dimension $n_x n_\psi$. This yields an algebraic expression for the stochastic-Galerkin approximation

$$\tilde{u}^{\text{SG}}(\xi) = (\psi(\xi)^T \otimes I_{n_x})E[\psi\psi^T \otimes A]^{-1}E[\psi \otimes Au]. \quad (5.10)$$

If $A(\xi)$ is symmetric positive definite, the solution of linear system (5.9) minimizes the solution error $e(x) \equiv u - x$ in the $A(\xi)$ -induced energy norm $\|v\|_A^2 \equiv E[v^T Av]$, i.e.,

$$\tilde{u}^{\text{SG}}(\xi) = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_A^2. \quad (5.11)$$

In general, however, the stochastic-Galerkin approximation does not minimize any measure of the solution error.

5.2.3 Pseudo-spectral method

The pseudo-spectral method directly approximates the unknown coefficients $\{\bar{u}_i\}_{i=1}^{n_\psi}$ of $\tilde{u}(\xi)$ in (5.5) by exploiting orthogonality of the polynomial basis $\{\psi_i(\xi)\}_{i=1}^{n_\psi}$. That is, the coefficients \bar{u}_i can be obtained by projecting the numerical solution $u(\xi)$ onto the orthogonal polynomial basis as

$$\bar{u}_i^{\text{PS}} = E[u\psi_i], \quad i = 1, \dots, n_\psi, \quad (5.12)$$

which can be expressed as

$$\bar{u}^{\text{PS}} = E[\psi \otimes A^{-1}b], \quad (5.13)$$

or equivalently

$$\tilde{u}^{\text{PS}}(\xi) = (\psi(\xi)^T \otimes I_{n_x})E[\psi \otimes u]. \quad (5.14)$$

The associated optimality property of the approximation, which can be derived from optimality of orthogonal projection, is

$$\tilde{u}^{\text{PS}}(\xi) = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_2^2. \quad (5.15)$$

In practice, the coefficients $\{\bar{u}_i^{\text{PS}}\}_{i=1}^{n_\psi}$ are approximated via numerical quadrature as

$$\bar{u}_i^{\text{PS}} = E[u\psi_i] = \sum_{k=1}^{n_q} u(\xi^{(k)})\psi_i(\xi^{(k)})w_k = \sum_{k=1}^{n_q} (A^{-1}(\xi^{(k)})b(\xi^{(k)})) \psi_i(\xi^{(k)})w_k, \quad (5.16)$$

where $\{(\xi^{(k)}, w_k)\}_{k=1}^{n_q}$ are the quadrature points and weights.

While stochastic Galerkin leads to an optimal approximation (5.11) under certain conditions and pseudo-spectral projection minimizes the ℓ^2 -norm of the solution error (5.15), neither approach provides the flexibility to tailor the optimality properties of the approximation. This may be important in applications where, for example, minimizing the error in a quantity of interest is desired. To address this, we propose a general optimization-based framework for spectral methods that enables the choice of a targeted weighted ℓ^2 -norm in which the solution error is minimized.

5.3 Stochastic least-squares Petrov–Galerkin method

As a starting point, we propose a residual-minimizing formulation that computes the coefficients \bar{u} by directly minimizing the ℓ^2 -norm of the residual, i.e.,

$$\tilde{u}^{\text{LSPG}}(\xi) = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|b - Ax\|_2^2 = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{A^T A}^2, \quad (5.17)$$

where $\|v\|_{A^T A}^2 \equiv E[v^T A^T A v]$. Thus, the ℓ^2 -norm of the residual is equivalent to a weighted ℓ^2 -norm of the solution error. Using (5.5) and (5.6), we have

$$\bar{u}^{\text{LSPG}} = \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|r(\bar{x})\|_2^2. \quad (5.18)$$

The definition of the residual (5.6) allows the objective function in (5.18) to be written in quadratic form as

$$\|r(\bar{x})\|_2^2 = \|b - (\psi^T \otimes A)\bar{x}\|_2^2 = \bar{x}^T E[\psi\psi^T \otimes A^T A]\bar{x} - 2E[\psi \otimes A^T b]^T \bar{x} + E[b^T b]. \quad (5.19)$$

Noting that the mapping $\bar{x} \mapsto \|r(\bar{x})\|_2^2$ is convex, the (unique) solution \bar{u}^{LSPG} to (5.18) is a stationary point of $\|r(\bar{x})\|_2^2$ and thus satisfies

$$E[\psi\psi^T \otimes A^T A]\bar{u}^{\text{LSPG}} = E[\psi \otimes A^T b], \quad (5.20)$$

which can be interpreted as the normal-equations form of the linear least-squares problem (5.18).

Consider a generalization of this idea that minimizes the solution error in a targeted weighted ℓ^2 -norm by choosing a specific weighting function. Let us define a weighting function $M(\xi) \equiv M_\xi(\xi) \otimes M_x(\xi)$, where $M_\xi : \Gamma \rightarrow \mathbb{R}$ and $M_x : \Gamma \rightarrow \mathbb{R}^{n_x \times n_x}$. Then, the stochastic LSPG method can be written as

$$\tilde{u}^{\text{LSPG}(M)}(\xi) = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|M(b - Ax)\|_2^2 = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{A^T M^T M A}^2, \quad (5.21)$$

with $\|v\|_{A^T M^T M A}^2 \equiv E[v^T A^T M^T M A v] = E[(M_\xi^T M_\xi \otimes (M_x A v)^T M_x A v)]$. Algebraically,

this is equivalent to

$$\begin{aligned}
\bar{u}^{\text{LSPG}(M)} &= \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|Mr(\bar{x})\|_2^2 = \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|(M_\xi \otimes M_x)(1 \otimes b - (\psi^T \otimes A) \bar{x})\|_2^2 \\
&= \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|M_\xi \otimes (M_x b) - ((M_\xi \psi^T) \otimes (M_x A)) \bar{x}\|_2^2.
\end{aligned} \tag{5.22}$$

We will restrict our attention to the case $M_\xi(\xi) = 1$ and denote $M_x(\xi)$ by $M(\xi)$ for simplicity. Now, the algebraic stochastic LSPG problem (5.22) simplifies to

$$\bar{u}^{\text{LSPG}(M)} = \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|Mr(\bar{x})\|_2^2 = \arg \min_{\bar{x} \in \mathbb{R}^{n_x n_\psi}} \|Mb - (\psi^T \otimes MA)\bar{x}\|_2^2. \tag{5.23}$$

The objective function in (5.23) can be written in quadratic form as

$$\|Mr(\bar{x})\|_2^2 = \bar{x}^T E[(\psi \psi^T \otimes A^T M^T M A)] \bar{x} - 2(E[\psi \otimes A^T M^T M f])^T \bar{x} + E[b^T M^T M b]. \tag{5.24}$$

As before, because the mapping $\bar{x} \mapsto \|Mr(\bar{x})\|_2^2$ is convex, the unique solution $\bar{u}^{\text{LSPG}(M)}$ of (5.23) corresponds to a stationary point of $\|Mr(\bar{x})\|_2^2$ and thus satisfies

$$E[\psi \psi^T \otimes A^T M^T M A] \bar{u}^{\text{LSPG}(M)} = E[\psi \otimes A^T M^T M f], \tag{5.25}$$

which is the normal-equations form of the linear least-squares problem (5.23). This yields the following algebraic expression for the stochastic-LSPG approximation:

$$\tilde{u}^{\text{LSPG}(M)}(\xi) = (\psi(\xi)^T \otimes I_{n_x}) E[\psi \psi^T \otimes A^T M^T M A]^{-1} E[\psi \otimes A^T M^T M A u]. \tag{5.26}$$

Petrov–Galerkin projection. Another way of interpreting the normal equations (5.25) is that the (weighted) residual $M(\xi)r(\bar{u}^{\text{LSPG}(M)}; \xi)$ is enforced to be orthogonal to the subspace spanned by the optimal test basis $\{\phi_i\}_{i=1}^{n_\psi}$ with $\phi_i(\xi) := \psi_i(\xi) \otimes M(\xi)A(\xi)$ and $\text{span}\{\phi_i\}_{i=1}^{n_\psi} \subseteq L^2(\Gamma)$. That is, this projection is precisely the (least-squares) Petrov–Galerkin projection,

$$E[\phi^T(b - (\psi^T \otimes MA)\bar{u}^{\text{LSPG}(M)})] = 0, \quad (5.27)$$

where $\phi(\xi) \equiv [\phi_1(\xi) \cdots \phi_{n_\psi}(\xi)]$.

Monotonic Convergence. The stochastic least-squares Petrov-Galerkin is monotonically convergent. That is, as the trial subspace S_{n_ψ} is enriched (by adding polynomials to the basis), the optimal value of the convex objective function $\|Mr(\bar{u}^{\text{LSPG}(M)})\|_2^2$ monotonically decreases. This is apparent from the LSPG optimization problem (5.21): Defining

$$\tilde{u}^{\text{LSPG}'(M)}(\xi) = \arg \min_{x \in (S_{n_\psi+1})^{n_x}} \|M(b - Ax)\|_2^2, \quad (5.28)$$

we have $\|M(b - A\tilde{u}^{\text{LSPG}'(M)})\|_2^2 \leq \|M(b - A\bar{u}^{\text{LSPG}(M)})\|_2^2$ (and $\|u - \tilde{u}^{\text{LSPG}'(M)}\|_{A^T M^T M A} \leq \|u - \bar{u}^{\text{LSPG}(M)}\|_{A^T M^T M A}$) if $S_{n_\psi} \subseteq S_{n_\psi+1}$.

Weighting strategies. Different choices of weighting function $M(\xi)$ allow LSPG to minimize different measures of the error. We focus on four particular choices:

1. $M(\xi) = C^{-1}(\xi)$, where $C(\xi)$ is a Cholesky factor of $A(\xi)$, i.e., $A(\xi) = C(\xi)C^T(\xi)$.

This decomposition exists if and only if A is symmetric positive semidefinite. In this case, LSPG minimizes the energy norm of the solution error $\|e(x)\|_A^2 \equiv \|C^{-1}r(\bar{x})\|_2^2$ ($= \|e((\Psi^T \otimes I_{n_x})\bar{x})\|_A^2$) and is mathematically equivalent to the stochastic Galerkin method described in Section 5.2.2, i.e., $\tilde{u}^{\text{LSPG}(C^{-1})} = \tilde{u}^{\text{SG}}$. This can be seen by comparing (5.11) and (5.21) with $M = C^{-1}$, as $A^T M^T M A = A$ in this case.

2. $M(\xi) = I_{n_x}$, where I_{n_x} is the identity matrix of dimension n_x . In this case, LSPG minimizes the ℓ^2 -norm of the residual $\|e(x)\|_{A^T A} \equiv \|r(\bar{x})\|_2^2$.
3. $M(\xi) = A^{-1}(\xi)$. In this case, LSPG minimizes the ℓ^2 -norm of solution error $\|e(x)\|_2^2 \equiv \|A^{-1}r(\bar{x})\|_2^2$. This is mathematically equivalent to the pseudo-spectral method described in Section 5.2.3, i.e., $\tilde{u}^{\text{LSPG}(A^{-1})} = \tilde{u}^{\text{PS}}$, which can be seen by comparing (5.15) and (5.21) with $M = A^{-1}$.
4. $M(\xi) = F(\xi)A^{-1}(\xi)$ where $F : \Gamma \rightarrow \mathbb{R}^{n_o \times n_x}$ is a linear functional of the solution associated with a vector of n_o output quantities of interest. In this case, LSPG minimizes the ℓ^2 -norm of the error in the output quantities of interest $\|Fe(x)\|_2^2 \equiv \|FA^{-1}r(\bar{x})\|_2^2$.

We again emphasize that two particular choices of the weighting function $M(\xi)$ lead to equivalence between LSPG and existing spectral-projection methods (stochastic Galerkin and pseudo-spectral projection), i.e.,

$$\tilde{u}^{\text{LSPG}(C^{-1})} = \tilde{u}^{\text{SG}}, \quad \tilde{u}^{\text{LSPG}(A^{-1})} = \tilde{u}^{\text{PS}}, \quad (5.29)$$

where the first equality is valid (i.e., the Cholesky decomposition $A(\xi) = C(\xi)C^T(\xi)$ can be computed) if and only if A is symmetric positive semidefinite. Table 5.1 summarizes the target quantities to minimize (i.e., $\|e(x)\|_{\Theta}^2 \equiv E[e(x)^T \Theta e(x)]$), the corresponding LSPG weighting functions, and the method names LSPG(Θ).

Table 5.1: Different choices for the LSPG weighting function.

Quantity minimized by LSPG		Weighting function	Method name
Quantity	Expression		
Energy norm of error	$\ e(x)\ _A^2$	$M(\xi) = C^{-1}(\xi)$	LSPG(A)/SG
ℓ^2 -norm of residual	$\ e(x)\ _{A^T A}^2$	$M(\xi) = I_{n_x}$	LSPG($A^T A$)
ℓ^2 -norm of solution error	$\ e(x)\ _2^2$	$M(\xi) = A^{-1}(\xi)$	LSPG(2)/PS
ℓ^2 -norm of error in quantities of interest	$\ F e(x)\ _2^2$	$M(\xi) = F(\xi)A^{-1}(\xi)$	LSPG($F^T F$)

5.4 Error analysis

If an approximation satisfies an optimal-projection condition

$$\tilde{u} = \arg \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{\Theta}^2, \quad (5.30)$$

then

$$\|e(\tilde{u})\|_{\Theta}^2 = \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{\Theta}^2. \quad (5.31)$$

Using norm equivalence

$$\|x\|_{\Theta'}^2 \leq C \|x\|_{\Theta}^2, \quad (5.32)$$

we can characterize the solution error $e(\tilde{u})$ in any alternative norm Θ' as

$$\|e(\tilde{u})\|_{\Theta'}^2 \leq C \min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{\Theta}^2. \quad (5.33)$$

Thus, the error in an alternative norm Θ' is controlled by the optimal objective-function value $\min_{x \in (S_{n_\psi})^{n_x}} \|e(x)\|_{\Theta}^2$ (which can be made small if the trial space admits accurate solutions) and the stability constant C .

Table 5.2 reports norm-equivalence constants for the norms considered in this work. Here, we have defined

$$\sigma_{\min}(M) \equiv \inf_{x \in (L^2(\Gamma))^{n_x}} \|Mx\|_2 / \|x\|_2, \quad \sigma_{\max}(M) \equiv \sup_{x \in (L^2(\Gamma))^{n_x}} \|Mx\|_2 / \|x\|_2. \quad (5.34)$$

Table 5.2: Stability constant C in (5.32).

	$\Theta' = A$	$\Theta' = A^T A$	$\Theta' = 2$	$\Theta' = F^T F$
$\Theta = A$	1	$\sigma_{\max}(A)$	$\frac{1}{\sigma_{\min}(A)}$	$\frac{\sigma_{\max}(F)^2}{\sigma_{\min}(A)}$
$\Theta = A^T A$	$\frac{1}{\sigma_{\min}(A)}$	1	$\frac{1}{\sigma_{\min}(A)^2}$	$\frac{\sigma_{\max}(F)^2}{\sigma_{\min}(A)^2}$
$\Theta = 2$	$\sigma_{\max}(A)$	$\sigma_{\max}(A)^2$	1	$\sigma_{\max}(F)^2$
$\Theta = F^T F$	$\frac{\sigma_{\max}(A)}{\sigma_{\min}(F)^2}$	$\frac{\sigma_{\max}(A)^2}{\sigma_{\min}(F)^2}$	$\frac{1}{\sigma_{\min}(F)^2}$	1

This exposes several interesting conclusions. First, if the number of output quantities of interest n_o is less than n_x , then the null space of F is nontrivial and so $\sigma_{\min}(F) = 0$. This implies that $\text{LSPG}(F^T F)$, for which $\Theta = F^T F$, will have an undefined value of C when the solution error is measured in other norms, i.e., for $\Theta' = A$, $\Theta' = A^T A$, and $\Theta' = 2$. It will have controlled errors only for $\Theta' = F^T F$, in

which case $C = 1$. Second, note that for problems with small $\sigma_{\min}(A)$, the ℓ^2 norm in the quantities of interest may be large for the LSPG(A)/SG, or LSPG($A^T A$), while it will remain well behaved for LSPG(2)/PS and LSPG($F^T F$).

5.5 Numerical experiments

This section explores the performance of the LSPG methods for solving elliptic SPDEs parameterized by one random variable (i.e., $n_\xi = 1$). The maximum polynomial degree used in the stochastic space S_{n_ψ} is p ; thus, the dimension of S_{n_ψ} is $n_\psi = p + 1$. In physical space, the SPDE is defined over a two-dimensional rectangular bounded domain D , and it is discretized using the finite element method with bilinear (Q_1) elements as implemented in the Incompressible Flow and Iterative Solver Software (IFISS) package [98]. Sixteen elements are employed in each dimension, leading to $n_x = 225 = 15^2$ degrees of freedom excluding boundary nodes. All numerical experiments are performed on an Intel 3.1 GHz i7 CPU, 16 GB RAM using MATLAB R2015a.

Measuring weighted ℓ^2 -norms. For all LSPG methods, the weighted ℓ^2 -norms can be measured by evaluating the expectations in the quadratic form of the objective function shown in (5.24). This requires evaluation of three expectations

$$\|Mr(\bar{x})\|_2^2 := \bar{x}^T T_1 \bar{x} - 2T_2^T \bar{x} + T_3, \quad (5.35)$$

with

$$T_1 := E[(\psi\psi^T \otimes A^T M^T M^T A)] \in \mathbb{R}^{n_x n_\psi \times n_x n_\psi}, \quad (5.36)$$

$$T_2 := E[\psi \otimes A^T M^T M b] \in \mathbb{R}^{n_x n_\psi}, \quad (5.37)$$

$$T_3 := E[b^T M^T M b] \in \mathbb{R}. \quad (5.38)$$

Note that T_3 does not depend on the stochastic-space dimension n_ψ . These quantities can be evaluated by numerical quadrature or analytically if closed-form expressions for those expectations exist. Unless otherwise specified, we compute these quantities using the `integral` function in MATLAB, which performs adaptive numerical quadrature based on the 15-point Gauss–Kronrod quadrature formula [95].

Error measures. In the experiments, we assess the error in approximate solutions computed using various spectral-projection techniques using four relative error measures (see Table 5.1):

$$\eta_r(x) := \frac{\|e(x)\|_{A^T A}^2}{\|b\|_2^2}, \quad \eta_e(x) := \frac{\|e(x)\|_2^2}{\|u\|_2^2}, \quad \eta_A(x) := \frac{\|e(x)\|_A^2}{\|u\|_A^2}, \quad \eta_Q(x) := \frac{\|Fe(x)\|_2^2}{\|Fu\|_2^2}. \quad (5.39)$$

5.5.1 Stochastic diffusion problems

Consider the steady-state stochastic diffusion equation with homogeneous boundary conditions,

$$\begin{cases} -\nabla \cdot (a(x, \xi) \nabla u(x, \xi)) = f(x, \xi) & \text{in } D \times \Gamma \\ u(x, \xi) = 0 & \text{on } \partial D \times \Gamma, \end{cases} \quad (5.40)$$

where the diffusivity $a(x, \xi)$ is a random field and $D = [0, 1] \times [0, 1]$. The random field $a(x, \xi)$ is specified as an exponential of a truncated Karhunen-Loève (KL) expansion [67] with covariance kernel, $C(x, y) \equiv \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{c} - \frac{|x_2 - y_2|}{c}\right)$, where c is the correlation length, i.e.,

$$a(x, \xi) \equiv \exp(\mu + \sigma a_1(x) \xi), \quad (5.41)$$

where $\{\mu, \sigma^2\}$ are the mean and variance of the KL expansion and $a_1(x)$ is the first eigenfunction in the KL expansion. After applying the spatial (finite-element) discretization, the problem can be reformulated as a parameterized linear system of the form (5.1), where $A(\xi)$ is a parameterized stiffness matrix obtained from the weak form of the problem whose (i, j) -element is $[A(\xi)]_{ij} = \int_D \nabla a(x, \xi) \varphi_i(x) \cdot \varphi_j(x) dx$ (with $\{\varphi_i\}$ standard finite element basis functions) and $b(\xi)$ is a parameterized right-hand side whose i th element is $[b(\xi)]_i = \int_D f(x, \xi) \varphi_i(x) dx$. Note that $A(\xi)$ is symmetric positive definite for this problem; thus LSPG(A)/SG is a valid projection scheme (the Cholesky factorization $A(\xi) = C(\xi)C(\xi)^T$ exists) and is equal to

stochastic Galerkin projection.

Output quantities of interest. We consider n_o output quantities of interest ($F(\xi)u(\xi) \in \mathbb{R}^{n_o}$) that are random linear functionals of the solution and $F(\xi)$ is of dimension $n_o \times n_x$ having the form:

(1) $F_1(\xi) := g(\xi) \times G$ with $G \in [0, 1]^{n_o \times n_x}$ a constant matrix: The elements of G are drawn from a uniform distribution (note that this is independent of the distribution $\rho(\xi)$) and $g(\xi)$ is a scalar-valued function of ξ . The resulting output QoI, $F_1(\xi)u(\xi)$, is a vector-valued function of dimension n_o .

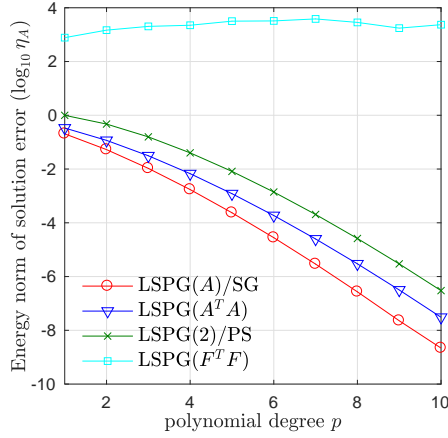
(2) $F_2(\xi) := b(\xi)^T \bar{M}$: \bar{M} is a mass matrix defined via $[\bar{M}]_{ij} \equiv \int_D \varphi_i(x)\varphi_j(x)dx$.

The output QoI is a scalar-valued function $F_2(\xi)u(\xi) = b(\xi)^T \bar{M}u(\xi)$, which approximates a spatial average $\frac{1}{|D|} \int_D f(x, \xi)u(x, \xi)dx$.

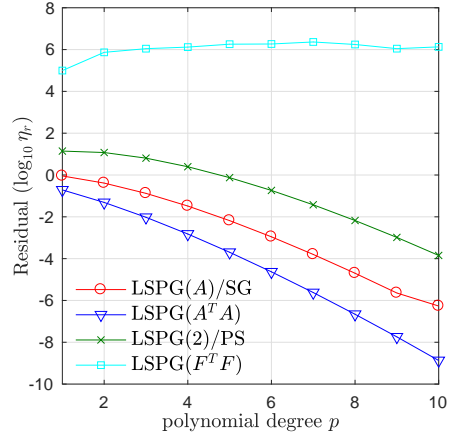
5.5.1.1 Diffusion problem 1: Lognormal random coefficient and deterministic forcing

In this example, we take ξ in (5.41) to follow a standard normal distribution (i.e., $\rho(\xi) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\xi^2}{2}\right)$ and $\xi \in (-\infty, \infty)$) and $f(x, \xi) = 1$ is deterministic. Because ξ is normally distributed, normalized Hermite polynomials (orthogonal with respect to $\langle \cdot, \cdot \rangle_\rho$) are used as polynomial basis $\{\psi_i(\xi)\}_{i=1}^{n_\psi}$.

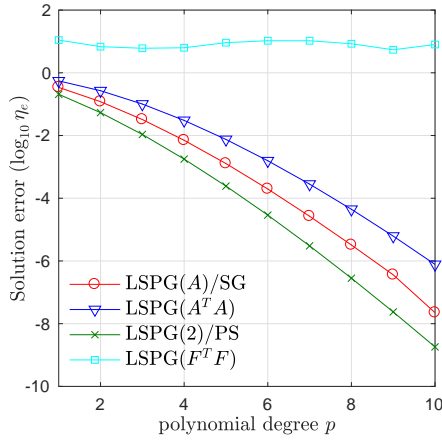
Figure 5.1 reports the relative errors (5.39) associated with solutions computed using four LSPG methods (LSPG(A)/SG, LSPG($A^T A$), LSPG(2)/PS, and LSPG($F^T F$)) for varying polynomial degree p . Here, we consider the random output QoI, i.e., $F = F_1$, $n_o = 100$, and $g(\xi) = \xi$. This result shows that three methods



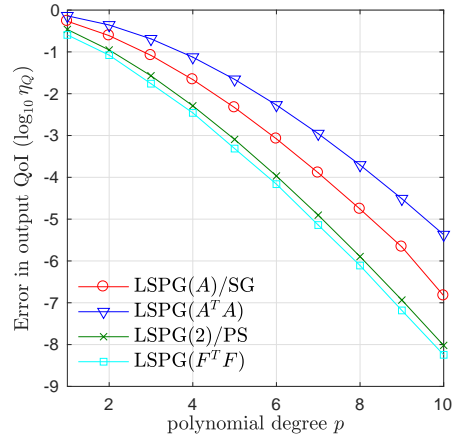
(a) Relative energy norm of solution error η_A



(b) Relative ℓ^2 -norm of residual η_r



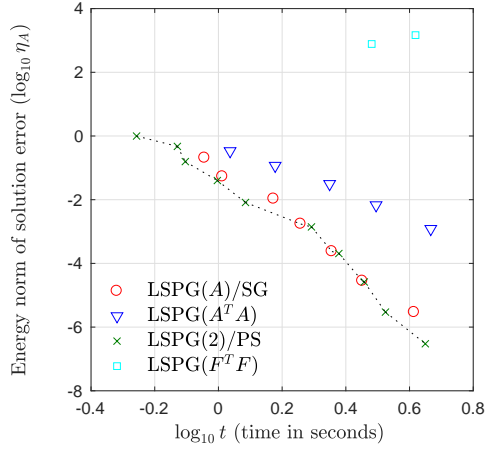
(c) Relative ℓ^2 -norm of solution error η_e



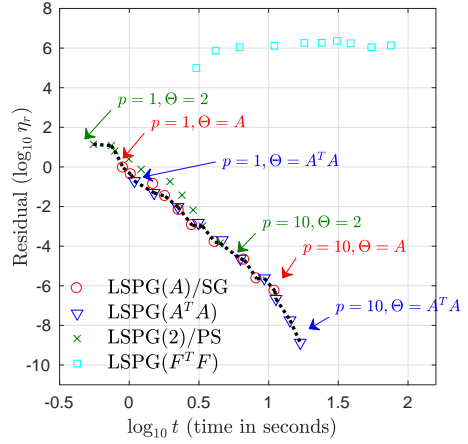
(d) Relative ℓ^2 -norm of output QoI error η_Q with $F = F_1$, $n_o = 100$, and $g(\xi) = \xi$

Figure 5.1: Relative error measures versus polynomial degree for diffusion problem 1: lognormal random coefficient and deterministic forcing. Note that each LSPG method performs best in the error measure it minimizes.

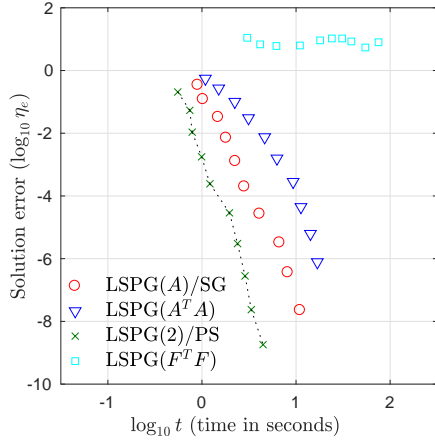
(LSPG(A)/SG, LSPG($A^T A$), and LSPG(2)/PS) monotonically converge in all four error measures, whereas LSPG($F^T F$) does not. This is an artifact of rank deficiency in F_1 , which leads to $\sigma_{\min}(F_1) = 0$; as a result, all stability constants C for which $\Theta = F^T F$ in Table 5.2 are unbounded, implying lack of error control. Figure 5.1 also shows that each LSPG method minimizes its targeted error measure for a given stochastic-subspace dimension (e.g., LSPG minimizes the ℓ^2 -norm of the residual);



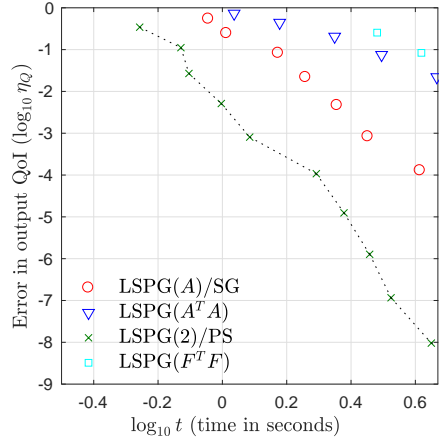
(a) Relative energy norm of solution error η_A



(b) Relative ℓ^2 -norm of residual η_r



(c) Relative ℓ^2 -norm of solution error η_e



(d) Relative ℓ^2 -norm of output QoI error η_Q with $F = F_1$, $n_o = 100$, and $g(\xi) = \xi$

Figure 5.2: Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 10 in increments of 1 going from left to right) for diffusion problem 1: lognormal random coefficient and deterministic forcing.

this is also evident from Table 5.2, as the stability constant realizes its minimum value ($C = 1$) for $\Theta = \Theta'$. Table 5.3 shows actual values of the stability constant of this problem and well explains the behaviors of all LSPG methods. For example, the first column of Table 5.3 shows that the stability constant is increasing in the order (LSPG(A)/SG, LSPG($A^T A$), LSPG(2)/PS, and LSPG($F^T F$)), which is represented in Figure 5.1(a).

Table 5.3: Stability constant C of Diffusion problem 1.

	$\Theta' = A$	$\Theta' = A^T A$	$\Theta' = 2$	$\Theta' = F^T F$
$\Theta = A$	1	26.43	2.06	11644.22
$\Theta = A^T A$	2.06	1	4.25	24013.48
$\Theta = 1$	26.43	698.53	1	5646.32
$\Theta = F^T F$	∞	∞	∞	1

The results in Figure 5.1 do not account for computational costs. This point is addressed in Figure 5.2, which shows the relative errors as a function of CPU time. As we would like to devise a method that minimizes both the error and computational time, we examine a Pareto front (black dotted line), that is, a curve identifying the methods that minimize the two competing objectives considered in the figure. This typically corresponds to LSPG(2)/PS. This is because this method does not require solution of a coupled system of linear equations of dimension $n_x n_\psi$, which is required by the other three LSPG methods (LSPG(A)/SG, LSPG($A^T A$), and LSPG($F^T F$)). As a result, pseudo-spectral projection (LSPG(2)/PS) generally yields the best overall performance in practice, even when it produces larger errors than other methods for a fixed value of p . Also, for a fixed value of p , LSPG(A)/SG is faster than LSPG($A^T A$) because the weighted stiffness matrix $A(\xi)$ obtained from the finite element discretization is sparser than $A^T(\xi)A(\xi)$. That is, the number of nonzero entries to be evaluated for LSPG(A)/SG in numerical quadrature is smaller than the ones for LSPG($A^T A$), and exploiting this sparsity structure in the numerical quadrature causes LSPG(A)/SG to be faster than LSPG($A^T A$). Also, note that there are cases (Figure 5.2(b)) where the Pareto front does not correspond to a single method; this outcome will occur with other benchmark problems considered

below.

5.5.1.2 Diffusion problem 2: Lognormal random coefficient and random forcing

This example uses the same random field $a(x, \xi)$ (5.41), but instead employs a random forcing term¹ $f(x, \xi) = \exp(\xi)|\xi - 1|$. Again, ξ follows a standard normal distribution and normalized Hermite polynomials are used as polynomial basis. We consider the second output QoI, $F = F_2$. As shown in Figure 5.3, the stochastic Galerkin method fails to converge monotonically in three error measures as the stochastic polynomial basis is enriched. In fact, it exhibits monotonic convergence only in the error measure it minimizes (for which monotonic convergence is guaranteed).

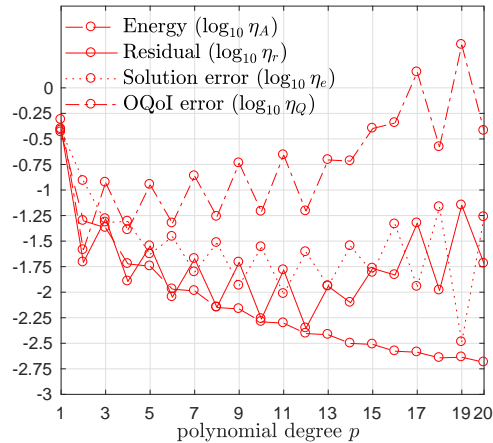
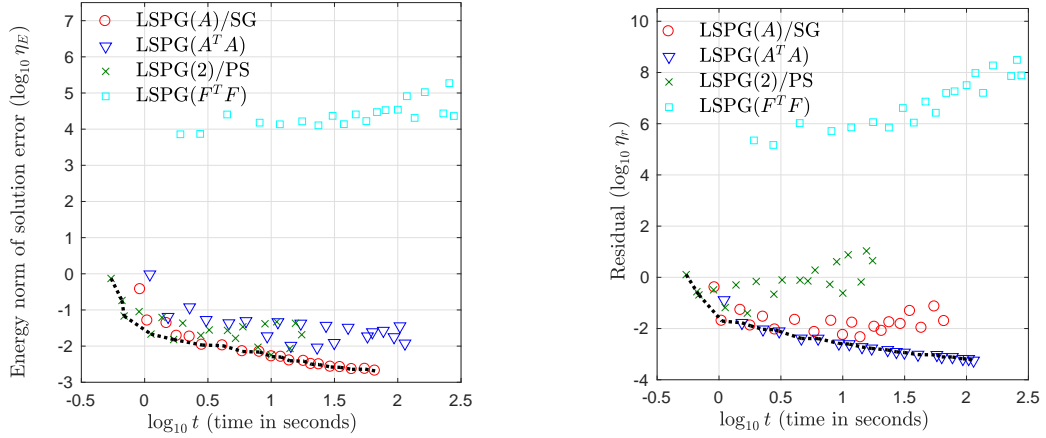


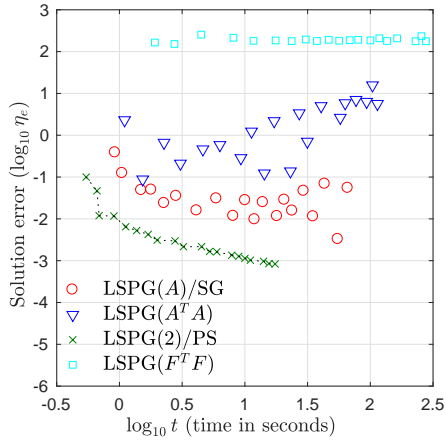
Figure 5.3: Relative errors versus polynomial degree for stochastic Galerkin (i.e., LSPG(A)/SG) for diffusion problem 2: lognormal random coefficient and random forcing. Note that monotonic convergence is observed only in the minimized error measure η_A .

¹In [73], it was shown that stochastic Galerkin solutions of an analytic problem $a(\xi)u(\xi) = f(\xi)$ with this type of forcing are divergent in the ℓ^2 -norm of solution errors as p increases.

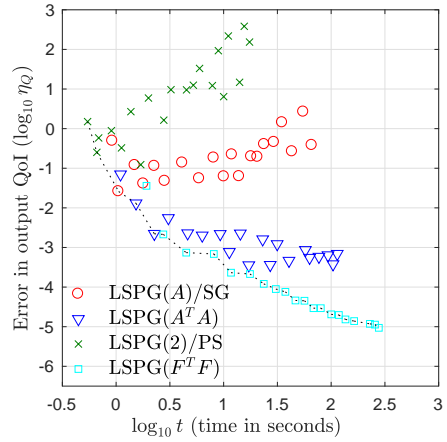


(a) Relative energy norm of solution error η_A

(b) Relative ℓ^2 -norm of residual η_r



(c) Relative ℓ^2 -norm of solution error η_e



(d) Relative ℓ^2 -norm of output QoI error η_Q with $F = F_2$

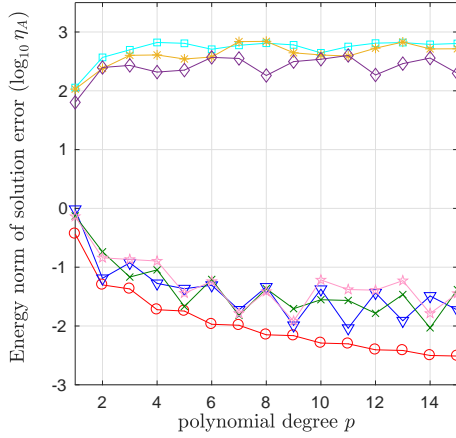
Figure 5.4: Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 2: lognormal random coefficient and random forcing

Figure 5.4 shows that this trend applies to other methods as well when effectiveness is viewed with respect to CPU time; each technique exhibits monotonic convergence in its tailored error measure only. Moreover, the Pareto fronts (black dotted lines) in each subgraph of Figure 5.4 shows that the LSPG method tailored for a particular error measure is Pareto optimal in terms of minimizing the error and computational wall time. In the next experiments, we examine

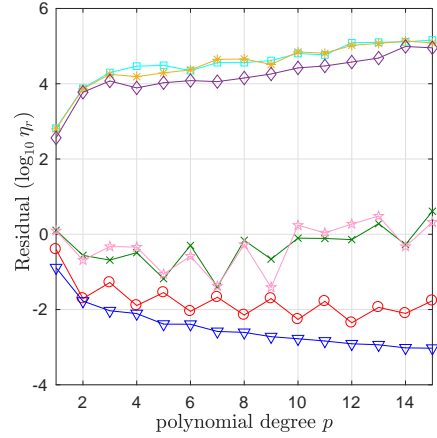
goal-oriented LSPG($F^T F$) for varying number of output quantities of interest n_o and its effect on the stability constant C . Figure 5.5 reports three error measures computed using all four LSPG methods. For LSPG($F^T F$), the first linear function $F = F_1$ is applied with $g(\xi) = \sin(\xi)$ and a varying number of outputs $n_o = \{100, 150, 200, 225\}$. When $n_o = 225$, LSPG($F^T F$) and LSPG(2)/PS behave similarly in all three weighted ℓ^2 -norms. This is because when $n_o = 225 = n_x$, then $\sigma_{\min}(F) > 0$, so the stability constants C for $\Theta = F^T F$ in Table 5.2 are bounded. Figure 5.6 reports relative errors in the quantity of interest η_Q associated with linear functionals $F = F_1$ for two different functions $g(\xi)$, $g_1(\xi) = \sin(\xi)$ and $g_2(\xi) = \xi$. Note that LSPG(A)/SG and LSPG($A^T A$) fail to converge, whereas LSPG(2)/PS and LSPG($F^T F$) converge, which can be explained by the stability constant C in Table 5.2 where $\sigma_{\max}(A) = 26.43$ and $\sigma_{\min}(A) = 0.48$ for the linear operator $A(\xi)$ of this problem. LSPG($F^T F$) converges monotonically and produces the smallest error (for a fixed polynomial degree p) of all the methods as expected.

5.5.1.3 Diffusion problem 3: Gamma random coefficient and random forcing

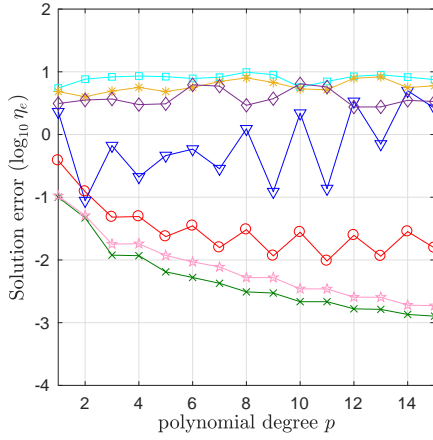
This section considers a stochastic diffusion problem parameterized by a random variable that has a Gamma distribution, where $a(x, \xi) \equiv \exp(1 + 0.25a_1(x)\xi + 0.01 \sin(\xi))$ with density $\rho(\xi) \equiv \frac{\xi^\alpha \exp(-\xi)}{\Gamma(\alpha+1)}$, $\bar{\Gamma}$ is the Gamma function, $\xi \in [0, \infty)$, and $\alpha = 0.5$. Normalized Laguerre polynomials (which are orthogonal with respect to $\langle \cdot, \cdot \rangle_\rho$) are used as polynomial basis. We consider a random forcing term



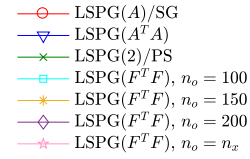
(a) Relative energy norm of solution error η_A



(b) Relative ℓ^2 -norm of residual η_r



(c) Relative ℓ^2 -norm of solution error η_e

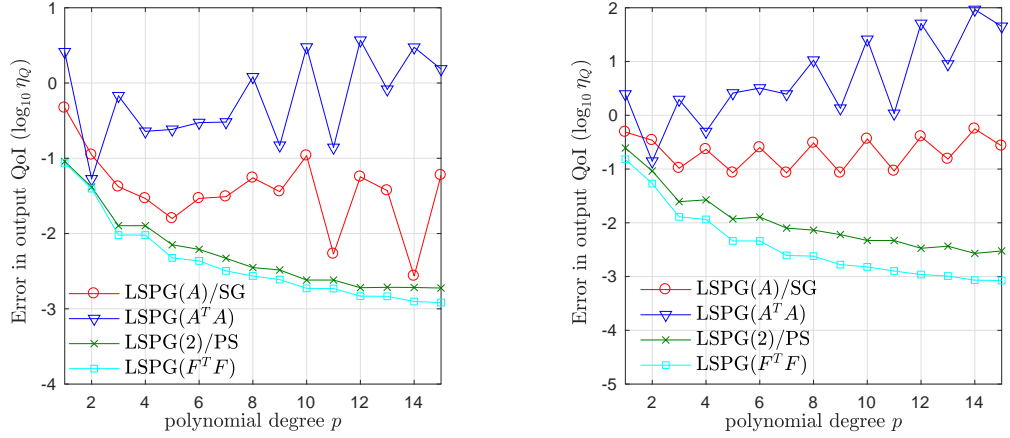


(d) Legend for subplots (b)–(d)

Figure 5.5: Relative error measures versus polynomial degree for a varying dimension n_o of the output matrix $F = F_1$ for diffusion problem 2: lognormal random coefficient and random forcing. Note that LSPG($F^T F$) has controlled errors only when $n_o = n_x$, in which case $\sigma_{\min}(F) > 0$.

$f(x, \xi) = \log_{10}(\xi)|\xi - 1|$ and the second QoI $F(\xi) = F_2(\xi) = b(\xi)^T \bar{M}$. Note that numerical quadrature is the only option for computing expectations arise in this problem.

Figure 5.7 shows the results of solving the problem with the four different LSPG methods. Again, each version of LSPG monotonically decreases its corresponding target weighted ℓ^2 -norm as the stochastic basis is enriched. Further, each



(a) Relative ℓ^2 -norm of output QoI error η_Q (b) Relative ℓ^2 -norm of output QoI error η_Q with $F = F_1$, $n_o = 100$, and $g(\xi) = g_1(\xi) = \sin(\xi)$ with $F = F_1$, $n_o = 100$, and $g(\xi) = g_2(\xi) = \xi \sin(\xi)$

Figure 5.6: Plots of the error norm of output QoI for diffusion problem 2: lognormal random coefficient and random forcing when a linear functional is (a) $F(\xi) \equiv \sin(\xi) \times [0, 1]^{100 \times n_x}$ and (b) $F(\xi) = \xi \times [0, 1]^{100 \times n_x}$ for varying p and varying n_o .

LSPG method is Pareto optimal in terms of minimizing its targeted error measure and the computational wall time.

5.5.2 Stochastic convection-diffusion problem: Lognormal random coefficient and deterministic forcing

We now consider a non-self-adjoint example, the steady-state convection-diffusion equation

$$\begin{cases} -\epsilon \nabla \cdot (a(x, \xi) \nabla u(x, \xi)) + \vec{w} \cdot \nabla u(x, \xi) = f(x, \xi) & \text{in } D \times \Gamma, \\ u(x, \xi) = g_D(x) & \text{on } \partial D \times \Gamma \end{cases} \quad (5.42)$$

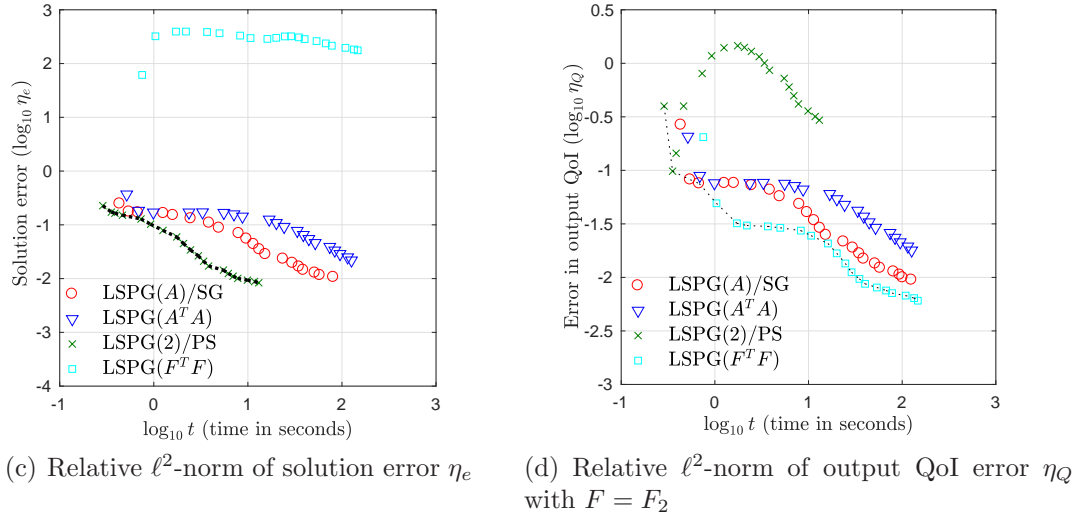
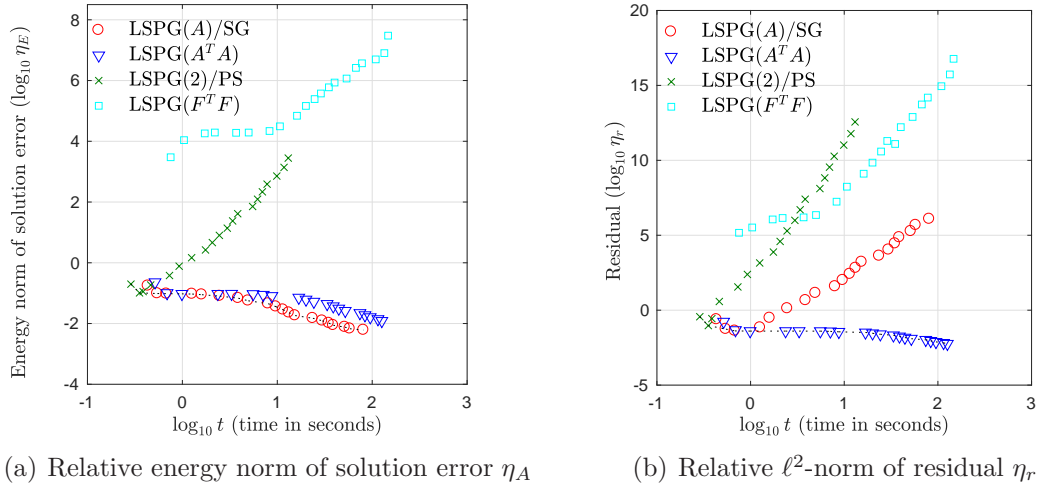
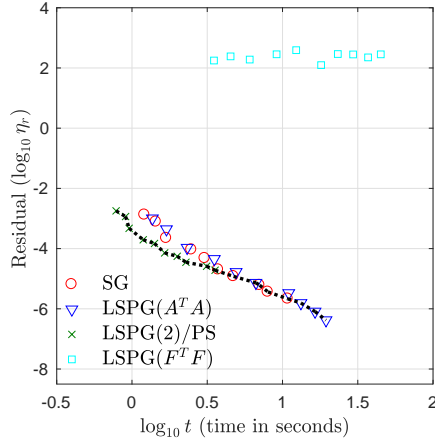


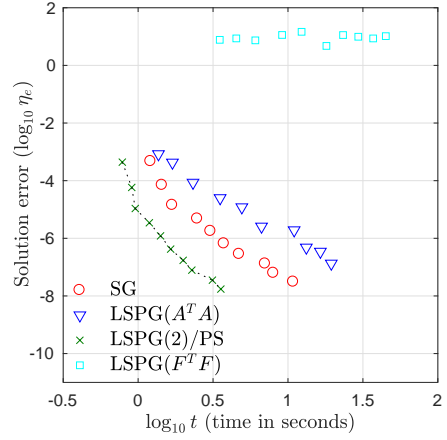
Figure 5.7: Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 3: Gamma random coefficient and random forcing. Note that each method is Pareto optimal in terms of minimizing its targeted error measure and computational wall time.

where $D = [-1, 1] \times [-1, 1]$, ϵ is the viscosity parameter, and u satisfies inhomogeneous Dirichlet boundary conditions

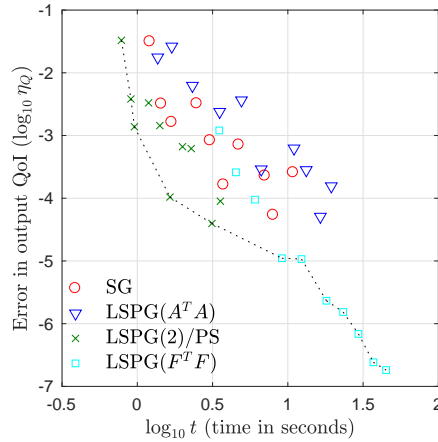
$$g_D(x) = \begin{cases} g_D(x, 1) = 0 & \text{for } [-1, y] \cup [x, 1] \cup [-1 \leq x \leq 0, -1], \\ g_D(1, y) = 1 & \text{for } [1, y] \cup [0 \leq x \leq 1, -1]. \end{cases} \quad (5.43)$$



(a) Relative ℓ^2 -norm of residual η_r



(b) Relative ℓ^2 -norm of solution error η_e



(c) Relative ℓ^2 -norm of output QoI error η_Q
with $F = F_1$, $n_o = 100$, $g(\xi) = \exp(\xi)|\xi - 1|$

Figure 5.8: Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 10 in increments of 1 going from left to right) for stochastic convection-diffusion problem: lognormal random coefficient and deterministic forcing term.

The inflow boundary consists of the bottom and the right portions of ∂D , $[x, -1] \cup [1, y]$ [39]. We consider a zero forcing term $f(x, \xi) = 0$ and a constant convection velocity $\vec{w} \equiv (-\sin \frac{\pi}{6}, \cos \frac{\pi}{6})$. We consider the convection-dominated case (i.e., $\epsilon = \frac{1}{200}$).

For the spatial discretization, we essentially use the same finite element as

above (bilinear Q_1 elements) applied to the weak formulation of (5.42). In addition, we use the streamline-diffusion method [17] to stabilize the discretization in elements with large mesh Peclet number. (See [39], Ch. 8 for details.) Such spatial discretization leads to a parameterized linear system of the form (5.1) with

$$A(\xi) = \epsilon D(a(\xi); \xi) + C(\xi) + S(\xi), \quad (5.44)$$

where $D(a(\xi); \xi)$, $C(\xi)$ and $S(\xi)$ are the diffusion term, the convection term, and the streamline-diffusion term, respectively, and $[b(\xi)]_i = \int_D f(x, \xi) \varphi_i(x) dx$. For this numerical experiment, the number of degrees of freedom in spatial domain is $n_x = 225$ (15 nodes in each spatial dimension) excluding boundary nodes. For $\text{LSPG}(F^T F)$, the first linear function $F = F_1$ is applied with $n_o = 100$ outputs and $g(\xi) = \exp(\xi)|\xi - 1|$.

Figure 5.8 shows the numerical results computed using the stochastic Galerkin method and three LSPG methods ($\text{LSPG}(A^T A)$, $\text{LSPG}(2)/\text{PS}$, $\text{LSPG}(F^T F)$). Note that the operator $A(\xi)$ is not symmetric positive-definite in this case; thus $\text{LSPG}(A)$ is not a valid projection scheme (the Cholesky factorization $A(\xi) = C(\xi)C(\xi)^T$ does not exist and the energy norm of the solution error $\|e(x)\|_A^2$ cannot be defined) and stochastic Galerkin does not minimize an any measure of the solution error. These results show that pseudo-spectral projection is Pareto optimal for achieving relatively larger error measures; this is because of its relatively low cost since, in contrast to the other methods, it does not require the solution of a coupled linear system of dimension $n_x n_\psi$. In addition, the stochastic Galerkin projection is not

Pareto optimal for any of the examples; this is caused by the lack of optimality of stochastic Galerkin in this case and highlights the significant benefit of optimal spectral projection, which is offered by the stochastic LSPG method. In addition, the residual η_r and solution error η_e incurred by $\text{LSPG}(F^T F)$ are uncontrolled, because $n_o < n_x$ and thus $\sigma_{\min}(F) = 0$. Finally, note that each LSPG method is Pareto optimal for small errors in its targeted error measure.

5.5.3 Numerical experiment with analytic computations

For the results presented above, expected values were computed using numerical quadrature (using the MATLAB function `integral`). This is a practical and general approach for numerically computing the required integrals of (5.36)–(5.38), and is the only option when analytic computations are not available (as in Section 5.5.1.3). In this section, we briefly discuss how the costs change if analytic methods based on closed-form integration exist and are used for these integrals. Note that in general, however, analytic computation are unavailable, for example, if the random variables have a finite support (e.g., truncated Gaussian random variables as shown in [106]).

Computing T_1 . Analytic computation of T_1 is possible if either $E[A^T M M A \psi_l]$ or $E[M A \psi_l]$ can be evaluated analytically. For $\text{LSPG}(A)/\text{SG}$ and $\text{LSPG}(A^T A)$, if $E[A \psi_l]$ can be evaluated so that the following gPC expansion can be obtained analytically

$$A(\xi) = \sum_{l=1}^{\infty} A_l \psi_l(\xi), \quad A_l \equiv E[A \psi_l], \quad (5.45)$$

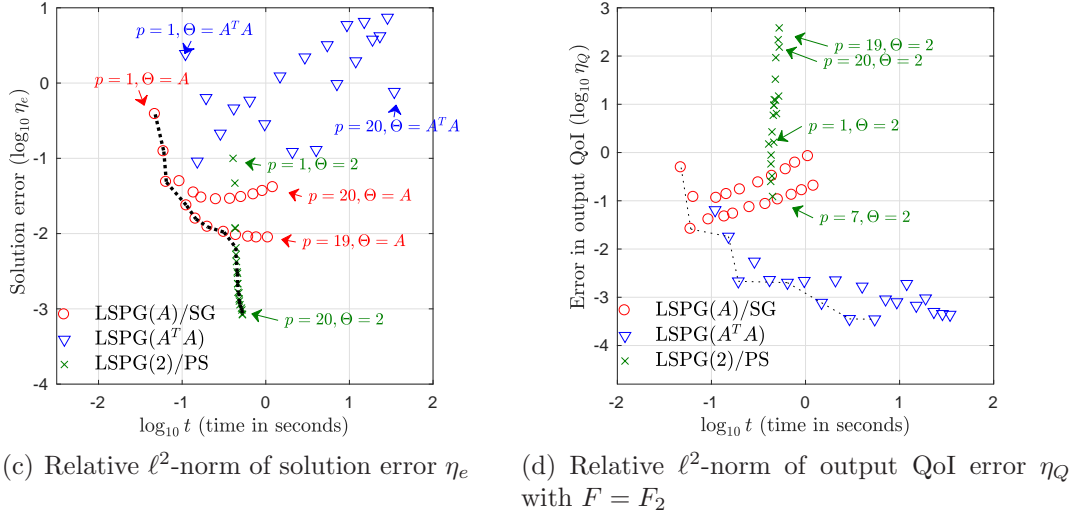
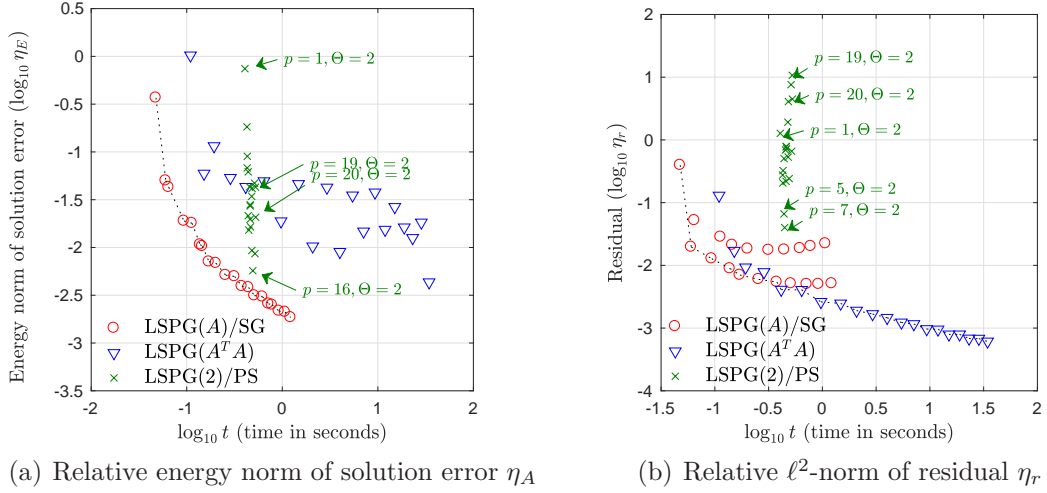


Figure 5.9: Pareto front of relative error measures versus wall time for varying polynomial degree p (p varies from 1 to 20 in increments of 1 going from left to right) for diffusion problem 2: Lognormal random coefficient and random forcing. Analytic computations are used as much as possible to evaluate expectations.

where $A_l \in \mathbb{R}^{n_x \times n_x}$, then T_1 can be computed analytically. Replacing $A(\xi)$ with the series of (5.45) for LSPG(A)/SG ($M(\xi) = C^{-1}(\xi)$) and LSPG($A^T A$) ($M(\xi) = I_{n_x}$) yields

$$T_1^{\text{LSPG}(A)} = \sum_{l=1}^{n_a} E[\psi \psi^T \otimes (A_l \psi_l)] = \sum_{l=1}^{n_a} E[\psi \psi^T \psi_l \otimes A_l], \quad (5.46)$$

and

$$T_1^{\text{LSPG}(A^T A)} = E[\psi\psi^T \otimes \sum_{k=1}^{n_a} \sum_{l=1}^{n_a} (A_k\psi_k)^T (A_l\psi_l)] = \sum_{k=1}^{n_a} \sum_{l=1}^{n_a} E[\psi\psi^T \psi_k\psi_l \otimes A_k^T A_l], \quad (5.47)$$

where the expectations of triple or quadruple products of the polynomial basis (i.e., $E[\psi_i\psi_j\psi_k]$ and $E[\psi_i\psi_j\psi_k\psi_l]$) can be computed analytically. For LSPG(2)/PS, an analytic computation of T_1 is straightforward because $M(\xi)A(\xi) = I_{n_x}$ and, thus,

$$T_1^{\text{LSPG}(2)} = E[\psi\psi^T \otimes I_{n_x}] = I_{n_x n_\psi}. \quad (5.48)$$

Similarly, analytic computation of T_1 is possible for LSPG($F^T F$) if there exists a closed formulation for $E[F\psi_l]$ or $E[F^T F\psi_l]$, which is again in general not available.

Computing T_2 . Analytic computation of T_2 can be performed in a similar way. If the random function $b(\xi)$ can be represented using a gPC expansion,

$$b(\xi) = \sum_{l=1}^{n_b} b_l \psi_l(\xi), \quad b_l \equiv E[b\psi_l], \quad (5.49)$$

then, for LSPG(A)/SG and LSPG($A^T A$), T_2 can be evaluated analytically by computing expectations of bi or triple products of the polynomial bases (i.e., $E[\psi_i\psi_j]$ and $E[\psi_i\psi_j\psi_k]$). For LSPG(2)/PS and LSPG($F^T F$), however, an analytic computation of T_2 is typically unavailable because a closed-form expression for $A^{-1}(\xi)$ does not exist.

We examine the impact of these observations on the cost of solution of the

problem studied in Section 5.5.1.2, a the steady-state stochastic diffusion equation (5.40) with lognormal random field $a(x, \xi)$ as in (5.41), and random forcing $f(x, \xi) = \exp(\xi)|\xi - 1|$.

Figure 5.9 reports results for this problem for analytic computation of expectations. For LSPG(A)/SG, analytic computation of the expectations $\{T_i\}_{i=1}^3$ requires fewer terms than for LSPG($A^T A$). In fact, comparing (5.46) and (5.47) shows that computing $T_1^{\text{LSPG}(A^T A)}$ requires computing and assembling n_a^2 terms, whereas computing $T_1^{\text{LSPG}(A)}$ involves only n_a terms. Additionally the quantities $\{A_k^T A_l\}_{k,l=1}^{n_a}$ appearing in the terms of $T_1^{\text{LSPG}(A^T A)}$ in (5.47) are typically denser than the counterparts $\{A_k\}_{k=1}^{n_a}$ appearing in (5.46), as the sparsity pattern of $\{A_k\}_{k=1}^{n_a}$ is identical to that of the finite element stiffness matrices. As a result, LSPG(A)/SG is Pareto optimal for small computational wall times when any error metric is considered. When the polynomial degree p is small, LSPG(A)/SG is computationally faster than LSPG(2)/PS, as LSPG(2)/PS requires the solution of $A(\xi^{(k)})u(\xi^{(k)}) = f(\xi^{(k)})$ at each quadrature point and cannot exploit analytic computation. As the stochastic basis is enriched, however, each tailored LSPG method outperforms other LSPG methods in minimizing its corresponding target error measure.

5.6 Conclusion

In this work, we have proposed a general framework for optimal spectral projection wherein the solution error can be minimized in weighted ℓ^2 -norms of interest. In particular, we propose two new methods that minimize the ℓ^2 -norm of the resid-

ual (LSPG($A^T A$)) and the ℓ^2 -norm of the error in an output quantity of interest (LSPG($F^T F$)). Further, we showed that when the linear operator is symmetric positive definite, stochastic Galerkin is a particular instance of the proposed methodology for a specific choice of weighted ℓ^2 -norm. Similarly, pseudo-spectral projection is a particular case of the method for a specific choice of weighted ℓ^2 -norm.

Key results from the numerical experiments include:

- For a fixed stochastic subspace, each LSPG method minimizes its targeted error measure (Figure 5.1).
- For a fixed computational cost, each LSPG method often minimizes its targeted error measure (Figures 5.4, 5.7). However, this does not always hold, especially for smaller computational costs (and smaller stochastic-subspace dimensions) when larger errors are acceptable. In particular pseudo-spectral projection (LSPG(2)/PS) is often significantly less expensive than other methods for a fixed stochastic subspace, as it does not require solving a coupled linear system of dimension $n_x n_\psi$ (Figures 5.2, 5.8). Alternatively, when analytic computations are possible, stochastic Galerkin (LSPG(A)/SG)) may be significantly less expensive than other methods for a fixed stochastic subspace (Figure 5.9).
- Goal-oriented LSPG($F^T F$) can have uncontrolled errors in error measures that deviate from the output-oriented error measure η_Q when the linear operator F has more columns n_x than rows n_o (Figure 5.5). This is because the minimum singular value is zero in this case (i.e., $\sigma_{\min}(F) = 0$), which leads to unbounded

stability constants in other error measures (Table 5.2).

- Stochastic Galerkin often leads to divergence in different error measures (Figure 5.3). In this case, applying LSPG with the appropriate targeted error measure can significantly improve accuracy (Figure 5.4).

Future work includes developing efficient sparse solvers for the stochastic LSPG methods and extending the methods to parameterized nonlinear systems.

Chapter 6: Conclusion

In this thesis, we proposed solution algorithms for addressing two difficulties in using the stochastic Galerkin method for solving high-dimensional parameterized PDEs: (1) the solution of the Galerkin systems are computationally expensive and (2) the stochastic Galerkin method does not always guarantee optimality in the solution error. For efficient computations, we proposed the two-level low-rank iterative solver for linear elliptic parameterized PDEs and the low-rank variant of the Newton–Krylov method for nonlinear parameterized PDEs. For optimality, we proposed the stochastic least-squares Petrov–Galerkin method. We examined the efficiency and the optimality of the proposed methods on several benchmark problems.

In Chapter 3, we presented the two-level low-rank iterative solver for linear elliptic parameterized PDEs, which identifies an important low-dimensional subspace with a coarse-grid computation and uses the identified subspace for truncating all intermediate quantities generated during the low-rank GMRES iteration on the fine-grid space. In the low-rank GMRES method, computational efficiency was achieved by using the matrix operations, which exploits the Kronecker-product structure. Numerical experiments on two benchmark problems, a stochastic diffusion prob-

lem and a stochastic convection-diffusion problem, demonstrated that the two-level algorithm achieved significant savings in computational costs.

In Chapter 4, we presented a low-rank variant of the Newton–Krylov method for solving the Navier–Stokes equations with uncertain viscosity. We adapted the hybrid linearization scheme, which employs a few steps of Picard iterations followed by the Newton iterations, to the low-rank variant of the nonlinear iteration. To further achieve computational savings, we consider the inexact version of the nonlinear iteration, which approximately solves the linear system at each nonlinear step. We demonstrated the performance of the proposed method with the set of benchmark problems with various configurations characterizing the statistical features of the uncertain viscosity. The numerical experiments showed that the proposed method achieved significant computational savings for the problems with smaller CoV and larger correlation lengths.

In Chapter 5, we presented the stochastic least-squares Petrov–Galerkin method, which produces an optimal solution in a given finite-dimensional subspace minimizing the solution error in a target norm. We showed that specific choices of the weighting function lead to certain minimization formulations that are mathematically equivalent to the stochastic Galerkin method and the pseudo-spectral method. The method is monotonic convergent in the sense that the method produces monotonically decreasing solution error in a target norm. Using extensive numerical experiments on benchmark problems, we demonstrated that each LSPG method is optimal in minimizing its targeted error measure and is optimal also in terms of computational costs when an accurate solution in a target error measure is sought.

Bibliography

- [1] I. Babuška and P. Chatzipantelidis. On solving elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 191(37):4093–4122, 2002.
- [2] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.
- [3] I. Babuška, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.
- [4] M. Bachmayr and A. Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Mathematics of Computation*, 86(304):701–724, 2017.
- [5] M. Bachmayr, A. Cohen, and W. Dahmen. Parametric PDEs: Sparse or low-rank approximations? *arXiv preprint arXiv:1607.04444*, 2016.
- [6] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013.
- [7] J. Beck, F. Nobile, L. Tamellini, and R. Tempone. Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients. *Computers and Mathematics with Applications*, 67(4):732–751, 2014.
- [8] J. Beck, R. Tempone, F. Nobile, and L. Tamellini. On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. *Mathematical Models and Methods in Applied Sciences*, 22(09):1250023.1–1250023.33, 2012.
- [9] P. Benner, S. Dolgov, A. Onwunta, and M. Stoll. Solving optimal control problems governed by random Navier-Stokes equations using low-rank methods. *arXiv preprint arXiv:1703.06097*, 2017.

- [10] P. Benner, A. Onwunta, and M. Stoll. Low-rank solution of unsteady diffusion equations with stochastic coefficients. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):622–649, 2015.
- [11] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.
- [12] G. Blatman and B. Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25(2):183–197, 2010.
- [13] P. B. Bochev and M. D. Gunzburger. Finite element methods of least-squares type. *SIAM review*, 40(4):789–837, 1998.
- [14] P. B. Bochev and M. D. Gunzburger. *Least-Squares Finite Element Methods*, volume 166. Springer, New York, 2009.
- [15] C. Brezinski and M. Redivo-Zaglia. The PageRank vector: properties, computation, approximation, and acceleration. *SIAM Journal on Matrix Analysis and Applications*, 28(2):551–575, 2006.
- [16] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer, New York, 2012.
- [17] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.
- [18] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.
- [19] K. Carlberg, C. Farhat, and C. Bou-Mosleh. Efficient nonlinear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, October 2011.
- [20] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.
- [21] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition. *Psychometrika*, 35(3):283–319, 1970.
- [22] J. Chung and J. G. Nagy. Nonlinear least squares and super resolution. *Journal of Physics: Conference Series*, 124(1):012019:1–012019:10, 2008.

- [23] A. Cohen, R. DeVore, and C. Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Foundations of Computational Mathematics*, 10(6):615–646, 2010.
- [24] A. Cohen, R. Devore, and C. Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE’s. *Analysis and Applications*, 9(01):11–47, 2011.
- [25] P. G. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.
- [26] P. G. Constantine and D. F. Gleich. Using polynomial chaos to compute the influence of multiple random surfers in the PageRank model. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 82–95. Springer, 2007.
- [27] S. Corveleyn, E. Rosseel, and S. Vandewalle. Iterative solvers for a spectral Galerkin approach to elliptic partial differential equations with fuzzy coefficients. *SIAM Journal on Scientific Computing*, 35(5):S420–S444, 2013.
- [28] M. K. Deb, I. Babuška, and J. T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.
- [29] B. J. Deusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2):698–719, 2004.
- [30] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.
- [31] A. Doostan, R. G. Ghanem, and J. Red-Horse. Stochastic model reduction for chaos representations. *Computer Methods in Applied Mechanics and Engineering*, 196(37):3951–3966, 2007.
- [32] A. Doostan and H. Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.
- [33] M. Eiermann, O. G. Ernst, and E. Ullmann. Computational aspects of the stochastic finite element method. *Computing and Visualization in Science*, 10(1):3–15, 2007.
- [34] H. C. Elman, O. G. Ernst, D. O’Leary, 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(9):1037–1055, 2005.

- [35] H. C. Elman and D. Furnival. Solving the stochastic steady-state diffusion problem using multigrid. *IMA Journal of Numerical Analysis*, 2007.
- [36] H. C. Elman, D. Furnival, and C. E. Powell. H (*div*) preconditioning for a mixed finite element formulation of the diffusion problem with random data. *Mathematics of Computation*, 79(270):733–760, 2010.
- [37] H. C. Elman, C. 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(1):19–33, 2011.
- [38] H. C. Elman, A. Ramage, and D. J. Silvester. IFISS: A computational laboratory for investigating incompressible flow problems. *SIAM Review*, 56(2):261–273, 2014.
- [39] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014.
- [40] O. G. Ernst, C. E. Powell, D. J. Silvester, and E. Ullmann. Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data. *SIAM Journal on Scientific Computing*, 31(2):1424–1447, 2009.
- [41] O. G. Ernst and E. Ullmann. Stochastic Galerkin matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1848–1872, 2010.
- [42] R. G. Ghanem. Scales of fluctuation and the propagation of uncertainty in random porous media. *Water Resources Research*, 34(9):2123–2136, 1998.
- [43] R. G. Ghanem. Ingredients for a general purpose stochastic finite elements implementation. *Computer Methods in Applied Mechanics and Engineering*, 168(1):19–34, 1999.
- [44] R. G. Ghanem. Stochastic finite elements with multiple random non-Gaussian properties. *Journal of Engineering Mechanics*, 125(1):26–40, 1999.
- [45] R. G. Ghanem and R. Kruger. Numerical solution of spectral stochastic finite element systems. *Computer Methods in Applied Mechanics and Engineering*, 129(3):289–303, 1996.
- [46] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: a Spectral Approach*. Dover Publications, 2003.
- [47] G. H. Golub and C. F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.
- [48] L. Grasedyck. Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. *Computing*, 72(3-4):247–265, 2004.

- [49] M. Grigoriu. Probabilistic models for stochastic elliptic partial differential equations. *Journal of Computational Physics*, 229(22):8406–8429, 2010.
- [50] J. Hampton and A. Doostan. Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. *Journal of Computational Physics*, 280:363–386, 2015.
- [51] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- [52] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.
- [53] H. Holden, B. Øksendal, J. Ubøe, and T. Zhang. *Stochastic Partial Differential Equations*. Springer, 1996.
- [54] S. Hosder, R. Walters, and R. Perez. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In *44th AIAA aerospace sciences meeting and exhibit*, 2006.
- [55] B.-N. Jiang and L. A. Povinelli. Least-squares finite element method for fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 81(1):13–37, 1990.
- [56] A. Keese. A review of recent developments in the numerical solution of stochastic partial differential equations (stochastic finite elements). Technical report, Institute of Scientific Computing, Department of Mathematics and Computer Science, Technische Universität Braunschweig, Brunswick, 2003.
- [57] C. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995.
- [58] O. M. Knio, H. N. Najm, and R. G. Ghanem. A stochastic projection method for fluid flow: I. basic formulation. *Journal of computational Physics*, 173(2):481–511, 2001.
- [59] D. Kressner, M. Steinlechner, and B. Vandereycken. Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure. *SIAM Journal on Scientific Computing*, 38(4):A2018–A2044, 2016.
- [60] D. Kressner and C. Tobler. Krylov subspace methods for linear systems with tensor product structure. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1688–1714, 2010.
- [61] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011.

- [62] O. P. Le Maître, B. Debusschere, H. N. Najm, R. G. Ghanem, and O. M. Knio. Natural convection in a closed cavity under stochastic non-Boussinesq conditions. *SIAM Journal on Scientific Computing*, 26(2):375–394, 2004.
- [63] O. P. Le Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification: with Applications to Computational Fluid Dynamics*. Springer, 2010.
- [64] O. P. Le Maître, O. M. Knio, B. J. Debusschere, H. N. Najm, and R. G. Ghanem. A multigrid solver for two-dimensional stochastic diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 192(41):4723–4744, 2003.
- [65] K. Lee and H. C. Elman. A preconditioned low-rank projection method with a rank-reduction scheme for stochastic partial differential equations. *arXiv preprint arXiv:1605.05297*, 2016.
- [66] C. V. Loan and N. Pitsianis. Approximation with Kronecker products. Technical report, Cornell University, 1992.
- [67] M. Loève. *Probability Theory, Vol. II*, volume 46. Springer, 1978.
- [68] D. Lucor, D. Xiu, and G. Karniadakis. Spectral representations of uncertainty in simulations: Algorithms and applications. *ICOSAHOM-01, Uppsala Sweden*, 2001.
- [69] H. G. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194(12):1295–1331, 2005.
- [70] H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra and its Applications*, 436(10):3819–3838, 2012.
- [71] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [72] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.
- [73] A. Mugler and H.-J. Starkloff. On the convergence of the stochastic Galerkin method for random elliptic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(5):1237–1263, 2013.
- [74] H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bulletin of the American Mathematical Society*, 84(6):957–1041, 1978.
- [75] A. Nouy. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 196(45):4521–4537, 2007.

- [76] A. Nouy and O. P. Le Maître. Generalized spectral decomposition for stochastic nonlinear problems. *Journal of Computational Physics*, 228(1):202–235, 2009.
- [77] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- [78] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.
- [79] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
- [80] M. F. Pellissetti and R. G. Ghanem. Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Advances in Engineering Software*, 31(8):607–616, 2000.
- [81] C. E. Powell and H. C. Elman. Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis*, 29:350–375, 2009.
- [82] C. E. Powell and D. J. Silvester. Preconditioning steady-state Navier–Stokes equations with random data. *SIAM Journal on Scientific Computing*, 34(5):A2482–A2506, 2012.
- [83] C. E. Powell, D. J. Silvester, and V. Simoncini. An efficient reduced basis solver for stochastic Galerkin matrix equations. *SIAM Journal on Scientific Computing*, 39(1):A141–A163, 2017.
- [84] C. E. Powell and E. Ullmann. Preconditioning stochastic Galerkin saddle point systems. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2813–2840, 2010.
- [85] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: an Introduction*, volume 92. Springer, 2015.
- [86] M. M. Rao and R. J. Swift. *Probability Theory with Applications*. Springer-Verlag, New York, 2nd edition, 2006.
- [87] P. Raviart and J. Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of Finite Element Methods*, pages 292–315. Springer, 1977.
- [88] M. Reagan, H. N. Najm, B. J. Debusschere, O. P. Le Maître, O. M. Knio, and R. G. Ghanem. Spectral stochastic uncertainty quantification in chemical systems. *Combustion Theory and Modelling*, 8(3):607–632, 2004.
- [89] E. Rosseel and S. Vandewalle. Iterative solvers for the stochastic finite element method. *SIAM Journal on Scientific Computing*, 32(1):372–397, 2010.

- [90] J. Ruge and K. Stüben. Algebraic multigrid. *Multigrid Methods*, 3:73–130, 1987.
- [91] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.
- [92] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [93] C. Schwab and C. J. Gittelsohn. Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. *Acta Numerica*, 20:291–467, 2011.
- [94] B. Seynaeve, E. Rosseel, B. Nicolai, and S. Vandewalle. Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *Journal of Computational Physics*, 224(1):132–149, 2007.
- [95] L. F. Shampine. Vectorized adaptive quadrature in MATLAB. *Journal of Computational and Applied Mathematics*, 211(2):131–140, 2008.
- [96] Y. Shin and D. Xiu. Nonadaptive quasi-optimal points selection for least squares linear regression. *SIAM Journal on Scientific Computing*, 38(1):A385–A411, 2016.
- [97] Y. Shin and D. Xiu. On a near optimal sampling strategy for least squares polynomial regression. *Journal of Computational Physics*, 326:931–946, 2016.
- [98] D. J. Silvester, H. C. Elman, and A. Ramage. Incompressible Flow and Iterative Solver Software (IFISS) version 3.4, August 2015. <http://www.manchester.ac.uk/ifiss/>.
- [99] B. Sousedík and H. C. Elman. Stochastic Galerkin methods for the steady-state Navier–Stokes equations. *Journal of Computational Physics*, 316:435–452, 2016.
- [100] B. Sousedík and R. G. Ghanem. Truncated hierarchical preconditioning for the stochastic Galerkin FEM. *International Journal for Uncertainty Quantification*, 4(4):333–348, 2014.
- [101] B. Sousedík, R. G. Ghanem, and E. T. Phipps. Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods. *Numerical Linear Algebra with Applications*, 21(1):136–151, 2014.
- [102] P. D. Spanos and R. G. Ghanem. Stochastic finite element expansion for random media. *Journal of Engineering Mechanics*, 115(5):1035–1053, 1989.
- [103] M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 37(1):B1–B29, 2015.

- [104] L. Tamellini, O. P. Le Maître, and A. Nouy. Model reduction based on proper generalized decomposition for the stochastic steady incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 36(3):A1089–A1117, 2014.
- [105] E. Ullmann. A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. *SIAM Journal on Scientific Computing*, 32(2):923–946, 2010.
- [106] 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(2):A659–A682, 2012.
- [107] 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(1):509–534, 2015.
- [108] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.
- [109] D. Xiu. Efficient collocational approach for parametric uncertainty analysis. *Communications in Computational Physics*, 2(2):293–309, 2007.
- [110] D. Xiu. *Numerical Methods for Stochastic Computations: a Spectral Method Approach*. Princeton University Press, 2010.
- [111] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.
- [112] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.
- [113] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.
- [114] D. Xiu, D. Lucor, C. Su, and G. E. Karniadakis. Stochastic modeling of flow-structure interactions using generalized polynomial chaos. *Journal of Fluids Engineering*, 124(1):51–59, 2002.
- [115] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*. Academic Press, 2001.