MONASH University

Advanced multi-level and multi-index Monte Carlo methods in uncertainty quantification

Stanislav Y. Polishchuk
Master of Science

A thesis submitted for the degree of Doctor of Philosophy at
Monash University in 2022
School of Mathematics

Copyright notice

Abstract

As a result of measurement noise and uncertainties in model-driven factors such as initial conditions, boundary conditions, domain geometry, model parameters, and other model inputs it is necessary to quantify uncertainties in solutions of PDE-based models for science and engineering applications. The Monte Carlo sampling method is one of the most popular approaches in handling uncertainties of high-dimensional and nonlinear problems, but it comes with a high computational cost due to repeated model evaluations.

In this work, we propose a suite of accelerated Monte Carlo methods for quantifying uncertainties in elliptic PDE problems. We explore the use of methods based on finite element discretization such as the *hp*-finite element method and the streamline-upwind/Petrov-Galerkin method in the context of multi-level Monte Carlo (MLMC) and multi-index Monte Carlo (MIMC) methods. In this setting, we employ finite element-based methods for the discretization of elliptic and convection-diffusion problems with random conductivity modelled as a convolution of a Gaussian process.

We propose and investigate several methods including geometric MLMC and MIMC with different polynomial order of the basis functions; *hp*-MLMC in which we refine mesh and increase the order of basis functions simultaneously with level; *p*-MLMC in which we only increase the order of basis functions without any further mesh refinement; and a homotopy-based MLMC in which we use a homotopy parameter to construct a hierarchy of discretized eigenvalue problems. In addition to these methods, we also develop and investigate new choices of the index sets in MIMC including various combinations of refining the grid spacing in x and y directions as well as polynomial order in x and y. To illustrate the efficiency of our methods, we consider a variety of quantities of interest with different modes of convergence and regularity properties for the mean and variance, such as the average solution in a given volume and the smallest eigenvalue of a non-self-adjoint operator.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Stanislav Y. Polishchuk
27 May 2022

Acknowledgements

I would like to thank my supervisors Professor Tiangang Cui and Professor Hans De Sterck for their thoughtful support and guidance. I thank the milestone panel members Professor Paul Cally, Dr. Simon Clarke, and Dr. Kengo Deguchi for their helpful feedback and comments. Special thanks are due Dr. Alexander D. Gilbert and Professor Robert Scheichl for their discussions. And finally, I would like to acknowledge my thesis examiners Professor Zhiwen Zhang and Dr. James A. Nichols for their valued comments and suggestions.

I would also like to thank Monash University and the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for providing financial support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Uncertainties arise in a variety of physical and scientific applications and their numerical simulations. Measurement noise, limitations of mathematical models, existence of hidden variables, randomness of input parameters, and other factors contribute to uncertainties in modelling and prediction of many phenomena. Fundamentally, two main classes of uncertainties are considered. Aleatory uncertainty concerns randomness that is inherent and epistemic uncertainty concerns the lack of knowledge in computational models [90]. Although uncertainties appear in many applications, the field of uncertainty quantification (UQ) emerged only recently.

UQ can be split into several areas but the two major ones are forward propagation of uncertainty and inverse uncertainty quantification. Forward uncertainty propagation considers processes with random input, such as initial and boundary conditions, domain geometry, material properties, etc. in the computational model and analyzes their effect on chosen output parameters or a quantity of interest (QoI). Typical outcomes of the uncertainty propagation phase may include summary statistics, such as output mean and variance. The models are usually described by stochastic partial differential equations (PDEs). In practice, random inputs of stochastic PDEs are approximated as a finite number of random variables which is usually large. As such, Monte Carlo sampling is one of the most flexible approaches for quantifying uncertainties of stochastic PDEs [13, 44, 69, 72]. Typically, Monte Carlo is used in the high-dimensional and nonlinear setting. The main drawback of Monte Carlo simulation is its slow convergence, as it needs $O(N^{-1/2})$ samples, and each drawn sample usually requires a solution of the PDE which can be expensive. Variance reduction techniques are used to reduce the computational cost of Monte Carlo, such as control variates, antithetic sampling, and importance sampling. A recent popular approach for solving stochastic PDEs is the multi-level Monte Carlo technique which is based on control variates. Alternative approaches may include modelling stochastic PDEs based on a polynomial chaos expansion, such as stochastic Galerkin, stochastic collocation, etc. [34, 63, 98, 99, 100], but these methods suffer from the so-called *curse of dimensionality* as well as from strong nonlinearity, with the number of terms in the stochastic expansion growing exponentially with dimension [41].

Inverse uncertainty quantification concerns estimating the input parameters given incomplete and noisy observations of outputs. Since most inverse problems are ill-posed, Bayesian inference is naturally suited to regularize the ill-posedness [93]. The framework allows to incorporate *a priori* information regarding the uncertainties into a prior distribution to obtain the posterior probability density function based on the likelihood [6, 25, 44]. However, its computational cost poses a challenge in constructing an efficient Markov chain Monte Carlo method for sampling from unknown posterior distributions [17, 29, 58, 93].

This work examines only forward uncertainty propagation with a focus on Monte Carlo sampling methods. The multi-level Monte Carlo (MLMC) method and its extension, the multi-index Monte Carlo (MIMC) method provide variance reduction techniques that utilize a sequence of approximate models of the QoI with increasing accuracy and computational cost. MLMC-based methods exploit the linearity of expectation to minimize the variance of the estimated output QoI. As a result, a smaller number of solutions of the finest and most expensive PDE model is required to achieve a certain accuracy. The main idea was introduced by Heinrich in 2001 [46], then generalized by Giles in 2008 [39]. There is vast literature available on the application of MLMC to stochastic PDEs [8, 10, 22, 37, 38, 66, 76, 87, 96]. A general overview of MLMC is presented in [40] by Giles with several applications for stochastic DEs, such as the simple Euler-Maruyama discretization, Lévy processes, elliptic PDEs, etc. One generalization of MLMC is MIMC which was introduced by Haji-Ali, Nobile and Tempone in 2014 [43]. Haji-Ali et al. constructed a set of approximate QoI based on the spatial discretization in each direction of a multi-dimensional PDE and provided an example for a 3D elliptic problem with random conductivity showing its superiority compared to MLMC and standard Monte Carlo.

The goal of the dissertation is to design and apply various efficient and practical MLMC and MIMC techniques in the context of 2D elliptic PDEs with random input. We consider random self-adjoint and non-self-adjoint elliptic problems. The conductivity in both cases is modelled as a random field which serves as an accurate representation of a realistic heterogeneous field. In the case of a self-adjoint elliptic problem, we employ the Galerkin finite element method to discretize the PDE with the use of high-order polynomial basis functions. This allows us to develop several MLMC and MIMC methods based on various combinations of h-, p-, and hp- refinement strategies. This thesis advances the MIMC approach and constructs sets of discretized PDE models based on the notion of incompleteness of polynomials in 2D of a different order in x and y directions of finite elements. This way, the variance reduction rate becomes even higher compared to MIMC with the use of only spatial discretization. For the case of non-self-adjoint elliptic problems, we consider the smallest eigenvalue as QoI. Since the non-self-adjoint eigenvalue problem demands special care in choosing the discretization method, we extend MLMC to continuation-based homotopy methods [19, 71]. We also consider an alternative approach based on the Petrov-Galerkin formulation [15, 16]. That way, we may include

cheaper models into the multi-level sequence, so the total computational time will be less than when using a plain Galerkin formulation. Since the multi-level differences require solutions with the same realization of the random field for adjacent levels, we utilize a two-grid method for the presented iterative solvers: the biconjugate gradient stabilized method (BiCGStab), and the Rayleigh quotient and Arnoldi methods. Finally, we analyze the convergence of the mean and the variance and the computational complexity of the presented methods and compare with classical Monte Carlo simulation.

1.1 Outline of thesis

This dissertation is organized as follows. Chapter 2 introduces the standard, multi-level, and multi-index Monte Carlo methods along with necessary theory. Chapter 3 focuses on MLMC and MIMC for the 2D elliptic boundary value problem with randomness in conductivity. It proposes several MLMC and MIMC strategies based on various properties of the finite element discretization. It also introduces some novel techniques utilizing incomplete multivariate polynomials. Then, the methods developed are compared for two quantities of interest. The convection-diffusion eigenvalue problem is the subject of Chapter 4. An additional MLMC method is developed based on the homotopy continuation method to find the smallest eigenvalue of the convection-diffusion operator. Two eigenvalue solvers are coupled with MLMC: the Rayleigh quotient and implicitly restarted Arnoldi iterations. Chapter 5 considers an approach based on the streamline-upwind/Petrov-Galerkin method for finding the smallest eigenvalue of the convection-diffusion operator in the context of high velocity. Finally, Chapter 6 summarizes the work and suggests future developments.

Chapter 2

Multi-level and multi-index Monte Carlo methods

In this chapter we discuss various Monte Carlo variance reduction techniques with the primary focus on the multi-level and multi-index Monte Carlo methods. We start with a general description of the classical Monte Carlo method. Next we introduce the two-level Monte Carlo method and its generalizations, the multi-level and multi-index Monte Carlo methods. This chapter serves as a foundation to the next chapters in which we develop new applications of these methods to a wide range of problems.

2.1 Classic Monte Carlo method

Suppose we want to estimate the value of an integral

$$I(g(\omega)) = \int_D g(\omega)\,\mathrm{d}\omega, \tag{2.1}$$

where D is a d-dimensional cube $[0;1]^d$ and $\omega \in D$. We could use Gaussian quadrature to calculate the integral but if the domain D is in a high-dimensional space or if the function $g(\omega)$ exhibits a highly nonlinear behaviour then the estimation would be computationally expensive.

In such case, the use of Monte Carlo techniques would be a more suitable approach. For that, we treat the independent variable ω and the function $g(\omega)$ as a random variable and a random function, respectively. Then we draw N independent and identically distributed (i.i.d.) random samples $\omega_1, \ldots, \omega_N$ from the uniform distribution. And thus, the classic Monte Carlo estimator for the integral (2.1) is defined as

$$I \approx I_N = \frac{1}{N}\sum_{n=1}^{N} g(\omega_n), \tag{2.2}$$

which is equivalent to finding the expectation $\mathbb{E}[g(\omega)]$ of the function $g(\omega)$, i.e. $I(g(\omega)) =$

$\mathbb{E}[g(\omega)]$. Because of the law of large numbers the value I_N converges to the integral I as $N \to \infty$ with probability 1.

If the variance $\mathbb{V}[g(\omega)] = \mathbb{E}[(g(\omega) - \mathbb{E}[g(\omega)])^2]$ of the function $g(\omega)$ is finite then the convergence rate of the estimator is given by the central limit theorem:

$$\varepsilon_N(g) = (I_N - I) \to \mathcal{N}\left(0, \frac{\mathbb{V}_N[g]}{N}\right) \quad \text{in distribution,} \tag{2.3}$$

where $\mathbb{V}_N[g(\omega)] = \mathbb{E}[(I_N - \mathbb{E}[I_N])^2]$. Thus, the root mean square error (RMSE) $\sqrt{\mathbb{E}[(\varepsilon_N(g))^2]}$ of the Monte Carlo estimator I_N is $O(N^{-1/2})$.

Therefore, it may require a very large number of samples to make an estimation of the integral $I(g)$ with a given RMSE. In situations when one needs to account for the cost of each sample, e.g., when each sample requires solving a linear system of equations, this would lead to a large computational cost. Various techniques exist to reduce the cost of the Monte Carlo estimator, mainly focusing on the minimization of the variance, such as importance sampling in which one uses specifically constructed sequences of samples that are no longer i.i.d random samples. Another approach is to use methods based on control variates for the variance reduction in which a known function is used to estimate the function of interest by utilizing the difference between them.

2.2 Two-level Monte Carlo method

Suppose the function g can be obtained via a discretization of the underlying PDEs of a model. To estimate the expected value $\mathbb{E}[g]$ with a given RMSE, one can use a solution g_0 computed with a low accuracy and with low cost as a control variate. For example, in a case where the domain of the PDEs is discretized (Figure 2.1), the function g_0 can represent a solution obtained on a coarse mesh while the function g represents a solution calculated on a fine mesh with higher computational cost and accuracy.

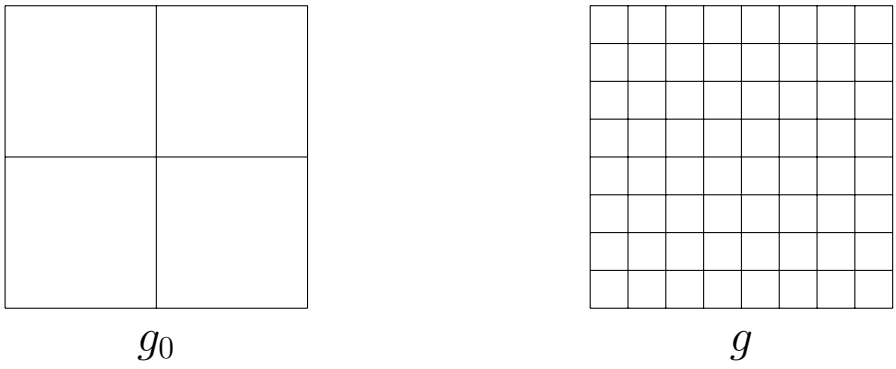


Figure 2.1 – Examples of coarse and fine meshes.

Since

$$\mathbb{E}[g] - \mathbb{E}[g_0] = \mathbb{E}[g - g_0], \tag{2.4}$$

we have

$$\mathbb{E}[g] = \mathbb{E}[g_0] + \mathbb{E}[g - g_0], \tag{2.5}$$

Then one can use the following two-level Monte Carlo estimator

$$\mathbb{E}[g] \approx \frac{1}{N_0} \sum_{n_0=1}^{N_0} g_0(\omega_{n_0}) + \frac{1}{N_1} \sum_{n_1=1}^{N_1} \left(g(\omega_{n_1}) - g_0(\omega_{n_1}) \right), \tag{2.6}$$

We define V_0, C_0 as the variance and cost of a single sample of g_0 and V_1, C_1 as the variance and cost of a single sample of $g - g_0$. Then to determine the numbers of samples N_0 and N_1 at each level one can perform a minimization of the total cost $N_0 C_0 + N_1 C_1$ for a total variance $N_0^{-1} V_0 + N_1^{-1} V_1$ fixed to a value of ε by using a Lagrangian function

$$\mathcal{L}(N_0, N_1, \lambda) = N_0 C_0 + N_1 C_1 + \lambda (N_0^{-1} V_0 + N_1^{-1} V_1 - \varepsilon). \tag{2.7}$$

By solving $\nabla_{N_0, N_1, \lambda} \mathcal{L}(N_0, N_1, \lambda) = 0$, we obtain the optimal relation between the numbers of samples at the coarse and fine levels

$$\frac{N_1}{N_0} = \frac{\sqrt{V_1 / C_1}}{\sqrt{V_0 / C_0}}. \tag{2.8}$$

In what follows we will see how this optimal choice of sample numbers can substantially reduce the total cost compared to applying Monte Carlo on the fine level only.

2.3 Multi-level Monte Carlo method

The multi-level Monte Carlo method is based on the simple idea of using a convergent sequence of approximate solutions to the quantity of interest. Suppose we have a sequence of values $Q_0, Q_1, \ldots, Q_{L-1}$ approximating with increasing accuracy and cost our quantity of interest Q, e.g. the sequence $\{Q_l\}$ may represent PDE solutions obtained on a sequence of discretized meshes using mesh size as a discretization parameter (Figure 2.2). Then we may construct the relation

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}], \tag{2.9}$$

and we can use the following unbiased estimator Y for $\mathbb{E}[Q_L]$

$$Y = \sum_{\ell=0}^{L} Y_\ell, \quad Y_\ell = \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} (Q_\ell(\omega_n) - Q_{\ell-1}(\omega_n)), \tag{2.10}$$

with $Q_{-1} \equiv 0$. The inclusion of the level ℓ in the superscript (ℓ, n) indicates that independent samples are used at each level and N_ℓ is the number of samples at level ℓ. Then

we have

$$\mathbb{E}[Y] = \mathbb{E}[Q_L], \;\; \mathbb{V}[Y] = \sum_{\ell=0}^{L} N_\ell^{-1} V_\ell, \;\; V_\ell \equiv \mathbb{V}[Q_\ell - Q_{\ell-1}], \tag{2.11}$$

with mean square error (MSE)

$$\mathrm{MSE} \equiv \mathbb{E}[(Y - \mathbb{E}[Q])^2] = \mathbb{V}[Y] + (\mathbb{E}[Y] - \mathbb{E}[Q])^2. \tag{2.12}$$

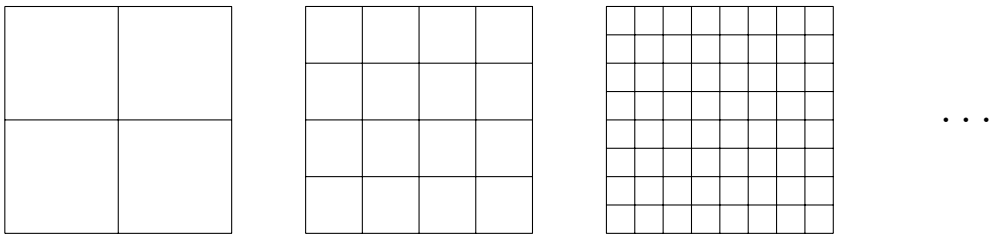


Figure 2.2 – A sequence of mesh refinements.

To ensure that the MSE is less than ε^2, it is sufficient to ensure that the variance $\mathbb{V}[Y]$ and $(\mathbb{E}[Q_L - Q])^2$ are both less than $\frac{1}{2}\varepsilon^2$.

In [40], the following theorem is proposed for the computational cost of the MLMC method.

Theorem 1. *[40] Let Q denote a random variable, and let Q_ℓ denote the corresponding level ℓ numerical approximation with a meshwidth $h = h_0 2^{-\ell}$ to Q. If there exist positive constants $\alpha, \beta, \gamma, c_1, c_2, c_3$ such that $\alpha \geq \frac{1}{2}\min(\beta, \gamma)$ and*

I (bias of estimator). $|\mathbb{E}[Q_\ell - Q]| \leq c_1 2^{-\alpha\ell}$

II (order of convergence of variance). $\mathbb{V}[Q_\ell - Q_{\ell-1}] \leq c_2 2^{-\beta\ell}$

III (cost of a sample). $C_\ell \leq c_3 2^{\gamma\ell}$, *where C_ℓ is the cost of one sample of $Q_\ell - Q_{\ell-1}$,*

then there exists a positive constant c_4 such that for any $\varepsilon < e^{-1}$ there are values L and N_ℓ for which the multilevel estimator

$$Y = \sum_{\ell=0}^{L} Y_\ell, \tag{2.13}$$

has a mean-square-error with bound

$$MSE \equiv \mathbb{E}[(Y - \mathbb{E}[Q])^2] < \varepsilon^2 \tag{2.14}$$

with a computational complexity C with bound

$$\mathbb{E}[C] \leq \begin{cases} c_4 \varepsilon^{-2}, & \beta > \gamma; \\ c_4 \varepsilon^{-2} (\log \varepsilon)^2, & \beta = \gamma; \\ c_4 \varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma, \end{cases} \tag{2.15}$$

where the constant c_4 is independent of α, β, and γ rates.

The theorem can be proved by computing the optimal N_ℓ to minimize the total cost for a fixed variance: for the Lagrangian

$$\sum_{\ell=0}^{L}(N_\ell C_\ell + \lambda^2 N_\ell^{-1} V_\ell) - \lambda^2 \frac{\varepsilon^2}{2} \tag{2.16}$$

with Lagrange multipliers λ^2, this gives $N_\ell = \lambda\sqrt{V_\ell/C_\ell}$ where $\lambda = 2\varepsilon^{-2}\sum_{\ell=0}^{L}\sqrt{V_\ell C_\ell}$, so the equations for the optimal N_ℓ become

$$N_\ell = 2\varepsilon^{-2}\sqrt{V_\ell/C_\ell}\sum_{i=0}^{L}\sqrt{V_i C_i}, \tag{2.17}$$

where V_ℓ and C_ℓ are the estimated variance and cost of a sample on level ℓ.

If $\mathbb{E}[Q_\ell - Q_{\ell-1}] \propto 2^{-\alpha\ell}$ then the remaining error is

$$\mathbb{E}[Q - Q_L] = \sum_{\ell=L+1}^{\infty}\mathbb{E}[Q_\ell - Q_{\ell-1}] = \mathbb{E}[Q_L - Q_{L-1}]/(2^\alpha - 1). \tag{2.18}$$

Then the convergence test is $\big|\mathbb{E}[Q_L - Q_{L-1}]\big|/(2^\alpha - 1) < \varepsilon/\sqrt{2}$ where ε is a targeted RMSE. This will ensure that $\big|\mathbb{E}[Q - Q_L]\big| < \varepsilon/\sqrt{2}$ reaches an MSE less than ε^2. Pseudocode for the MLMC algorithm is given in Algorithm 1.

Algorithm 1 Multilevel Monte Carlo algorithm [40]

1: start with $L = 2$ and initial target of N_0 samples on levels $\ell = 0, 1...$
2: **while** extra samples need to be evaluated **do**
3: evaluate extra samples on each level
4: compute/update estimates for V_ℓ, $\ell = 0, ..., L$
5: define optimal N_ℓ, $\ell = 0, ..., L$
6: test for weak convergence
7: if not converged, set $L := L + 1$, and initialize target N_L
8: **end while**

In case when $\beta > \gamma$, it means that variance reduces faster than the cost increases with level, and most of the work is spent on the coarsest level. As a result, only $O(\varepsilon^{-2})$ samples are required to estimate QoI with desired accuracy ε and the total cost will be $C \approx \varepsilon^{-2}V_0C_0$. As an example, a 2D elliptic problem could be considered in which the problem is discretized with the standard finite element method using linear elements. Then if we consider a linear functional as QoI, the uniform second order accuracy yields $\alpha = 2$ and variance reduction rate is $\beta = 4$. The cost increase rate is usually $\gamma < 3$ and therefore the total complexity is $O(\varepsilon^{-2})$.

In the opposite case, when $\beta < \gamma$ the total computational work is almost of the same complexity as the classic Monte Carlo method, meaning that one spends most of the work

on the finest level. Then the total cost is $C \approx \varepsilon^{-2} V_L C_L$. But if $\beta = 2\alpha$ as $\mathbb{V}[Q_\ell - Q_{\ell-1}]$ is typically of the same order as $\mathbb{E}[Q_\ell - Q_{\ell-1}]^2$, then $C_L = O(\varepsilon^{-2-(\gamma-\beta)/\alpha}) = O(\varepsilon^{-\gamma/\alpha})$ (2.15) which potentially could be better than the standard Monte Carlo method.

In the case when the variance reduction rate and cost increase rate are equal $\beta = \gamma$, the computational cost and contributions to total variance are evenly distributed across all levels. The total cost is then $\varepsilon^{-2} L^2 V_0 C_0 = \varepsilon^{-2} L^2 V_L C_L$.

2.4 Multi-index Monte Carlo method

The multi-index Monte Carlo (MIMC) method is a generalization of the multi-level Monte Carlo method. Instead of a sequence of levels with approximate models, we now have an ordered set of models with index-level $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_D)$. Suppose we have a finite element approximation for a 2D problem. Then the refinement can be done in the usual way by decreasing the element size h. Having a QoI Q, we may construct the following sequence of levels $(Q^{h_0}, Q^{h_1}, \ldots, Q^{h_\ell}, \ldots, Q^{h_L})$ where h_l is mesh size on level ℓ. In MIMC setting, we can use finite element models by refining in a spatial direction, either in x or y. That way, we have a 2D index set of approximate models

$$\begin{pmatrix} Q^{h_x^0 h_y^0} & Q^{h_x^1 h_y^0} & \ldots & Q^{h_x^L h_y^0} \\ Q^{h_x^0 h_y^1} & Q^{h_x^1 h_y^1} & \ldots & Q^{h_x^L h_y^1} \\ \vdots & \vdots & \ddots & \vdots \\ Q^{h_x^0 h_y^L} & Q^{h_x^1 h_y^L} & \ldots & Q^{h_x^L h_y^L} \end{pmatrix}. \tag{2.19}$$

MIMC uses high-order mixed differences to reduce the variance of the resulting estimator and its corresponding work [43]. In MLMC, the difference is defined as

$$\Delta Q_\ell \equiv Q_\ell - Q_{\ell-1},$$

with $Q_{-1} \equiv 0$. Then the telescopic sum becomes

$$\mathbb{E}[Q] = \sum_{\ell \geq 0} \mathbb{E}[\Delta Q_\ell].$$

In MIMC, we define a difference in one particular dimension

$$\Delta_d Q_{\boldsymbol{\ell}} \equiv Q_{\boldsymbol{\ell}} - Q_{\boldsymbol{\ell} - \boldsymbol{e}_d},$$

where $\boldsymbol{e}_d$ is the unit vector in direction d. Then generalizing to D dimensions, we can define the cross-difference

$$\Delta Q_{\boldsymbol{\ell}} \equiv \left(\prod_{d=1}^{D} \Delta_d \right) Q_{\boldsymbol{\ell}},$$

the telescopic sum in MIMC is

$$\mathbb{E}[Q] = \sum_{\boldsymbol{\ell} \in \mathfrak{L}} \mathbb{E}[\Delta Q_{\boldsymbol{\ell}}],$$

and we can use the following unbiased estimator Y for $\mathbb{E}[Q_L]$

$$Y = \sum_{\boldsymbol{\ell} \in \mathfrak{L}} Y_{\boldsymbol{\ell}}, \quad Y_{\boldsymbol{\ell}} = \frac{1}{N_{\boldsymbol{\ell}}} \sum_{n=1}^{N_{\boldsymbol{\ell}}} \Delta Q_{\boldsymbol{\ell}}^{(\boldsymbol{\ell},n)}. \tag{2.20}$$

where $\mathfrak{L}$ is some index set to be specified later.

For example, consider $D = 2$ (Figure 2.3). Letting $\boldsymbol{\ell} = (\ell_1, \ell_2)$ we have

$$\Delta Q_{\boldsymbol{\ell}} = \left(\prod_{d=1}^{D} \Delta_d \right) Q_{\boldsymbol{\ell}} = \Delta_1 \Delta_2 Q_{\boldsymbol{\ell}} = \Delta_1 (Q_{\boldsymbol{\ell}} - Q_{\boldsymbol{\ell} - e_2}) = \Delta_1 Q_{\boldsymbol{\ell}} - \Delta_1 Q_{\boldsymbol{\ell} - e_2} =$$
$$(Q_{\boldsymbol{\ell}} - Q_{\boldsymbol{\ell} - e_1}) - (Q_{\boldsymbol{\ell} - e_2} - Q_{\boldsymbol{\ell} - e_1 - e_2}) = Q_{\ell_1, \ell_2} - Q_{\ell_1 - 1, \ell_2} - Q_{\ell_1, \ell_2 - 1} + Q_{\ell_1 - 1, \ell_2 - 1}$$

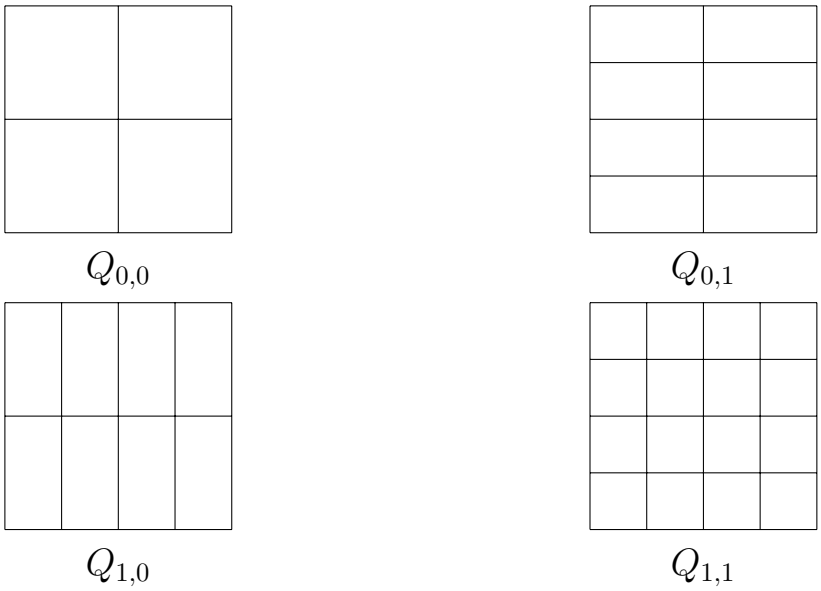


Figure 2.3 – A set of meshes using h_x and h_y as discretization parameters.

Haji-Ali et al. [43] formulated the following theorem for the computational cost of the MIMC method.

Theorem 2. *Let denote Q a random variable, and let $Q_{\boldsymbol{\ell}}$ denote the corresponding level $\boldsymbol{\ell} = (\ell_1, \ell_2, \ldots, \ell_D)$ numerical approximation with a meshwidth $h_d = h_{0,d} 2^{-\ell_d}$ $d = \overline{1, D}$. If there exist positive D-dimensional vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$, with $\alpha_d \geq \frac{1}{2}\beta_d$ for $d = 1, \ldots, D$, and also positive constants c_1, c_2, c_3 such that*

I (weak convergence of estimator). $|\mathbb{E}[Q_{\boldsymbol{\ell}} - Q]| \to 0$ *as* $\min_d l_d \to \infty$,

II (bias of estimator). $\mathbb{E}_{\boldsymbol{\ell}} = |\mathbb{E}[\Delta Q_{\boldsymbol{\ell}}]| \leq c_1 2^{-\boldsymbol{\alpha}\boldsymbol{\ell}}$,

III (order of convergence of variance). $V_{\boldsymbol{\ell}} = \mathbb{V}[\Delta Q_{\boldsymbol{\ell}}] \leq c_2 2^{-\boldsymbol{\beta}\boldsymbol{\ell}}$,

IV (cost of a sample). $C_{\boldsymbol{\ell}} = C(\Delta Q_{\boldsymbol{\ell}}) \leq c_3 2^{\boldsymbol{\gamma}\boldsymbol{\ell}}$, *where* $C_{\boldsymbol{\ell}}$ *is the cost of one sample of* $\Delta Q_{\boldsymbol{\ell}}$,

then there exists a positive constant c_4 *such that for any* $\varepsilon < e^{-1}$ *there is a set of levels* $\mathfrak{L}$, *and integers* $N_{\boldsymbol{\ell}}$ *for which the multi-index estimator*

$$Y = \sum_{\boldsymbol{\ell} \in \mathfrak{L}} Y_{\boldsymbol{\ell}},$$

has the mean-square error with bound

$$MSE \equiv \mathbb{E}[(Y - \mathbb{E}[Q])^2] < \varepsilon^2,$$

with a computational complexity C *with bound*

$$\mathbb{E}[C] \leq \begin{cases} c_4 \varepsilon^{-2}, & \eta < 0; \\ c_4 \varepsilon^{-2} |\log \varepsilon|^{e_1}, & \eta = 0; \\ c_4 \varepsilon^{-2-\eta} |\log \varepsilon|^{e_2}, & \eta > 0, \end{cases} \tag{2.21}$$

where

$$\eta = \max_d \frac{\gamma_d - \beta_d}{\alpha_d},$$

and the exponents e_1 *and* e_2 *are detailed in [43].*

The theorem can be proved by computing the optimal index-set

$$\mathfrak{L}(L) = \{\boldsymbol{\ell} \in \mathbb{N}^d, L \in \mathbb{R} : \boldsymbol{\ell} \cdot \boldsymbol{\delta} = \sum_{d=1}^{D} \ell_d \delta_d \leq L\},$$

where $\boldsymbol{\delta}$ is a general vector of weights satisfying the following property

$$\sum_{d \in D} \delta_d = 1 \quad \text{and} \quad 0 < \delta_d \leq 1,$$

with its components defined as

$$\delta_d = \frac{\log 2(\alpha_d + \frac{\gamma_d - \beta_d}{2})}{C_{\boldsymbol{\delta}}},$$

and

$$C_{\boldsymbol{\delta}} = \sum_{d=1}^{D} \log(2)(\alpha_j + \frac{\gamma_j - \beta_j}{2}).$$

The quasi-optimal number of samples is given by

$$N_{\boldsymbol{\ell}} = \varepsilon^{-2} \sqrt{\frac{V_{\boldsymbol{\ell}}}{C_{\boldsymbol{\ell}}}} \sum_{\tau \in \mathfrak{L}} \sqrt{V_\tau C_\tau},$$

for all $\boldsymbol{\ell} \in \mathfrak{L}$.

If it is possible to construct an optimal index set $\mathcal{L}$ with the property $\beta_d > \gamma_d$ in each direction, then the overall computational complexity will be the optimal $O(\varepsilon^{-2})$ because $\eta < 0$ in Eq. (2.21). The multi-index Monte Carlo method is applied in cases where the multi-level Monte Carlo may fail to perform compared to the standard Monte Carlo. Such cases usually include high-dimensional PDEs, since the variance reduction rate β is independent of the dimension of PDEs but the cost increase rate γ typically increases at least linearly with dimension. On the other hand, the multi-index Monte Carlo method removes the dependence on the dimension D for the cost γ. An example is given in [40] for the case of a D-dimensional elliptic PDE.

The goal of this thesis is to develop and test new MIMC approaches that improve the efficiency of uncertainty quantification methods for two-dimensional PDE problems. In the next chapter we will investigate new choices for the dimensions d along which the MIMC approach is applied, including various combinations of refining the grid spacing in x and y and the polynomial order in x and y.

Chapter 3

Fast Monte Carlo methods for elliptic PDEs

In this chapter, we review some existing multi-level and multi-index Monte Carlo methods in the context of the *hp*-finite element method (*hp*-FEM), and introduce new multi-index strategies that can exploit the hierarchy created by the *hp*-FEM discretization. We consider an elliptic problem with random coefficients modelled as a random field constructed by convolution of Gaussian random variables. Then we describe the *hp*-finite element method as the discretization method for the model problem used in the formulation of the *hp*-multi-index Monte Carlo method. We conclude the chapter with various numerical experiments for *hp*-multi-level Monte Carlo methods, including geometric multi-level, standard multi-index, and *hp*-multi-index Monte Carlo methods.

3.1 Model problem

Given a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, we consider a linear elliptic PDE for the function $u(x;\omega) : D \times \Omega \to \mathbb{R}$ with random coefficients as a model problem

$$-\nabla \cdot \kappa(x;\omega)\nabla u(x;\omega) = f(x;\omega), \quad x \in D, \omega \in \Omega, \tag{3.1}$$

in a domain $D \subset \mathbb{R}^d$ for $d = 1, 2$, or 3 with Lipschitz boundary $\Gamma = \Gamma_0 \cup \Gamma_1$, $\Gamma_0 \cap \Gamma_1 = \emptyset$. The conductivity $\kappa(x;\omega) : D \times \Omega \to \mathbb{R}$ is a Gaussian process satisfying $\kappa(\cdot,\omega) \in L^\infty(D)$ for almost all $\omega \in \Omega$, where x is a spatial coordinate, and ω is a random variable. We impose the Dirichlet boundary conditions on the boundary Γ_0

$$u|_{\Gamma_0} = u_g, \tag{3.2}$$

and the Neumann boundary conditions on the boundary Γ_1

$$n \cdot \kappa \nabla u|_{\Gamma_1} = 0, \tag{3.3}$$

where n is the outer unit normal vector.

The problem (3.1) may describe a stationary heat distribution u [55, 67, 102]. In that case κ is the heat conductivity, f is the heat sources, the Dirichlet boundary conditions (3.2) enforce the temperature on the boundary Γ_0, and the Neumann boundary conditions (3.3) impose vanishing heat flow.

3.2 Log-normal random field

To ensure the non-negativity of our random field, we model it as a log-normal random field through a Gaussian field $G(x;\omega) : \Omega \times D \to \mathbb{R}$, so that $\kappa(x;\omega) = \exp[G(x;\omega)]$. We also consider only mean zero homogeneous Gaussian fields with Lipschitz continuous covariance kernel

$$K(x_1, x_2) = \mathbb{E}\left[(G(x_1;\omega) - \mathbb{E}[G(x_1;\omega)])(G(x_2;\omega) - \mathbb{E}[G(x_2;\omega)])\right] = k(\|x_1 - x_2\|), \quad (3.4)$$

where $k(\cdot) \in C^{0,1}(\mathbb{R}^+)$ is a covariance function with some norm $\|\cdot\|$ in $\mathbb{R}^d$.

Although Monte Carlo methods do not require an approximation of the random field $G(x;\omega)$, for the simplicity we represent it as a function of a finite number of random variables. The usual way to approximate the random function is through the use of the truncated Karhunen-Loève (KL) expansion [1] in which the random process is represented as a series of bi-orthogonal functions. Another approach to approximate the random field is to build a convolution process using very simple kernel or point functions [49].

3.2.1 Truncated Karhunen-Loève expansion

The KL expansion is similar to Fourier analysis in terms of a representation of a function, the key difference is that in the Fourier series approximation the coefficients are deterministic and the expansion basis consists of trigonometric functions [1, 70]. In the KL expansion, in contrast, the coefficients are random and the orthogonal functions are derived from the covariance function of the random process. In its simplest form, the Karhunen-Loéve expansion is

$$G(x;\omega) = \sum_{i=1}^{\infty} \sqrt{\theta_i}\phi_i(x)\xi_i(\omega), \quad (3.5)$$

where $\{\xi_i\}$ is a set of independent, standard Gaussian random variables, θ_i are the eigenvalues and also non-negative, and ϕ_i are the corresponding normalized eigenfunctions of the covariance operator with a kernel function $K(x_1, x_2)$. In practical situation, the KL expansion is usually truncated after m terms

$$G_m(x;\omega) = \sum_{i=1}^{m} \sqrt{\theta_i}\phi_i(x)\xi_i(\omega), \quad (3.6)$$

which also gives a good approximation of the random field G for sufficiently large m. This way then our log-normal random filed yields a finite-dimensional approximation:

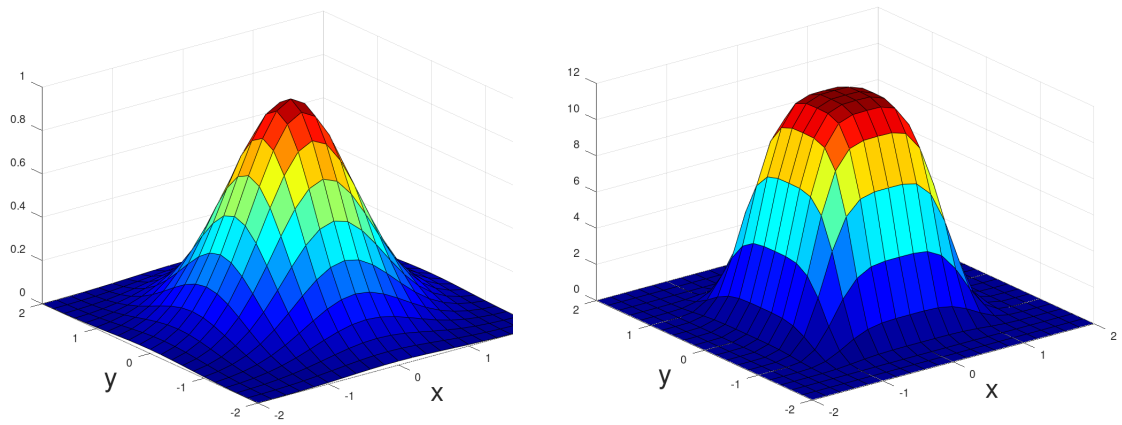
$$\kappa_m(x;\omega) = \exp\left[\sum_{i=1}^{m} \sqrt{\theta_i}\phi_i(x)\xi_i(\omega)\right]. \tag{3.7}$$

3.2.2 Convolution process

We can also use an alternative approach based on the convolution of a Gaussian process [49]. By modelling the random process this way, we may reduce the computational cost of constructing the field and also we gain a flexibility in setting more complicated models. To construct a Gaussian random field $G(x;\omega)$ with zero mean, we convolute i.i.d. Gaussian random variables over the domain D using a smoothing kernel $k(x)$. The Gaussian field in this case is

$$G(x;\omega) = \sum_i \omega_i k(x - c_i), \tag{3.8}$$

where $k(x - c_i)$ is a kernel centered at points c_i and random variables $\omega_i \sim \mathcal{N}(0,1)$ i.i.d. Figure 3.1 shows some examples of kernels that can be used as a smoothing kernel $k(x)$ yielding different properties for the field $\kappa(x;\omega)$ while Figure 3.2 shows a log-normal random field convoluted from 25 exponential kernels.



(a) $k(x) \propto \exp\{-\frac{1}{2}\|x\|^2\}$.

(b) $k(x) \propto (1 - \frac{\|x\|^3}{r^3})^3 I[x \leq r]$.

Figure 3.1 – Examples of kernels used in defining convolution processes.

3.3 Finite element method

We use the finite element method to obtain the discretization of the elliptic operator (3.1) as it allows us to work with arbitrary domain geometry and simple use of high-order

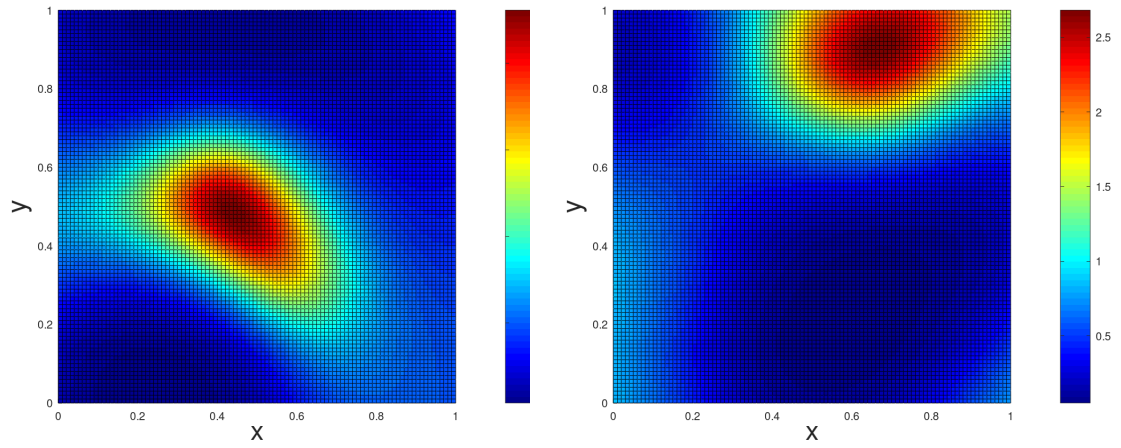


Figure 3.2 – Examples of log-normal random fields both generated by convolution of 25 i.i.d Gaussian random variables with exponential kernels with uniformly placed centers in the 5×5 grid.

polynomial basis functions. This then is utilized in multi-level and multi-index Monte Carlo settings.

3.3.1 Function spaces

Before going into details of the finite element method, we first define all relevant function spaces and their associated scalar products and norms. We define the Sobolev space

$$H^s := \left\{ v(x) : \int_D \sum_{0 \leq \alpha_1 + \ldots + \alpha_n \leq s} \left(\frac{\partial^{\alpha_1 + \ldots + \alpha_n} v}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}} \right)^2 \mathrm{d}x < \infty \right\}, \tag{3.9}$$

with the norm

$$||v||_s = \sqrt{\int_D \sum_{0 \leq \alpha_1 + \ldots + \alpha_n \leq s} \left(\frac{\partial^{\alpha_1 + \ldots + \alpha_n} v}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}} \right)^2 \mathrm{d}x}. \tag{3.10}$$

We also introduce the infinity-dimensional Lebesgue space $L^2(D)$ of square-integrable functions on $D \subset \mathbb{R}^d$ which is a special case of the Sobolev space, i.e. $L^2 \subset H^1$

$$L^2 = \left\{ v(x) : \int_D v(x)^2 \, \mathrm{d}x < \infty \right\}, \tag{3.11}$$

with the scalar product $(u, v)_{L^2(D)} = \int_D uv dx$ and norm $||v||_{L^2(D)} = (v, v)^{1/2}_{L^2(D)}$.

In addition, we introduce the Hölder space $C^t(\overline{D})$ with the following semi-norm and norm

$$|v|_{C^t(\overline{D})} = \sup_{x,y \in \overline{D}: x \neq y} \frac{|v(x) - v(y)|}{|x - y|^t} \quad \text{and} \quad \|v\|_{C^t(\overline{D})} = \sup_{x \in \overline{D}} |v(x)| + |v|_{C^t(\overline{D})}.$$

We also consider the Bochner space $L^p(\Omega, \mathcal{B})$ of p-summable $\mathcal{B}$-valued random variables

X with the norm

$$\|X\| = \begin{cases} (\int_\Omega \|X(\omega)\|_{\mathcal{B}}^p \, \mathrm{d}\mathbb{P}(\omega))^{1/p}, & \text{for} \quad p < \infty, \\ \operatorname{ess\,sup}_{\omega\in\Omega} \|X\|_{\mathcal{B}}, & \text{for} \quad p = \infty. \end{cases}$$

We introduce the following assumptions on the random field. For $p \in (0, \infty)$ we have:

1. $\kappa_{\min} \geq 0$ almost surely and $1/\kappa_{\min} \in L^p(\Omega)$;
2. $\kappa \in L^p(\Omega, C^t(\overline{D}))$ for some $0 < t \leq 1$,

where $\kappa_{\min} := \min_{x\in\overline{D}} \kappa(x;\omega)$, $\kappa_{\max} := \max_{x\in\overline{D}} \kappa(x;\omega)$.

3.3.2 Weak formulation

We derive the weak form for fixed $\omega \in \Omega$. For that we rewrite it in the following manner:

$$R(u;\omega) = 0, \tag{3.12}$$

where

$$R(u;\omega) = -\nabla \cdot \kappa(\omega)\nabla u - f(\omega), \tag{3.13}$$

is called the residual for our PDE (3.1). Next, we require the residual $R(u;\omega)$ to be orthogonal to a test space Φ

$$\int_D (-\nabla \cdot \kappa(x;\omega)\nabla u(x) - f(x;\omega))v(x)\,\mathrm{d}x = 0 \qquad \forall v \in \Phi. \tag{3.14}$$

Applying Green's theorem to (3.14), we obtain

$$\int_D \kappa(x;\omega)\nabla u(x)\cdot\nabla v(x)\,\mathrm{d}x - \int_{\Gamma_1} \kappa(x;\omega)\frac{\partial u(x)}{\partial n}v(x)\,\mathrm{d}\Gamma - \int_D f(x;\omega)v(x)\,\mathrm{d}x = 0 \quad \forall v \in \Phi. \tag{3.15}$$

In equation (3.15) we have derivatives of test functions v and as a test space Φ we can choose the space H_0^1 which is the space of square-integrable functions with square-integrable partial that vanish on the Dirichlet boundary Γ_0

$$H_0^1 := \left\{v(x) \in H^1 : v|_{\Gamma_0} = 0\right\} \subset H^1. \tag{3.16}$$

As a result of decreasing the order of the derivative of the original elliptic problem, we have weakened the requirements for our solution u. And now, the weak formulation in Galerkin form is to find a function $u \in H_g^1$ such that

$$\int_D \kappa(x;\omega)\nabla u(x)\cdot\nabla v(x)\,\mathrm{d}x = \int_D f(x;\omega)v(x)\,\mathrm{d}x \quad \forall v \in H_0^1, \tag{3.17}$$

where the trial space H_g^1 is defined as

$$H_g^1 := \left\{ v(x) \in H^1 : v|_{\Gamma_0} = u_g \right\} \subset H^1, \tag{3.18}$$

which contains functions with discontinuous derivatives.

3.3.3 Finite element spaces

We define a finite element as a triplet $\{K, P, \Sigma\}$ [21] where

- $K \subseteq R^n$ is a bounded, closed subspace with non-empty interior and with piecewise smooth boundary;
- P is a finite space of the functions defined in K with $\dim P = n$;
- $\Sigma = \{L_1, L_2, \ldots, L_n\}$ is a basis of the dual space P^* of linear forms L_i.

We assume that our bounded domain D with Lipschitz boundary is approximated by a domain D_h with piecewise smooth boundary. We also define a finite element mesh $\Xi_{h,p} = \{K_1, K_2, \ldots, K_m\}$ with polynomials of degree $p \geq 1$ of the domain $D_h \subset R^n$ as a discretization of the domain D_h such that $D_h = \cup_{i=1}^m K_i$. Then instead of looking for the solution in the entire infinite-dimensional space H_g^1, we seek an approximation in a subspace V_g^h of the finite-dimensional subspace V^h of space H^1, i.e. in $V_g^h \subset V^h \subset H^1$. We approximate the test space H_0^1 by a finite-dimensional space V_0^h, $V_0^h \subset V^h \subset H^1$. These spaces are defined as

$$V_g^h := \{v(x) \in C^0(\overline{D}) \cap H_g^1(D) : v(x)|_K \in P(K), \forall K \in \Xi_{h,p}\}, \tag{3.19}$$

$$V_0^h := \{v(x) \in C^0(\overline{D}) \cap H_0^1(D) : v(x)|_K \in P(K), \forall K \in \Xi_{h,p}\}. \tag{3.20}$$

As a result, we can now formulate the discrete variational problem of the weak form (3.17): find $u_h \in V_g^h$ such that

$$\int_D \kappa(x;\omega) \nabla u_h(x) \cdot \nabla v_h(x) \, \mathrm{d}x = \int_D f(x;\omega) v_h(x) \, \mathrm{d}x \quad \forall v_h \in V^h. \tag{3.21}$$

3.3.4 Error analysis

In this section, we list some of the important properties of the finite element solution based on the fact that our random field $G(x,\omega)$ is a log-normal random field $G(x,\omega) = \log \kappa(x,\omega)$ satisfying the assumptions A1 and A2. To present main results on the convergence and error analysis from the finite element framework, we rewrite our weak formulation (3.21) in a variational form

$$\mathcal{A}(u, v; \omega) = F(v; \omega), \tag{3.22}$$

$$\mathcal{A}(u, v; \omega) := \int_D \kappa(x; \omega) \nabla u(x) \cdot \nabla v(x) \, \mathrm{d}x, \tag{3.23}$$

$$F(v; \omega) := \int_D f(x; \omega) v(x) \, \mathrm{d}x, \tag{3.24}$$

where $\mathcal{A}(u, v; \omega)$ and $F(v; \omega)$ are called a bilinear form and a linear form, respectively. Then we consider the following important properties for our bilinear form $\mathcal{A}(\cdot, \cdot; \omega)$.

A bilinear form $\mathcal{A}(\cdot, \cdot; \omega)$ in a Hilbert space H is *continuous (bounded)* if $\exists C_0 < \infty$ such that

$$|\mathcal{A}(u, v; \omega)| \leq C_0(\omega) ||u||_H \cdot ||v||_H \quad \forall u, v \in H.$$

and also is *coercive (elliptic or positive definite)* in $V \subset H$ if there exists $\alpha > 0$ such that

$$\mathcal{A}(u, v; \omega) \geq \alpha ||v||_H^2 \quad \forall v \in V.$$

Let H be a Hilbert space and a bilinear form $\mathcal{A}(\cdot, \cdot; \omega)$ is symmetric and bounded in H and coercive in a subspace $V \subset H$ then $(V, \mathcal{A}(\cdot, \cdot; \omega))$ is also a Hilbert space, then the Lax-Milgram theorem ensures the existence and uniqueness of the weak solution for almost all random realizations ω (A1).

Theorem 3 (Lax-Milgram theorem)**.** *Let $(V, (\cdot, \cdot; \omega))$ be a Hilbert space, $\mathcal{A}(\cdot, \cdot; \omega)$ is a continuous and coercive bilinear form and $F \in V^*$ then there exists a unique solution $u \in V$ such that*

$$\mathcal{A}(u, v; \omega) = F(v; \omega) \quad \forall v \in V. \tag{3.25}$$

Lemma 1 (Céa's lemma)**.** *Let u_h be the Galerkin approximation then*

$$||u - u_h||_V \leq \frac{C_1(\omega)}{\alpha} \min_{v \in V_h} ||u - v||_V. \tag{3.26}$$

Our bilinear form defines the energy norm

$$||v||_e = \sqrt{\mathcal{A}(v, v; \omega)_e}, \tag{3.27}$$

so the Céa lemma shows that the approximation error between the finite element solution u_h and the exact solution u for the problem (3.1) in the energy norm satisfies the best approximation principle. More precisely we have

$$||u - u_h||_e^2 \leq ||u - v_g^h||_e^2 \quad \forall v_g^h \in V_g^h, \tag{3.28}$$

where V_g^h is the finite element space satisfying the Dirichlet boundary conditions. Thus, the inequality (3.28) shows that the finite element solution is the most accurate approximation of the exact solution u in the energy norm (3.27) of the finite element space V_g^h.

Theorem 4. *The approximation error of the finite element solution in the L_2-norm is [33]*

$$||u - u_h||_{L^2} \leq C_3(\omega)h^{p+1}, \tag{3.29}$$

where u is the exact solution, u^h is the finite element solution, the constant $C(\omega)$ depends only on the problem operator and the domain, h is the length of the elements, and p is the order of the elements.

3.3.5 Finite element matrices

Every function $v_h \in V_0^h$ can be expressed as a linear combination

$$v_h(x) = \sum_{i \in N} w_i \psi_i(x), \tag{3.30}$$

where n is the total number of basis functions, $N := \{1, \ldots, n_0, \ldots, n\}$ is the index set of i of basis functions ψ_i of the space V_0^h and n_0 is the number of nodes corresponding to the Dirichlet boundary conditions. Then the variational formulation is equivalent to the following system of linear equations

$$\int_D \kappa(x;\omega)\nabla u_h(x) \cdot \nabla \psi_i(x)\,\mathrm{d}x = \int_D f(x;\omega)\psi_i(x)\,\mathrm{d}x, \quad i \in N. \tag{3.31}$$

The function $u_h \in V_g^h$ can be represented as a linear combination in the space V^h

$$u_h(x) = \sum_{j=1}^{n} w_j \psi_j(x), \tag{3.32}$$

the n_0 weights w_j which correspond to the Dirichlet boundary condition are fixed

$$u_h|_{\Gamma_0} = u_g. \tag{3.33}$$

Substituting (3.32) into (3.31) we get the system of linear algebraic equations

$$\sum_{j=1}^{n} \left(\int_D \kappa(x;\omega)\nabla \psi_j(x) \cdot \nabla \psi_i(x)\,\mathrm{d}x \right) w_j = \int_D f(x;\omega)\psi_i(x)\,\mathrm{d}x, \quad i \in N. \tag{3.34}$$

Then, by solving this system we obtain the finite element solution u_h.

3.3.6 Hierarchical polynomial basis

The choice of basis functions depends on the desired properties for the computational method. While the Lagrange polynomial basis is simple to construct; the hierarchical basis functions are advantageous in terms of the condition number of the resulting system of linear equations and, thus, may improve the convergence of iterative solver methods [102].

We define hierarchical basis as a family $\{\mathfrak{B}_k\}_{k\geq 0}$ of sets of polynomials $\mathfrak{B}_k$ such that $\forall k \geq 0$ $\mathfrak{B}_k$ is a basis for P_k and $\mathfrak{B}_k \subset \mathfrak{B}_{k+1}$.

Hierarchical basis functions on triangular elements. We define a set of hierarchical basis functions on the triangle (Figure 3.3) with vertices $(\hat{x}_0, \hat{y}_0)$, $(\hat{x}_1, \hat{y}_1)$, $(\hat{x}_2, \hat{y}_2)$ via its barycentric coordinates $\mathcal{L}_i$ [102]

$$\mathcal{L}_i(x, y) = \alpha_{i0} + \alpha_{i1}x + \alpha_{i2}y, \quad i = 0, 1, 2, \tag{3.35}$$

so that each function $\mathcal{L}_i$ is equal to 1 on the vertex $(\hat{x}_i, \hat{y}_i)$ and zero on the other vertices of the triangle Ω_k. The first three basis functions are the usual $\mathcal{L}$ coordinates:

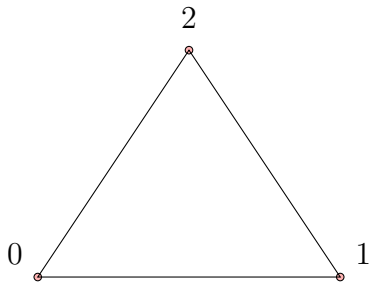


Figure 3.3 – A triangle with nodes defining linear basis functions.

$$\phi_0 = \mathcal{L}_0, \qquad \phi_1 = \mathcal{L}_1, \qquad \phi_2 = \mathcal{L}_2. \tag{3.36}$$

To make the approach second order we need to add another three quadratic functions associated with the edges:

$$\phi_3 = \mathcal{L}_0\mathcal{L}_1, \qquad \phi_4 = \mathcal{L}_1\mathcal{L}_2, \qquad \phi_5 = \mathcal{L}_0\mathcal{L}_2. \tag{3.37}$$

We add three more cubic functions associated with the edges

$$\phi_6 = \mathcal{L}_0\mathcal{L}_1(\mathcal{L}_0 - \mathcal{L}_1), \qquad \phi_7 = \mathcal{L}_1\mathcal{L}_2(\mathcal{L}_1 - \mathcal{L}_2), \qquad \phi_8 = \mathcal{L}_0\mathcal{L}_2(\mathcal{L}_0 - \mathcal{L}_2), \tag{3.38}$$

and a one function associated with the center of the triangle

$$\phi_9 = \mathcal{L}_0\mathcal{L}_1\mathcal{L}_2, \tag{3.39}$$

to increase the order of the elements further by one. More generally, to construct a basis of the order p we need to add three basis functions associated with the edges and $p-2$ basis functions associated with the center of the element to the set of basis functions of the order $p-1$

$$\phi_i = \mathcal{L}_0\mathcal{L}_1 \cdot P_p(\mathcal{L}_0 - \mathcal{L}_1), \quad \phi_{i+1} = \mathcal{L}_1\mathcal{L}_2 \cdot P_p(\mathcal{L}_1 - \mathcal{L}_2), \quad \phi_{i+2} = \mathcal{L}_0\mathcal{L}_2 \cdot P_p(\mathcal{L}_0 - \mathcal{L}_2), \tag{3.40}$$

$$\phi_{i+3+j} = \mathcal{L}_0\mathcal{L}_1\mathcal{L}_2 \cdot P_p(\mathcal{L}_0 - \mathcal{L}_1) \cdot P_{p-2-j}(2\mathcal{L}_2 - 1), \quad j = 0 \dots p-3, \tag{3.41}$$

where i is the number of basis functions of the order $p-1$ and $P_p(\xi)$ is a polynomial of order p. For example, one can choose the most simple form $P_p(\xi) = \xi^p$.

Hierarchical basis functions on rectangular elements. While triangular elements allow us to work with complex domain geometry, one can exploit the simplicity of constructing polynomials for rectangular elements. We use rectangular elements to obtain incomplete polynomials which we utilize later in the multi-index Monte Carlo methods. For example, the Pascal triangle (Figure 3.4) shows the number of terms needed in order to construct the complete cubic polynomial and Figure 3.5 shows the number of terms to construct incomplete polynomials of a different order in x and y directions.

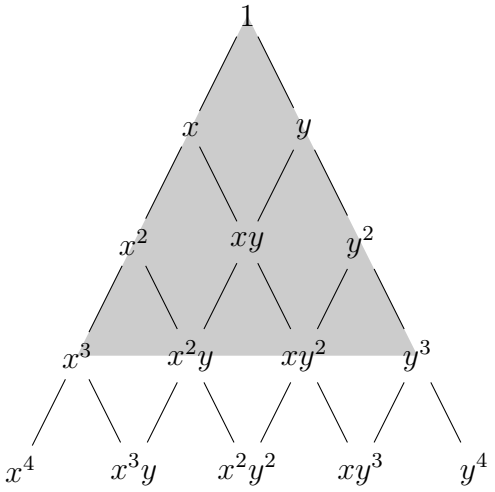


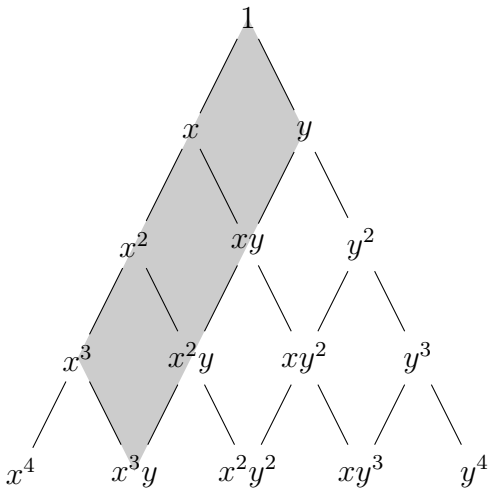
Figure 3.4 – Complete cubic expansion shaded in 2D – 10 terms [102].

We construct basis functions on rectangular elements as the tensor product of one-dimensional hierarchical basis functions for x and y coordinates,

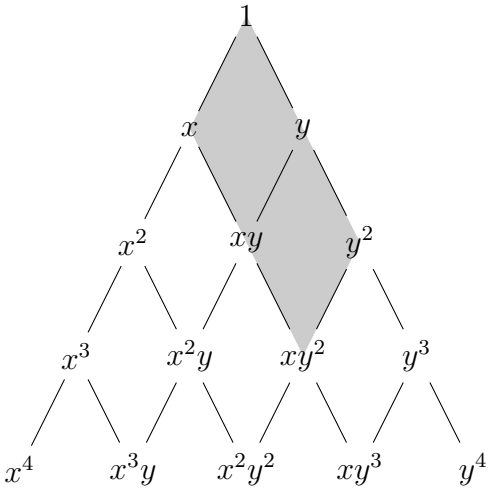
$$\phi_i(x,y) = \hat{\phi}_l(x)\hat{\phi}_m(y), \quad l = 1, \dots, L, \quad m = 1, \dots M, \tag{3.42}$$

where M and L are the numbers of basis functions in x and y coordinates, respectively. The formulation of basis functions in the tensor form allows us to specify the order of polynomials in any particular direction, x or y. As a result, we are able to use incomplete finite elements (Figure 3.7). For example, in Figure 3.7a the order of polynomial in x direction is one, while in y direction the second order is used.

We want our basis to satisfy certain conditions on the orthogonality with respect to the inner product $\int_D \nabla\psi_j \cdot \nabla\psi_i \, \mathrm{d}x = 0$ for at least some i, j. Since Legendre polynomials possess this property, we apply them in the construction of the hierarchical basis. They



(a) Cubic expansion into direction x and linear expansion into direction y; $p_x = 3, p_y = 1$.



(b) Linear expansion into direction x and quadratic expansion into direction y; $p_x = 1, p_y = 2$.

Figure 3.5 – Examples of incomplete polynomials in 2D for rectangular finite elements.

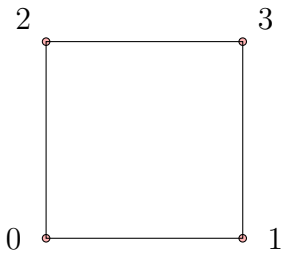
can be defined in several ways such as

$$P_p(\xi) = \frac{1}{p!}\frac{1}{2^p}\frac{d^p}{d\xi^p}\left((\xi^2-1)^p\right), \tag{3.43}$$

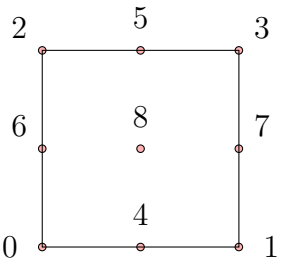
or using a recurrence relation

$$P_p(\xi) = \frac{2p-1}{p}\xi P_{p-1}(\xi) - \frac{p-1}{p}P_{p-2}(\xi), \tag{3.44}$$

with $P_0(\xi) = 1$ and $P_1(\xi) = \xi$.



(a) First-order rectangle with nodes defining basis.



(b) Second-order rectangle with nodes defining basis.

Figure 3.6 – Complete rectangular finite elements with nodes defining a linear basis (left) and a second order basis (right).

Hierarchical basis functions can be obtained from the Legendre polynomials as

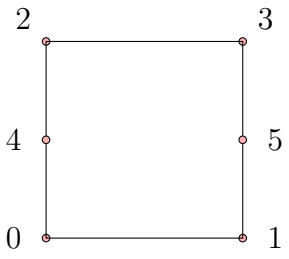
$$\hat{\phi}_p(\xi) = c_p\left(P_p(\xi) - P_{p-2}(\xi)\right), \quad p \geq 2, \tag{3.45}$$

where c_p is a non-zero coefficient that can be chosen arbitrarily. The first four basis

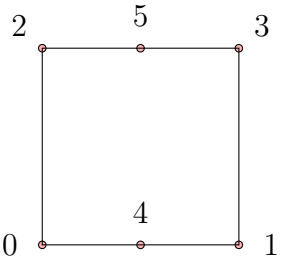
functions are

$$\hat{\phi}_0(\xi) = \frac{1}{2}(1-\xi), \quad \hat{\phi}_1\xi = \frac{1}{2}(1+\xi), \tag{3.46}$$

$$\hat{\phi}_2(\xi) = \xi^2 - 1, \quad \hat{\phi}_3(\xi) = \xi^3 - \xi. \tag{3.47}$$



(a) Rectangular element with nodes defining a basis with order $p_x = 1$, $p_y = 2$.



(b) Rectangular element with nodes defining a basis with order $p_x = 2$, $p_y = 1$.

Figure 3.7 – Examples of incomplete finite elements with nodes.

3.3.7 Numerical quadrature

In general, analytical expressions of integrals $\int_{D_k} \psi_i \psi_j \,\mathrm{d}x$ and $\int_{D_k} \nabla\psi_i \nabla\psi_j \,\mathrm{d}x$ can be tedious to obtain [2, 80]. Instead, we use Gaussian quadrature to perform numerical integration to assemble the global matrix A_h

$$\int_D f(x) \,\mathrm{d}x \approx \sum_{i=1}^{n} \hat{w}_i f(x_i), \tag{3.48}$$

where n is the number of nodes x_i and w_i are weights.

3.4 *hp*-Multi-level Monte Carlo method for elliptic PDEs

3.4.1 Model problem

For the testing purposes, we consider the same problem setup for both multi-level and multi-index Monte Carlo methods. To demonstrate the efficiency of our methods we consider two functionals as our quantities of interest (QoI). The first $Q_1(u)$ is the average of the solution $u(x;\omega)$ stored in a given volume $V = [0.125; 0.375] \times [0.125; 0.625]$ of the unit domain $D = [0;1] \times [0;1]$ (Figure 3.8). This way, we have

$$Q_1(u) = \int_V u(x) \,\mathrm{d}x. \tag{3.49}$$

The second QoI is the average of flux at three locations

$$Q_2(u) = \frac{1}{3}\sum_{i=1}^{3} \|\nabla u(x_i)\|, \tag{3.50}$$

where $x_1 = [0.5; 0.5]$, $x_2 = [0.21; 0.78]$, and $x_3 = [0.23; 0.3]$.

As a test problem, we consider our elliptic equation in two dimensions with four Gaussian sources (Figure 3.9a) in the following form

$$f(x;\omega)_k = f(x)_k = \frac{c_k}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_k)^T(x-\mu_k)}{2\sigma^2}\right), \quad k = 1\dots 4. \tag{3.51}$$

centered at μ_k where $\mu_1 = [0.25; 0.25], \mu_2 = [0.25; 0.75], \mu_3 = [0.75; 0.25], \mu_4 = [0.75; 0.75]$ with weights $\{c_k : -100, 100, 100, -100\}$, and with the same standard deviation $\sigma^2 = 0.005$. We also note that the sources are independent of the random variable ω.

The boundary conditions are

$$u|_{\Gamma_0} = \begin{cases} 0, & y = 0, \\ 1, & y = 1, \end{cases} \tag{3.52}$$

$$n \cdot \kappa \nabla u|_{\Gamma_1} = 0, \tag{3.53}$$

with boundaries $\Gamma_0 = [0; 0] \times [0; 1] \cup [1; 0] \times [1; 1]$ and $\Gamma_1 = [0; 0] \times [1; 0] \cup [0; 1] \times [1; 1]$.

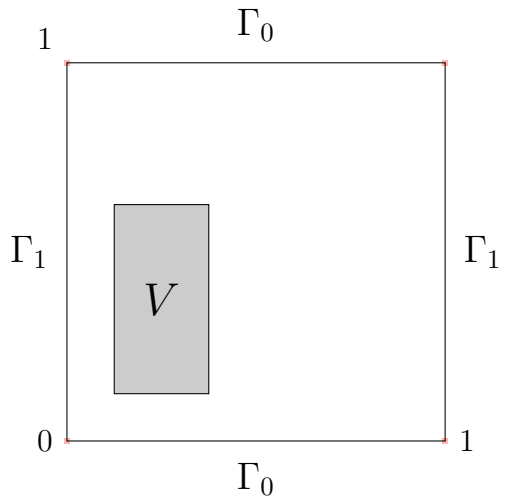
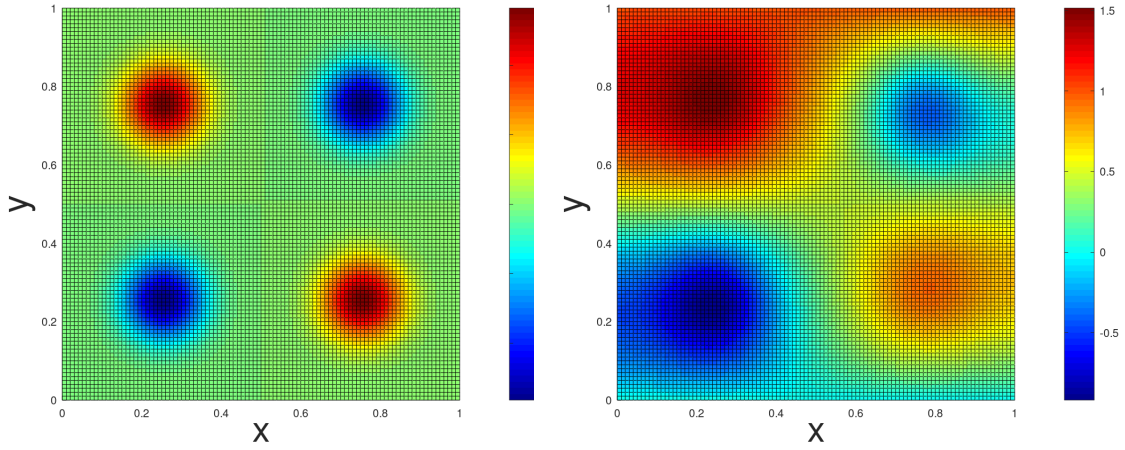


Figure 3.8 – A unit domain containing the volume V in which the average of solution u presents the first QoI.

As mentioned before, we use a convolution process $\log \kappa(x;\omega)$ as our log-normal random field. For the test case, we construct the field by convoluting 25 i.i.d. Gaussian random variables

$$\log \kappa(x;\omega) = \sum_{i=1}^{25} \omega_i k(x - c_i), \tag{3.54}$$

where our smoothing kernel $k(x-c_i) = \exp\left[-\frac{25}{2}\|x - c_i\|\right]$ with centers c_i placed uniformly as a grid 5×5 in our domain D. An example of a random field is shown in Figure 3.10.



(a) Four Gaussian sources. (b) Solution u for a random realization ω.

Figure 3.9 – Source $f(x)$ and a solution $u_\omega(x)$ for a single realization of random field.

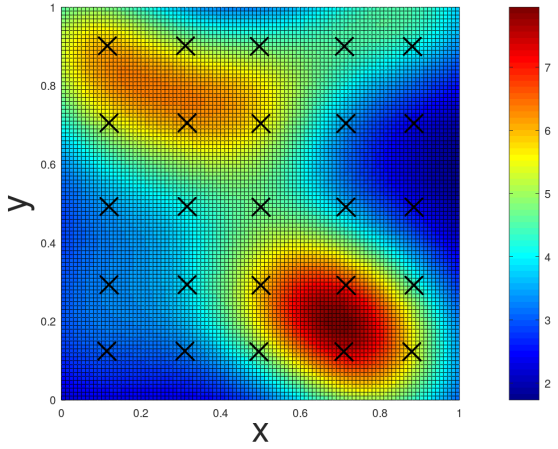


Figure 3.10 – Log-normal random field for a single realization ω. Kernels used for defining the convolution process are marked by crosses.

In the multi-level Monte Carlo based methods, we use a mesh with 128 number of triangular elements, corresponding to the mesh size $h_0 = 1/8$ as the coarsest level.

Since the finite element matrices are nonsymmetric, we use the biconjugate gradient stabilized (BiCGStab) iterative solver with a given threshold ε_ℓ depending on the discretized mesh to find the solution of the linear systems. Based on the convergence theorem for the finite element solution given in Section 3.3.4, we may use the following stopping criteria for the BiCGStab solver

$$\varepsilon_\ell = \tilde{C}\|\mathbb{E}[Q_l - Q_{l-1}]\|/2^{p+1}, \tag{3.55}$$

where p is the basis order and $\tilde{C}$ is a small constant $0 < \tilde{C} < 1$, in our cases, $\tilde{C} = 10^{-2}$. This

enforces the threshold ε_ℓ to be smaller than the discretization error of the finite element method. For the first two levels in each MLMC method, we solve the resulting linear systems almost up to the machine precision error, $\varepsilon_0 = \varepsilon_1 = 10^{-14}$. This is because the *a priori* discretization error for the initial level is unknown. But as soon as the discretization bias is obtained, we are able to derive the error bounds for the further levels based on the theoretical assumptions about the convergence rate of a given approximation method.

3.4.2 *hp*-Multi-level Monte Carlo methods

In Chapter 2, we described the case of a general multi-level Monte Carlo method (in which the quantity of interest Q_L corresponds to the solution obtained on the finest level L). Here we use four different strategies to define model hierarchies in the multi-level Monte Carlo algorithm.

***h*-MLMC with the linear basis functions ($p = 1$).** This is the standard, so-called "geometric" multi-level Monte Carlo method which utilizes a sequence of discretized elliptic PDE models constructed in such a way that the mesh resolution is doubled with level increase. The initial level consists of the grid with the first order linear basis functions ($p = 1$) and the characteristic length (mesh size in one dimension) $h_0 = 1/8$. The subsequent levels are constructed in the following form

$$h_\ell = 2^{-\ell} h_0. \tag{3.56}$$

Then $\{\Xi_h\}_{h>0}$ is a family of (quasi)-uniform, triangular, conforming finite element meshes on the spatial domain D_h with corresponding nested spaces $V_\ell \equiv V_g^{h_\ell}$

$$V_L \subset \ldots \subset V_\ell \subset V_{\ell-1} \subset \ldots \subset V_0.$$

The geometric MLMC has been applied extensively for a variety of cases since the introduction of MLMC, see [10, 66, 76], and [8, 22, 37, 38, 87, 96] specifically for elliptic problems. While in many cases it is simple to construct the multi-level sequence, the main difficulty comes from the numerical analysis of the resulting estimators. In [20], Charrier et al. provided the following theoretical bounds on the multi-level expectation of linear functionals as QoIs. In our case, the multi-level expectation for the first QoI $Q_1(u)$ (3.49) is

$$|\mathbb{E}[Q_\ell(\omega) - Q_{\ell-1}(\omega)]| \leq C_{1,1} 2^{-2\ell},$$

while the variance is bounded by

$$|\mathbb{V}[Q_\ell(\omega) - Q_{\ell-1}(\omega)]| \leq \mathbb{E}[Q_\ell - Q_{\ell-1}]^2 \leq C_{2,1} 2^{-4\ell},$$

where the constants $C_{1,1}$ and $C_{2,1}$ are independent of the random variable ω and of the grid size h.

h-MLMC with the second order basis functions ($p = 2$). This is a modified version of the previous method but, instead of using linear basis functions, we exploit the use of the second basis functions which increases the smoothness of the discretized solution.

hp-MLMC. In this method, we define our approximate sequence in such a way that we increase the polynomial order of basis functions with each mesh refinement. We start with the same initial mesh setup $h_0 = 1/8$ and $p_0 = 1$. Then the approximation on the level ℓ is defined as

$$h_\ell = 2^{-\ell} h_0 \quad \text{and} \quad p_\ell = \ell + 1. \tag{3.57}$$

The method has been proposed in [10] for the compressible Navier-Stokes equations coupled with the discontinuous Galerkin method. The authors combined p-MLMC [78] and h-MLMC into hp-MLMC and provided a generalization of the complexity.

p-MLMC. In our last multi-level-based method, we obtain the model sequence by simple incrementing the polynomial order of the basis functions without any further refinement in the grid resolution. In this method, we construct our approximation sequence in this form

$$h_\ell = \text{const} = h_0 \quad \text{and} \quad p_\ell = \ell + 1. \tag{3.58}$$

This method was proposed in [78] for hyperbolic problems using a high-order discontinuous Galerkin method. The authors obtained the following theoretical estimations on the multi-level expectation for hyperbolic problems

$$|\mathbb{E}[Q_\ell - Q_{\ell-1}]| \leq C_{1,4} 2^{-p_\ell},$$

and on the variance

$$|\mathbb{V}[Q_\ell - Q_{\ell-1}]| \leq C_{2,4} 2^{-2p_\ell}.$$

3.4.3 Numerical results

Figures 3.11a and 3.11b show the absolute value of the mean, $|\mathbb{E}[Q_\ell - Q_{\ell-1}]|$, and the variance, $\mathbb{V}[Q_\ell - Q_{\ell-1}]$, for levels $\ell = 0, \ldots, 3$. The slope of the function $\log_2 \mathbb{V}[Q_\ell - Q_{\ell-1}]$ is about -4 for the h-MLMC with $p = 1$ and -8 for the h-MLMC with the second order basis functions (corresponding to $\beta_{h\text{-MLMC},p=1} \approx 4$ and $\beta_{h\text{-MLMC},p=2} \approx 8$, respectively, in Theorem 1). The line for $\log_2 |\mathbb{E}[Q_\ell - Q_{\ell-1}]|$ has a slope of approximately -2 for the h-MLMC with $p = 1$, corresponding to $\alpha_{h,p=1} = 2$ in Theorem 1. This implies an $O(h^2)$ convergence rate for the h-MLMC with $p = 1$, similar to the convergence in L_2-norm of the finite element solution with the linear basis functions.

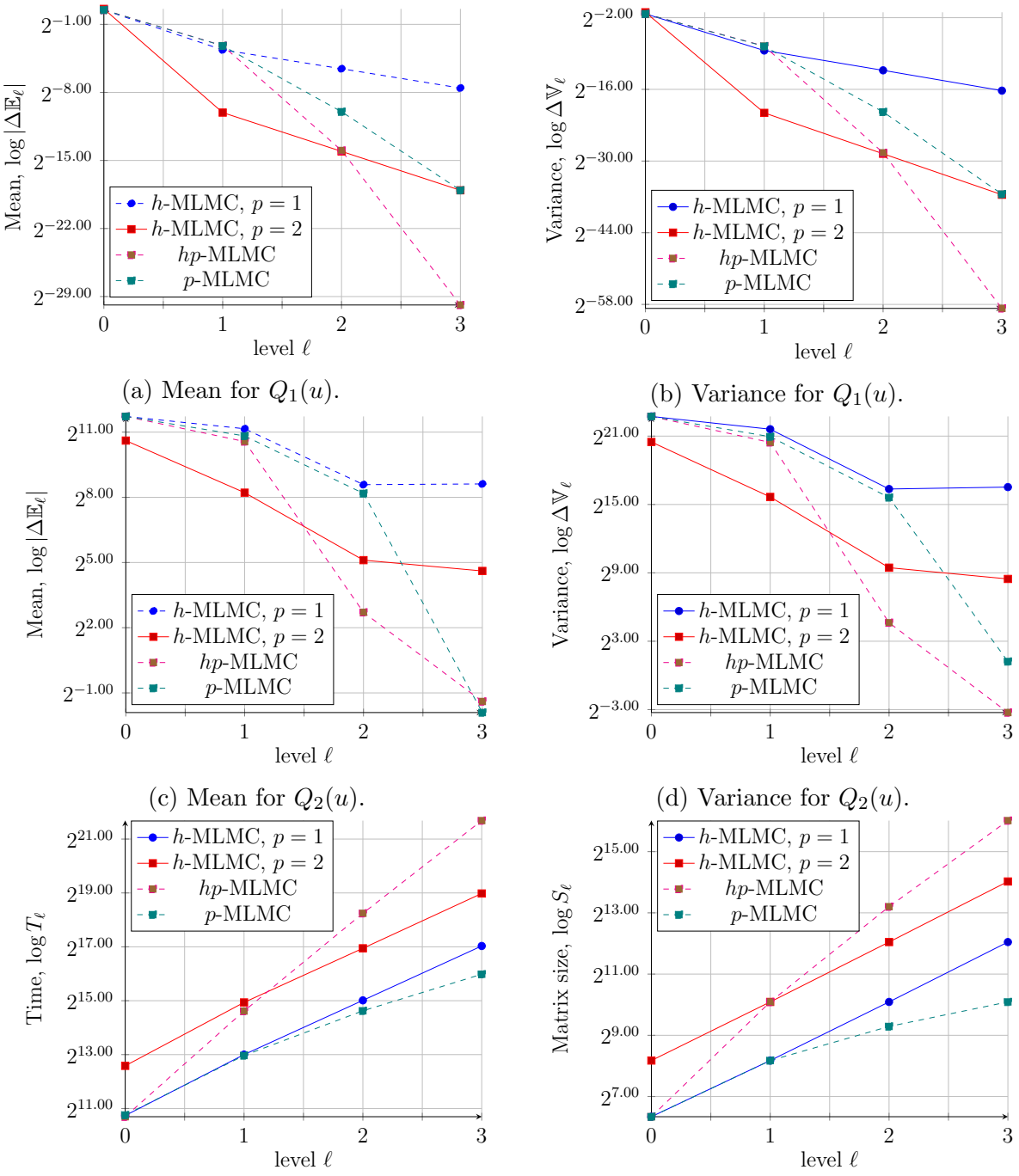
On the other hand, the slope of the line for $\log_2 |\mathbb{E}[Q_\ell - Q_{\ell-1}]|$ is about -4 for the h-MLMC with $p = 2$ (corresponding to $\alpha_{h,p=2} = 4$ in Theorem 1). This implies an $O(h^4)$ convergence rate for the h-MLMC with $p = 2$, which differs from the convergence in L_2-norm of the finite element solution with the second order basis functions ($O(h^3)$).

For other two methods, hp-MLMC and p-MLMC, the functions $\log_2 |\mathbb{E}[Q_\ell - Q_{\ell-1}]|$ in Figure 3.11b are no longer linear, and thus, the variance reduction rates β are non-constant. The same results are observed in Figure 3.11a. These two methods have much higher convergence rate compared to the first two methods.

Figure 3.11e shows that the computational cost increasing factor γ for the geometric multi-level Monte Carlo method with $p = 1$ and $p = 2$ is about 2. For the hp-MLMC the cost increasing factor is 4. Finally, for the p-MLMC the cost increases at a constant rate 1.5. Thus, for the first quantity of interest (QoI) $Q_1(u)$, the variance reduction factor β in all cases is bigger than the cost increasing factor γ as shown in Figures 3.11b and 3.11e. This delivers the optimal complexity $O(\varepsilon^{-2})$ (Chapter 2, Theorem 1). The numerical experiments confirm this complexity as we can see from Figure 3.12a. All the curves lie on the same line except the curve for the h-MLMC with $p = 2$, where we use the second order basis functions on the coarsest level. Hence, the most computations are done on the coarsest level for all presented multi-level Monte Carlo methods. Overall, the h-MLMC method with $p = 2$ would be the worst for this QoI.

For the second QoI $Q_2(u)$, the situation is quite different. In Figure 3.11d, we observe that the variance reduction rate for the h-MLMC with linear basis functions is non-constant compared to the previous QoI. Moreover the variance increases from level 2 to level 3, which indicates the poor performance of $p = 1$ for this type of QoI. The reason behind this behaviour is that the output QoI is a discontinuous function of the intermediate solution u. As a result, this leads to a larger variance, and hence lower value for β. In comparison, the variance reduction rate for the hp- and p-MLMC is much greater than those of two other methods, their computational cost (Figure 3.12b) is slightly higher than using h-MLMC with $p = 2$.

In conclusion, all the methods showed similar computational complexity for the first quantity of interest. For QoIs with discontinuous output functions (in our example, it is $Q_2(u)$), it seems that the use of higher order methods, such as h-MLMC with $p \geq 2$, will yield better approximation properties and eliminates unnecessary fluctuations in the mean and variance.



(a) Mean for $Q_1(u)$.

(b) Variance for $Q_1(u)$.

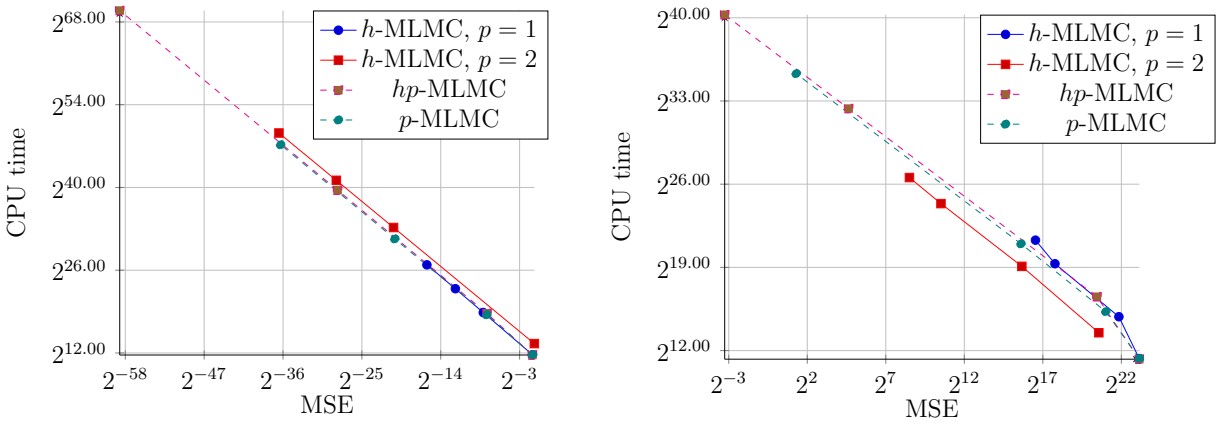
(c) Mean for $Q_2(u)$.

(d) Variance for $Q_2(u)$.

(e) Computational time (BiCGStab).

(f) Matrix size.

Figure 3.11 – MLMC numerical results for both QoIs using four different strategies at each level. (a) and (b): Absolute value of mean and variance for $Q_1(u)$. (c) and (d): Absolute value of mean and variance for $Q_2(u)$. (e): Total computational time at each level. (d): Matrix sizes resulting from the finite element approximation.



(a) $Q_1(u)$: average solution u in volume V.

(b) $Q_2(u)$: average of flux at three points.

Figure 3.12 – CPU time vs. MSE, BiCGStab solver for four developed MLMC strategies. *Left*: Average solution u in the volume V. *Right*: Average of flux $Q_2(u)$.

3.5 *hp*-Multi-index Monte Carlo method for elliptic PDEs

In this section, we propose a novel multi-index Monte Carlo method based on the incomplete polynomials (Section 3.3.6) in 2D. We compare the results with the standard geometric MIMC method as well as with MLMC methods described in the previous section for the same quantities of interest (3.49) and (3.50).

3.5.1 *hp*-Multi-index Monte Carlo method

We begin with a description of the MIMC methods used to perform numerical experiments. The first one is the standard multi-index Monte Carlo ($h_x h_y$-MIMC) method described in Chapter 2. This method in its original form was introduced by Haji-Ali et al. [43] for solving 2D elliptic problems with random coefficients as an extension of MLMC. It uses directional refinements h_x and h_y in two spatial directions x and y, which we denote as $h_1 := h_x$ and $h_2 := h_y$ for simplicity. Moreover, we use uniform meshes with standard bilinear basis functions on rectangular elements to discretize the weak form of the elliptic PDE. The number of elements in each dimension for level $\ell_i \in \mathbb{N}_0$ is $N_i^{\ell_i}$, to give a mesh size of $h_i^{\ell_i} = h_0 N_i^{-\ell_i}$ for all $i = 0, 1, 2$. In other words, given a multi-index $\boldsymbol{\ell} = (\ell_1, \ell_2)$, we use the following number of elements $N_i = 8 \cdot 2^{\ell_i}$ in each direction.

In the second method, we use uniform meshes with complete and incomplete rectangular finite element basis functions to approximate our test problem. We define a multi-index $\boldsymbol{\ell} = (\ell_1, \ell_2)$ in such a way that we refine not only in each dimension but also in each polynomial order x and y. The resulting level $\boldsymbol{\ell} = (\ell_1, \ell_2)$ uses

$$h_i^{\ell_i} = h_0 N_i^{-\ell_i} \quad \text{and} \quad p_i^{\ell_i} = \ell_i + 1, \quad i = 0, 1, 2. \tag{3.59}$$

We denote this method as $h_x p_x, h_y, p_y$-MIMC.

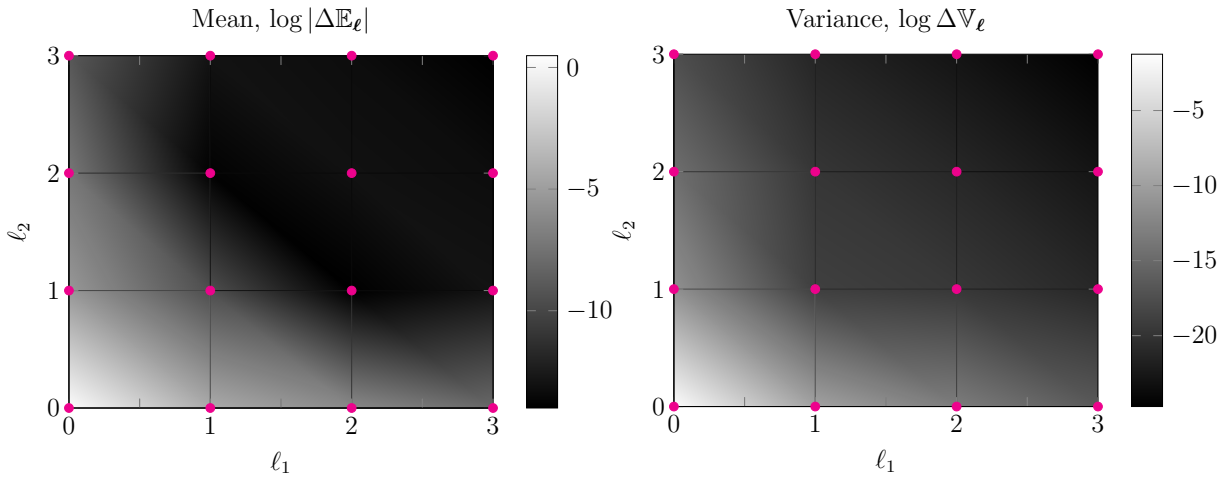
3.5.2 Numerical results

Figures 3.13a, 3.13b, and 3.13e illustrate that the assumptions in Theorem 2 (Chapter 2, Section 2.4) are indeed satisfied for the first QoI. Specifically, these figures indicate that our QoI satisfies the mixed regularity property. Thus, we observe that the variance reduction rate in each direction is bigger than the cost increase. In addition, Figure 3.15a provides numerical evidence to this claim and also indicates the optimal complexity $O(\varepsilon^{-2})$ for the $h_x h_y$-MIMC method.

Similar, Figures 3.14a, 3.14b, and 3.14e demonstrate the same results for the $h_x p_x, h_y p_y$-MIMC method. Moreover, the slopes of the lines shown in Figure 3.15a are almost identical. This implies the same optimal complexity $O(\varepsilon^{-2})$ as in case of the standard MIMC.

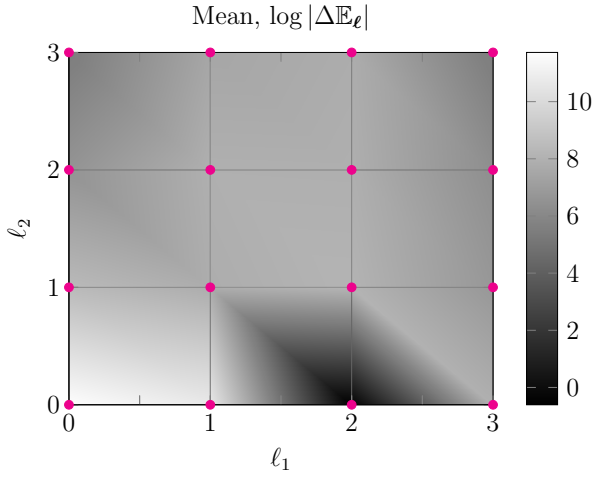
In case of the point value QoI ($Q_2(u)$), the mixed regularity property is no longer

satisfied for the $h_x h_y$-MIMC as it is shown in Figures 3.13c and 3.13d. On the other hand, Figures 3.13c and 3.13d show that the $h_x p_x, h_y p_y$-MIMC performs better in this case. As it can be seen in Figure 3.15b, both methods yield the best optimal complexity $O(\varepsilon^{-2})$ for this QoI with the standard MIMC method costing slightly more.

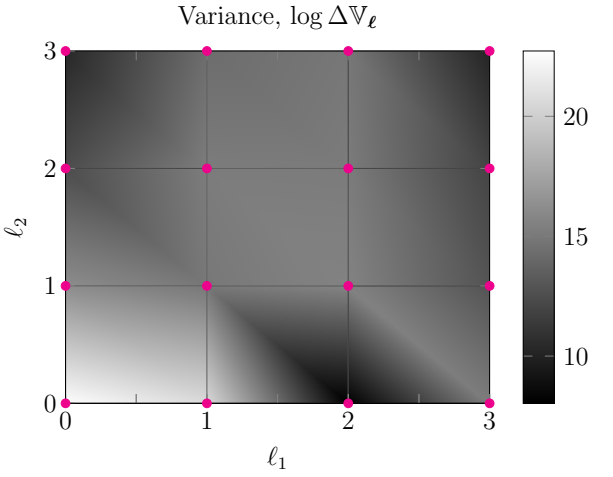


(a) Mean for $Q_1(u)$.

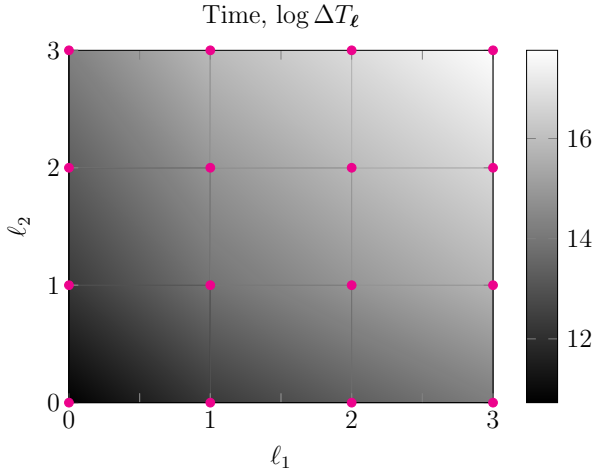
(b) Variance for $Q_1(u)$.



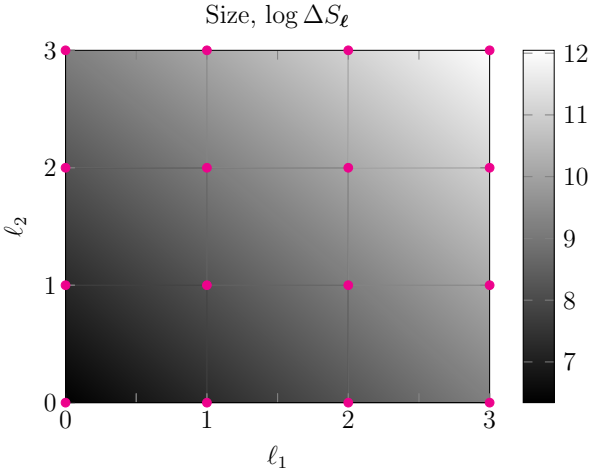
(c) Mean for $Q_2(u)$.



(d) Variance for $Q_2(u)$.

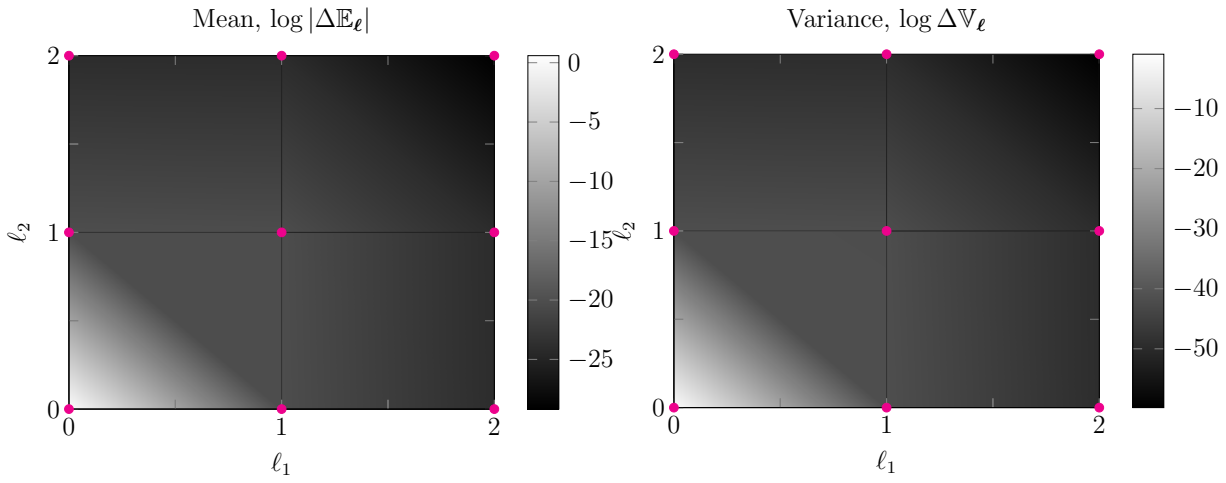


(e) Computational time.



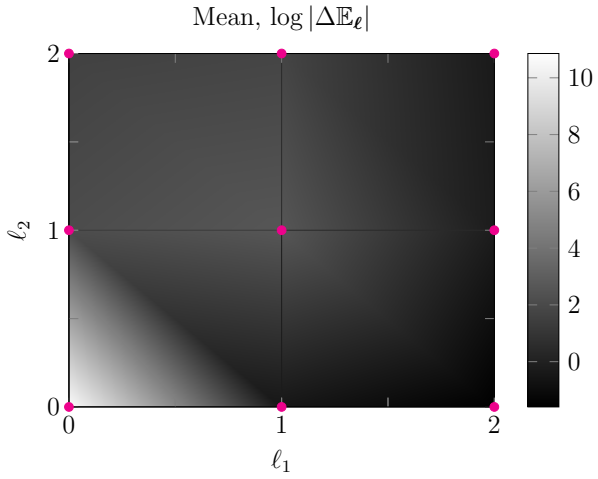
(f) Matrix size.

Figure 3.13 – $h_x h_y$-MIMC numerical results for both QoIs $Q_1(u)$ and $Q_2(u)$. (a) and (b): Logarithmic values of mean and variance for the average solution u in volume V. (c) and (d): Logarithmic values of mean and variance for the average of flux $Q_2(u)$. (e): Total computational time at each index level (ℓ_1, ℓ_2) using BiCGStab. (f): Matrix sizes resulting from finite element discretization.

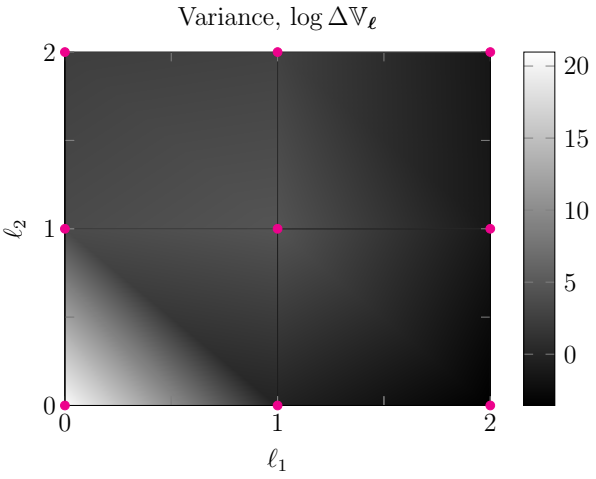


(a) Mean for $Q_1(u)$.

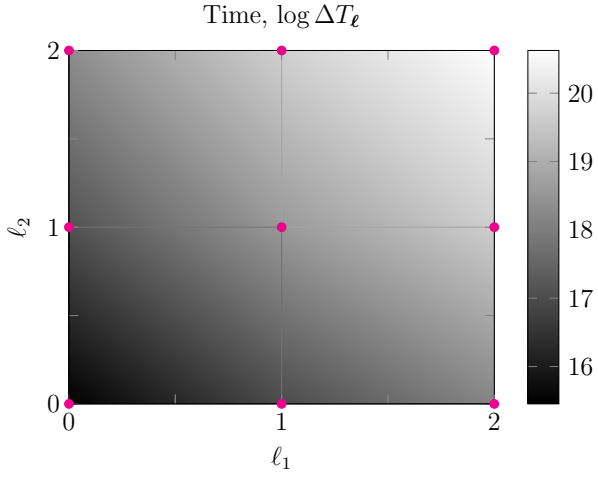
(b) Variance for $Q_1(u)$.



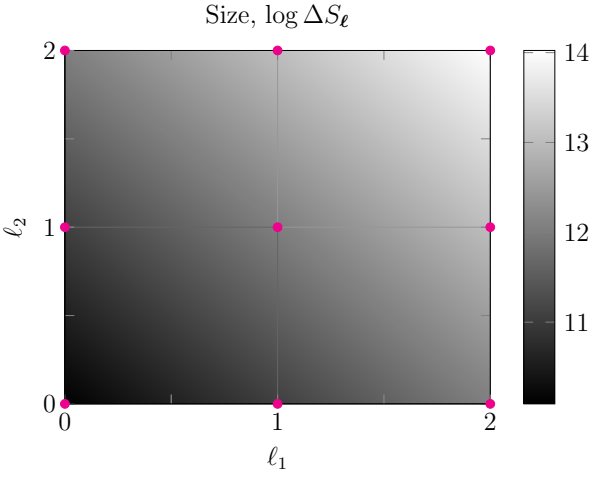
(c) Mean for $Q_2(u)$.



(d) Variance for $Q_2(u)$.

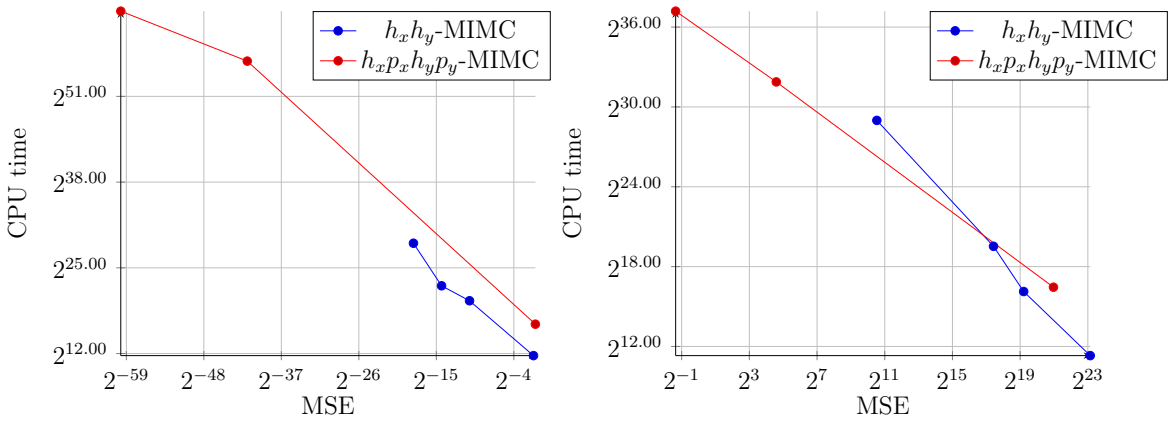


(e) Computational time.



(f) Matrix size.

Figure 3.14 – $h_x p_x, h_y p_y$-MIMC numerical results for both QoIs $Q_1(u)$ and $Q_2(u)$. (a) and (b): Logarithmic values of mean and variance for the average solution u in volume V. (c) and (d): Logarithmic values of mean and variance for the average of flux $Q_2(u)$. (e): Total computational time at each index level (ℓ_1, ℓ_2) using BiCGStab. (f): Matrix sizes resulting from finite element discretization.



(a) $Q_1(u)$: average solution u in volume V. (b) $Q_2(u)$: average of flux at three points.

Figure 3.15 – CPU time vs. MSE using BiCGStab solver for h_xh_y-MIMC and h_xp_x, h_yp_y-MIMC. *Left*: Average solution u in the volume V. *Right*: Average of flux, $Q_2(u)$.

Chapter 4

Multi-level Monte Carlo method for the convection-diffusion eigenvalue problem

In this chapter we develop new multi-level Monte Carlo (MLMC) methods based on the grid refinement and homotopy method for finding the smallest eigenvalues of the stochastic convection-diffusion operator. The eigenvalue problem is described in the first section followed by a derivation of a finite element approximation. After that, we discuss the Rayleigh quotient (RQ) iteration and the implicitly restarted Arnoldi (IRA) method as the eigenvalue solvers. Then, we apply a homotopy continuation method to the convection-diffusion operator for solving the eigenvalue problem. Various numerical simulations are performed in order to show several uses of the developed multi-level Monte Carlo methods based on different combinations of the mesh and homotopy discretization. We propose two classes of MLMC method. One class is the geometric MLMC and the other is the homotopy MLMC. It is known that the convection-diffusion problem requires fine grid discretizations in order to have a stable solution. As such, for cases with high velocity, we utilize the solution of the pure diffusion problem as a means of introducing an additional level with a coarse mesh. We analyze both eigenvalue solvers, the Rayleigh quotient and the implicitly restarted Arnoldi iterations, in terms of the number of required iterations and the total computational time for each of these two classes. At last, we give a comparison between all these methods which includes the standard Monte Carlo method as well.

4.1 Convection-diffusion eigenvalue problem

We consider a convection-diffusion eigenvalue problem with random coefficients: find a non-trivial eigenpair $(\lambda, u) \in \mathbb{C} \times H_0^1(D;\mathbb{C})$ with normalized eigenfunction $\|u\|_{L^2} = 1$ such that

$$\mathbf{a}(x;\omega)\cdot\nabla u(x;\omega) - \nabla\cdot\kappa(x;\omega)\nabla u(x;\omega) = \lambda(\omega)u(x;\omega), \tag{4.1}$$

in a given probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and a bounded Lipschitz domain $D \in \mathbb{R}^d$ for $d = 1, 2,$ or 3 with boundary Γ for almost all random variables $\omega \in \Omega$. The velocity $\mathbf{a}(x;\omega) : D \times \Omega \to \mathbb{R}^d$ and the conductivity $\kappa(x;\omega) : D \times \Omega \to \mathbb{R}$ are random processes. The Dirichlet boundary conditions on the boundary Γ are given by

$$u|_\Gamma = 0. \tag{4.2}$$

We also consider velocity $\mathbf{a}$ being divergence-free almost surely (a.s.)

$$\nabla \cdot \mathbf{a}(x;\omega) = 0. \tag{4.3}$$

The goal is to compute the expectation of the smallest eigenvalue

$$\mathbb{E}[\lambda(\omega)] = \int_\Omega \lambda(\omega)\, \mathrm{d}\omega.$$

Stochastic eigenvalue problems are common in a wide range of scientific and engineering applications. Typical applications include determining the geometry to design a nuclear reactor for criticality [4, 5, 32, 56, 104], finding the natural frequencies of a given aircraft or a naval vessel [57], obtaining the spectrum of photonic crystals [28, 35, 79], computing the ultrasonic resonance frequencies to detect the presence of gas hydrates [73], analyzing the elastic properties of crystals with the use of rapid measurements [74, 75, 88], or calculating acoustic vibrations [18, 48, 97]. In particular, stochastic convection-diffusion equations are used to describe simple cases of turbulent [31, 60, 77, 94] or subsurface flows [95, 101].

4.2 Finite element discretization

The eigenvalue problem (4.1) should be discretized, because the solution in the analytical form is not available for arbitrary geometries and parameters. As such, we apply the standard finite element method in order to obtain an approximation of the desired eigenpairs (λ, u). But before deriving the finite element approximation, we first establish certain assumptions about our random field $\kappa(\omega)$ for almost all $\omega \in \Omega$.

Assumption 1. $\kappa(\cdot;\omega) \in L^\infty(D)$;

Assumption 2. $0 < \kappa_{\min} \leq \kappa(x;\omega) \leq \kappa_{\max} < \infty$.

4.2.1 Weak formulation

The derivation of the finite element approximation of the convection-diffusion eigenvalue problem is similar to the elliptic problem described in Chapter 3. For that, we multiply

Equation (4.1) by a test function $v \in H_0^1$

$$\int_D \mathbf{a}(x;\omega)\cdot\nabla u(x)v(x)\,\mathrm{d}x - \int_D \nabla\cdot\kappa(x;\omega)\nabla u(x)v(x)\,\mathrm{d}x = \lambda(\omega)\int_D u(x)v(x)\,\mathrm{d}x. \quad (4.4)$$

To decrease the derivative order of the solution $u(x)$, we perform integration by parts, noting that we have no Neumann boundary condition term since $u(x) = 0$ on Γ, so we obtain

$$\int_D \mathbf{a}(x;\omega)\cdot\nabla u(x)v(x)\,\mathrm{d}x + \int_D \kappa(x;\omega)\nabla u(x)\cdot\nabla v(x)\,\mathrm{d}x = \lambda(\omega)\int_D u(x)v(x)\,\mathrm{d}x. \quad (4.5)$$

We rewrite the variational form in a more convenient way: find a non-trivial eigenpair $(\lambda(\omega), u(\omega)) \in \mathbb{C}\times H_0^1$ with $\mathcal{B}(u,u) = 1$ such that

$$\mathcal{A}(u(\omega),v;\omega) + \mathcal{C}(u(\omega),v;\omega) = \lambda(\omega)\mathcal{B}(u(\omega),v;\omega) \quad \forall v \in H_0^1 \quad (4.6)$$

where

$$\mathcal{A}(u(\omega),v;\omega) := \int_D \kappa(x;\omega)\nabla u(x)\cdot\nabla v(x)\,\mathrm{d}x, \quad (4.7)$$

$$\mathcal{C}(u(\omega),v;\omega) := \int_D \mathbf{a}(x;\omega)\cdot\nabla u(x)v(x)\,\mathrm{d}x, \quad (4.8)$$

$$\mathcal{B}(u(\omega),v;\omega) := \int_D u(x)v(x)\,\mathrm{d}x. \quad (4.9)$$

We consider in addition to the primal form its dual eigenvalue problem for the future error analysis: find a non-trivial dual eigenpair $(\lambda^*(\omega), u^*(\omega)) \in \mathbb{C}\times H_g^1$ with $\mathcal{B}(u^*(\omega),u^*(\omega)) = 1$ such that

$$\mathcal{A}(w,u^*(\omega);\omega) + \mathcal{C}(w,u^*(\omega);\omega) = \overline{\lambda^*}(\omega)\mathcal{B}(w(\omega),u^*(\omega);\omega) \quad \forall w \in H_0^1. \quad (4.10)$$

The primal and dual eigenvalues are related to each other via $\lambda(\omega) = \overline{\lambda^*}(\omega)$.

The variational eigenvalue problem (4.6) admits a countable set of eigenvalues which can be represented as a sequence of their absolute values

$$\lambda_1(\omega) < |\lambda_2(\omega)| \le |\lambda_3(\omega)| \le \dots. \quad (4.11)$$

Theorem 5 (Krein-Rutman Theorem)**.** *If the linear operator (4.1) is compact then the smallest eigenvalue is real and simple (its algebraic multiplicity $m = 1$). Moreover, there exists a constant $\rho(\omega) > 0$, such that*

$$|\lambda_2(\omega) - \lambda_1(\omega)| \ge \rho(\omega), \quad (4.12)$$

meaning that the distance between the two smallest eigenvalues of the convection-diffusion

operator is constant.

Proof. See [36, 47]. □

An example is presented in Figure 4.1 using the finite element method with triangular elements for different mesh sizes. As Figure 4.1 shows, the ratio of the two smallest eigenvalues converges to some constant ρ.

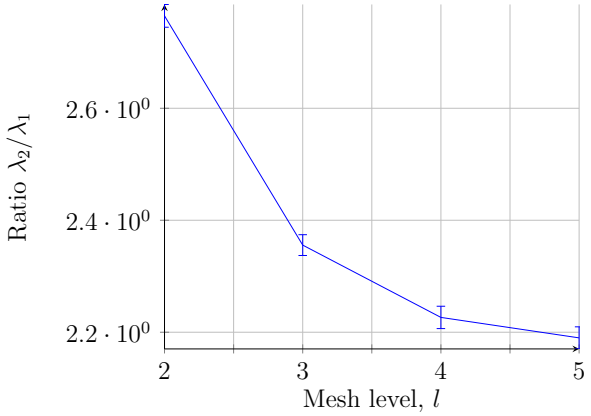


Figure 4.1 – Ratio between the two smallest eigenvalues using the finite element approximation on the unit domain for mesh sizes from $h = 2^{-2}$ to $h = 2^{-7}$. The convection-diffusion problem is with $\mathbf{a} = [50, 0]^T$ and $\kappa = 1$.

4.2.2 Finite element matrices

As usual, we approximate the infinite-dimensional spaces H_g^1 and H_0^1 by finite-dimensional subspaces V_g^h and V_0^h, respectively (Chapter 3). Then, the resulting discrete variational problem is to find non-trivial primal and dual eigenpairs $(\lambda(\omega), u_h(\omega)) \in \mathbb{C} \times V_0^h$ and $(\lambda^*(\omega), u_h^*(\omega)) \in \mathbb{C} \times V_0^h$ such that

$$\mathcal{A}(u_h(\omega), v_h(\omega)) + \mathcal{C}(u_h(\omega), v_h(\omega)) = \lambda_h(\omega)\mathcal{B}(u_h(\omega), v_h(\omega)) \quad \forall v_h \in V^h, \tag{4.13}$$

and

$$\mathcal{A}(w_h, u_h^*(\omega); \omega) + \mathcal{C}(w_h, u_h^*(\omega); \omega) = \overline{\lambda}_h^*(\omega)\mathcal{B}(w_h, u_h^*(\omega); \omega) \quad \forall w_h \in V^h. \tag{4.14}$$

The domain D is also discretized into elements D_k. Recall that the right and left eigenfunctions u_h^*, u_h and the test functions v and w can be represented as linear combinations in the finite spaces V_g^h and V_0^h,

$$\begin{aligned} u_h(x) &= \sum_{j=1}^{n} q_j \psi_j(x), \qquad u_h^*(x) = \sum_{j=1} q_j^* \psi_j(x), \\ v_h(x) &= \sum_{i=1}^{n} q_i^v \psi_i(x), \qquad w_h(x) = \sum_{i=1}^{n} q_i^{*,w} \psi_i(x). \end{aligned} \tag{4.15}$$

Then the discrete primal and dual formulations for the right and the left eigenfunctions can be written in matrix form

$$\mathbf{A}(\omega)\mathbf{q} = \lambda(\omega)\mathbf{M}\mathbf{q}, \qquad \mathbf{q}^H\mathbf{A}(\omega) = \overline{\lambda}(\omega)\mathbf{q}^H\mathbf{M}, \tag{4.16}$$

where $\mathbf{A}^H(\omega)$ denotes the transpose conjugate of the matrix $\mathbf{A}(\omega)$. The matrix elements are defined as

$$A_{ij}(\omega) = \sum_k \int_{D_k} \kappa(x;\omega)\nabla\psi_j(x)\nabla\psi_i(x)\,\mathrm{d}x + \sum_k \int_{D_k} \mathbf{a}(x;\omega)\cdot\psi_i(x)\nabla\psi_j(x)\,\mathrm{d}x, \tag{4.17}$$

and

$$M_{ij} = \sum_k \int_{D_k} \psi_j(x)\psi_i(x)\,\mathrm{d}x, \quad i,j = \overline{1\ldots n}. \tag{4.18}$$

The homogeneous Dirichlet boundary condition allows us to reduce the size of the resulting matrices by $n - n_0$ and to reduce the computational cost of iterative eigenvalue solvers by simply eliminating the n_0 equations from the linear system corresponding to the boundary conditions.

4.2.3 Finite element approximation error

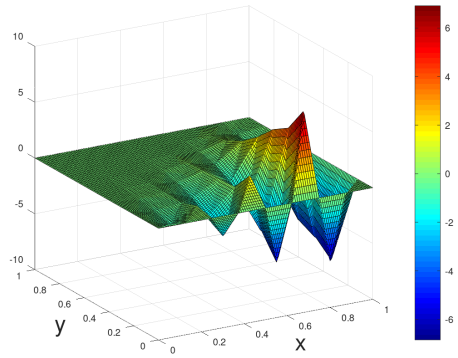
Unlike the case of the elliptic PDE (3.1), the finite element method for the convection-diffusion problem has stability issues in the convection-dominated regions if the element size h does not capture all necessary information about the flow. The Peclet number (sometimes called the mesh Peclet number) [103]

$$\mathbf{Pe}(x;\omega) = \frac{|\mathbf{a}(\mathbf{x};\omega)|h}{2\kappa(x;\omega)} \tag{4.19}$$

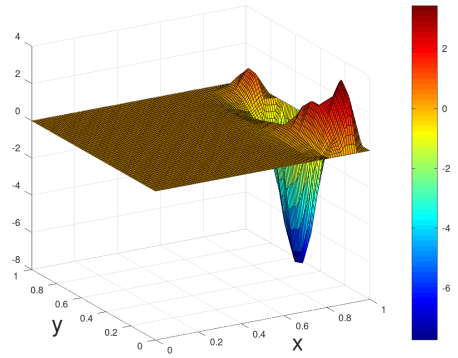
shows how small the mesh size h should be in order to have a stable solution using the Galerkin approximation.

To illustrate this we consider the following example: let $\mathbf{a}(x;\omega) = [50;0]^T$ and $\kappa(x;\omega) = 1$ with boundary condition $u(x;\omega)|_{\partial D} = 0$. Then, we solve the convection-diffusion problem using the finite element method with linear basis functions on triangular elements. Figure 4.2 shows the finite element solutions u_h obtained using different mesh sizes where h indicates the mesh size in both directions ($h = h_x = h_y$). Coarse grid solutions (Figures 4.2a and 4.2b) produce non-physical oscillations as a result of having large Peclet numbers compared to the solutions obtained using finer grids (Figures 4.2c and 4.2d). This instability is related to the smallest eigenvalue associated with the eigenvalue problem $\mathbf{A}$ (Eq. 4.16) itself as it is not simple and real but complex. As it can be seen from Figure 4.3 for an example with high velocity $\mathbf{a} = [100,0]^T$ and a single realization of random field $\kappa(x)$, the smallest eigenvalue λ_h is complex for a coarse grid discretization with mesh size $h = 2^{-3}$ (Figure 4.3a). At the same time, the smallest eigenvalue obtained

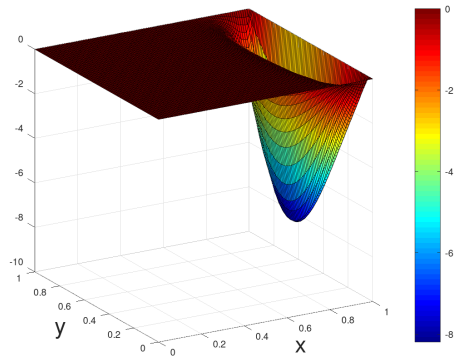
on a finer mesh is real and simple (Figure 4.3b). An analysis for 1D convection-diffusion problem is given in [68, 84].



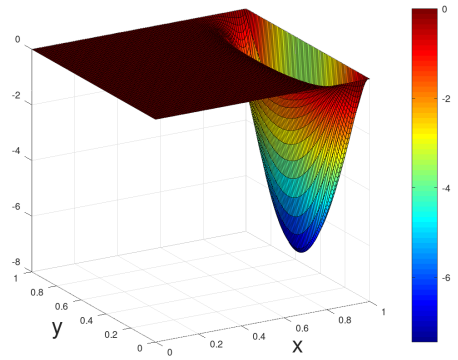
(a) $h = 2^{-3}, |\mathbf{Pe}| = 3.125$.



(b) $h = 2^{-4}, |\mathbf{Pe}| = 1.5625$.

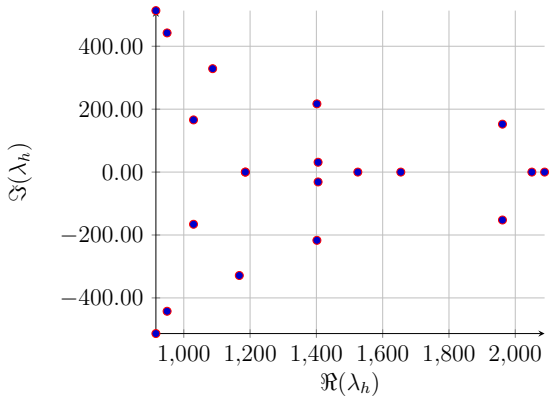


(c) $h = 2^{-5}, |\mathbf{Pe}| = 0.78125$.

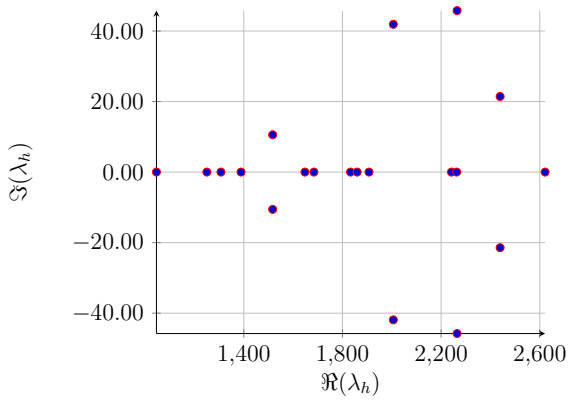


(d) $h = 2^{-6}, |\mathbf{Pe}| = 0.390625$.

Figure 4.2 – FE eigenfunction approximations u_h with homogeneous boundary conditions for different mesh sizes, $\mathbf{a}(x) = [50, 0]^T$ and $\kappa(x) = 1$ where **Pe** is the Peclet number. (a) and (b): unstable solutions. (c) and (d): stable solutions.



(a) $h = 2^{-3}$.



(b) $h = 2^{-6}$.

Figure 4.3 – Eigenvalue spectrum (the first 20) of the convection-diffusion operator for a single realization of random field $\kappa(x)$ with $\mathbf{a} = [100, 0]^T$ approximated by the finite element method. *Left:* Unstable solution obtained on mesh size with $h = 2^{-3}$. *Right:* Stable solution obtained on mesh with $h = 2^{-6}$.

Similarly to the continuous problem (4.6), for each ω the discrete problem (4.13) has a finite set of eigenvalues which approximate the first $M_h = \dim V_h$ eigenvalues of (4.6) for sufficiently small h

$$\lambda_{h,1}(\omega), \lambda_{h,2}(\omega), \ldots, \lambda(\omega)_{h,M_h}.$$

Lemma 2. *Let $\lambda_j(\omega)$ $j \geq 1$ be an eigenvalue of the convection-diffusion operator (4.11) with algebraic multiplicity m. Since the finite element approximation converges in norm, m eigenvalues $\lambda_{h,j}(\omega), \ldots, \lambda_{h,m+j}(\omega)$ converge to λ and the error is*

$$\left| \lambda_j(\omega) - \frac{1}{m} \sum_{i=j}^{m+j} \lambda_{h,i}(\omega) \right| \leq C_\lambda(\omega) h^2, \tag{4.20}$$

provided the FE approximation satisfies the stability condition on the Peclet number, $|\mathbf{Pe}| \leq 1$. $C_\lambda(\omega)$ is a constant which depends only on the operator of the eigenvalue problem.

Proof. See [14]. □

By Lemma 2 and Krein-Rutman theorem 5 we have a simple relation for the smallest eigenvalue $\lambda_1(\omega)$ (4.11) taking $\lambda_h \equiv \lambda_{h,1}$ and $\lambda(\omega) \equiv \lambda_1(\omega)$ [53]

$$|\lambda(\omega) - \lambda_h(\omega)| \leq C(\omega) h^2, \tag{4.21}$$

which ensures the convergence of our finite element approximation in the context of the geometric multi-level Monte Carlo method. Figure 4.4 shows the eigenpath of the smallest eigenvalue obtained on the mesh sequence from the coarsest mesh with the size $h = 2^{-2}$ to the finest one with $h = 2^{-6}$ for the problem 4.1 with $\mathbf{a} = [50, 0]^T$ and $\kappa = 1$.

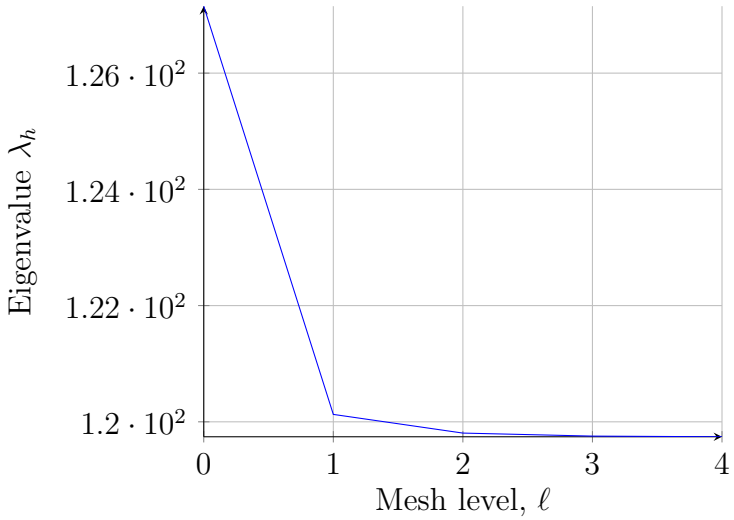


Figure 4.4 – Eigenpaths of the smallest eigenvalue of the unit domain with $\mathbf{a} = [20, 0]^T$ and $\kappa = 1$ obtained using the FE method with different mesh sizes $h_\ell = 2^{-2+\ell}$ starting with $h_0 = 2^{-2}$.

4.3 Eigenvalue problem in the matrix formulation

Our finite element approximation leads to a generalized eigenvalue problem in the matrix form

$$\mathbf{A}(\omega)\mathbf{q} = \lambda(\omega)\mathbf{M}\mathbf{q} \quad \text{and} \quad \mathbf{A}^H(\omega)\mathbf{q} = \lambda(\omega)\mathbf{M}^H\mathbf{q}, \tag{4.22}$$

where the mass matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is symmetric and positive-definite whereas the left-hand-side matrix $\mathbf{A}(\omega) \in \mathbb{R}^{n\times n}$ is nonsymmetric. We also note that the mass matrix $\mathbf{M}$ is deterministic and depends only on the mesh discretization and as such its assembly should be done only once for each grid discretization.

A variety of eigenvalue solvers can be applied to solve a generalized eigenvalue problem. This includes the simple power iteration, QR algorithm, subspace iterations, etc. A reasonable choice of an eigenvalue solver should be able to exploit the underlying properties of the eigenvalue problem. In our case, the eigenvalue problem is not only non-Hermitian but also is large and sparse. Also, we are interested only in computing an approximation of the smallest eigenvalue, instead of looking for the whole spectrum of the problem. For our purposes, we consider here only two eigenvalue solvers, the Rayleigh quotient iteration [24, 61] and implicitly restarted Arnoldi iteration [64], as they both able to compute the smallest eigenvalue of a nonsymmetric matrix.

4.4 Rayleigh quotient iteration

The Rayleigh quotient iteration is a simple but powerful method for finding eigenvalues and eigenvectors. Its ability to seek only one eigenvalue at a time and to work with any matrix structures gives its preference compared to other eigenvalue solvers. The Rayleigh quotient iteration greatly benefits from the sparsity of the matrices as it requires the solution of linear systems of equations at each iteration. The method itself was suggested by Lord Rayleigh in 1894 [83] for solving a quadratic eigenvalue problem of oscillations of a mechanical system. Various extensions were developed since then for more complicated cases such as for symmetric generalized eigenvalue problems by Crandall and Temple [24], nonsymmetric cases by Ostrowski [81], nonsymmetric generalized cases by Lancaster [61], and many others.

At its core, the method calculates the Rayleigh quotient of the estimate of the eigenvalue for two given vectors, and then uses it as the new shift of the shifted inverse power method at each iteration, whereas in the shifted inverse power method, the same shift is used at each iteration. The Rayleigh quotient for our generalized eigenvalue problem (4.22) is defined as

$$R(\mathbf{w}, \mathbf{r}) = \frac{\mathbf{r}^H\mathbf{A}(\omega)\mathbf{w}}{\mathbf{r}^H\mathbf{M}\mathbf{w}}, \tag{4.23}$$

where $\mathbf{r}$ and $\mathbf{w}$ are the left and right eigenvectors of our eigenvalue problem, respectively. Then, having initial approximations of the right and left eigenfunctions $\xi_{h,0}$, $\eta_{h,0}$ as well

as of the eigenvalue $\lambda_{h,0}$, an iterative process can be constructed through the use of the shifted inverse power method. A variant of such iteration is presented in Algorithm 2 which is a slightly modified version of the algorithm proposed in [61]. Note, as our eigenvalue of interest is real-valued, the iterative process does not involve any complex-valued arithmetic. The main computational cost comes from the inversion of the ill-conditioned real-valued matrices. Fortunately, the convergence rate of the Rayleigh quotient iteration is at least quadratic as stated in the following lemma.

Lemma 3. *[24, 81] Suppose we have an approximation $\lambda_{h,0}$ to the eigenvalue of interest $\lambda_h(\omega)$. Then if $|\lambda_{h,0} - \lambda_h(\omega)|$ is sufficiently small and defining $|\lambda_{h,i} - \lambda_h(\omega)| = \Delta(\omega)$, the Rayleigh quotient iteration sequence $\lambda_{h,i}$ $i = 0, 1, 2, \ldots$ converges to $\lambda_h(\omega)$ quadratically*

$$|\lambda_{h,i+1} - \lambda_h(\omega)| = O(\Delta^2). \tag{4.24}$$

Proof. See Crandall [24]. □

We have a similar lemma for a generalized eigenvalue problem.

Lemma 4. *Suppose we have approximations $\xi_{h,0}$ and $\eta_{h,0}$ to the left and right eigenvectors of interest $\xi_h(\omega)$ and $\eta_h(\omega)$. Then if $\|\xi_{h,0} - \xi_h(\omega)\|$ and $\|\eta_{h,0} - \eta_h(\omega)\|$ are both sufficiently small, then the sequences $\xi_{h,i}$ and $\eta_{h,i}$ $i = 0, 1, 2, \ldots$ of the Rayleigh quotient method converge cubically to the left and right eigenvectors $\xi_h(\omega)$ and $\eta_h(\omega)$ at least quadratically*

$$\begin{aligned} \|\xi_{h,i} - \xi_h(\omega)\| &= O(\Delta_\xi^2), \\ \|\eta_{h,i} - \eta_h(\omega)\| &= O(\Delta_\eta^2), \end{aligned} \tag{4.25}$$

and the sequence of estimated eigenvalues $\lambda_{h,i}$ converges with the same quadratic rate

$$|\lambda_{h,i+1} - \lambda_h(\omega)| = O(\Delta_\lambda^2). \tag{4.26}$$

Proof. See Lancaster [61]. □

In the Rayleigh quotient iteration the approximate eigenvalue pushes the solution of the linear system into its null space ($\lambda_i \mathbf{M} - \mathbf{A} \to 0$ as $i \to \infty$, see line 5 in Algorithm 2). The matrix $\lambda_i \mathbf{M} - \mathbf{A}$ is getting more and more ill-conditioned. As a consequence, the use of a direct solver may be preferable to the use of an iterative solver as the iterative solver may lead to numerical instability and to arithmetic overflow. One of these two criteria can be used to stop the iteration

$$\|\mathbf{A}\xi_i - \lambda_i \mathbf{M}\xi_i\| \leq \varepsilon_{rq}, \tag{4.27}$$

and

$$|\lambda_i - \lambda_{i-1}| \leq \varepsilon_{rq}. \tag{4.28}$$

Algorithm 2 Rayleigh Quotient Algorithm [61]

Given initial parameters ξ_0, η_0, λ_0, ε_{rq}, maxiter
while $\|\mathbf{A}\xi_i - \lambda_i \mathbf{M}\xi_i\| > \varepsilon_{rq}$ and $i \leq$ maxiter **do**
  Normalize $\xi_i \leftarrow \xi_i \|\xi_i\|_2^{-1}$
  Normalize $\eta_i \leftarrow \eta_i \|\eta_i\|_2^{-1}$
  Solve $[\lambda_i M - A]\xi_{i+1} = \xi_i$
  Solve $[\lambda_i M - A]^H \eta_{i+1} = \eta_i$
  Compute $\lambda_i \leftarrow \eta_i^H A \xi_i [\eta_i^H M \xi_i]^{-1}$
  $i := i + 1$
end while

Tables 4.1 and 4.2 show the results for the convergence of the Rayleigh quotient iteration for the discretized convection-diffusion operator with the velocity $\mathbf{a} = [20, 0]^T$ and the conductivity $\kappa(x)$ as a log-uniform random field (Figure 3.2). The results in Table 4.1 show that the practical convergence rate of the Rayleigh quotient iteration is at least quadratic as established in Lemma 3 and 4. Table 4.2 indicates that the use of the Rayleigh quotient algorithm with an initial guess obtained from the final output of the Rayleigh quotient iteration on a coarse mesh takes at least one iteration less compared to the use of a random initial vector instead. Although this saving in the computational time is insignificant in terms of the computational complexity, this still can be applied in the context of the multi-level Monte Carlo method to accelerate the computations. Indeed, we need to calculate the difference between approximate eigenvalues obtained from two different mesh discretizations at each level for the same Monte Carlo sample.

This difference in the number of iterations as well as in the total computational time becomes more clear when we consider a problem with a higher velocity and on a finer mesh. Figure 4.5 shows an example with velocity $\mathbf{a} = [50; 0]^T$ and $\kappa = 1$. The computational time for the Rayleigh quotient iteration with a random initial guess is 130 s while the total time for the same problem but with the final output from a coarser mesh used as the initial guess is 31 s which includes the computational times from both coarse and fine meshes.

Table 4.1 – Rayleigh quotient iteration for one sample for the problem (4.1) with $\mathbf{a} = [20; 0]^T$, $h = 2^{-5}$, random initial vectors, $\lambda_0 = 100$.

Iteration	$\lvert\lambda_h - \lambda_{h,i}\rvert$	$\|\mathbf{A}\xi_i - \lambda_{h,i}\mathbf{M}\xi_i\|$
0	2.683e+01	7.685e+00
1	2.575e-02	2.694e-03
2	7.915e-08	2.600e-06
3	5.400e-13	7.994e-12

Table 4.2 – Rayleigh quotient iteration for one sample for the problem 4.1 with $\mathbf{a} = [20; 0]^T$, $h = 2^{-5}$ using previous solution obtained from mesh discretization $h = 2^{-4}$ as an initial guess.

Iteration	$\lvert\lambda_h - \lambda_{h,i}\rvert$	$\lVert\mathbf{A}\xi_i - \lambda_{h,i}\mathbf{M}\xi_i\rVert$
0	1.640e-01	2.399e-02
1	1.317e-10	5.300e-08
2	2.558e-13	2.018e-15

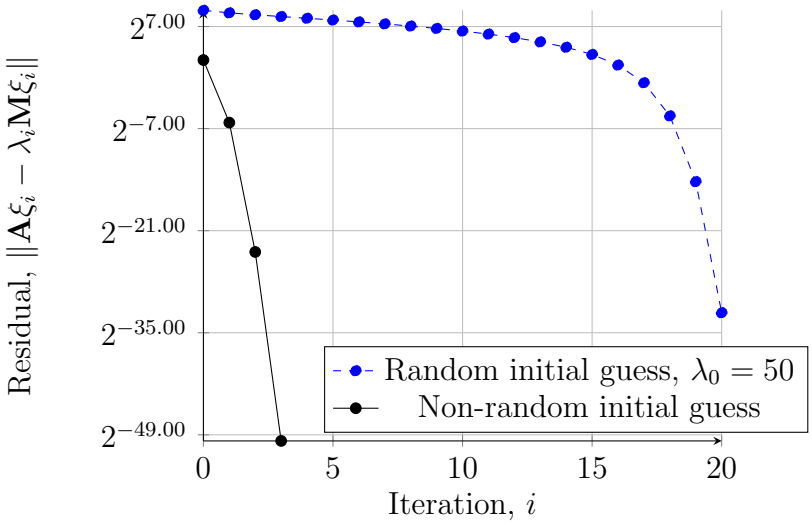


Figure 4.5 – Residuals of the Rayleigh quotient iteration for the convection-diffusion operator (4.1) with $\mathbf{a} = [50, 0], \kappa = 1$ on the unit domain using the FE method on a mesh with $h = 2^{-7}$. The initial guess in the second case (black line) was projected from the FE solution on the mesh size $h = 2^{-6}$.

4.5 Implicitly restarted Arnoldi method

For completeness we discuss the basic Arnoldi method first. The Arnoldi method efficiently computes a specified number of eigenpairs (u, λ) of large and sparse matrices with a given tolerance. The method was developed by Arnoldi in 1951 [3] to translate a matrix into its Hessenberg form. For more information, see [11, 64, 82, 85, 86, 89, 91, 92]. In the case of Hermitian matrices, the Arnoldi algorithm becomes Lanczos' iterative solver [62].

First, we consider a generalized eigenvalue problem and convert it to standard form

$$\mathbf{Au} = \lambda\mathbf{Mu} \Leftrightarrow \mathbf{Su} = \lambda\mathbf{u}, \tag{4.29}$$

where $\mathbf{M}$ is symmetric positive semi-definite and $\mathbf{S} = \mathbf{M}^{-1}\mathbf{A}$.

The method constructs an orthogonal basis $\mathbf{V}_m$ of the Krylov subspace produced by Arnoldi's iteration

$$\mathcal{K}_k(\mathbf{S}, \mathbf{v}_1) = \text{span}\{\mathbf{v}_1, \mathbf{Sv}_1, \mathbf{S}^2\mathbf{v}_1, \ldots, \mathbf{S}^{k-1}\mathbf{v}_1\}, \tag{4.30}$$

where $\mathbf{v}_1$ is an initial vector. For that, we impose a Galerkin condition

$$(\mathbf{w}, \mathbf{S}\mathbf{u} - \mathbf{u}\theta) = 0, \quad \forall \mathbf{w} \in \mathcal{K}_k(\mathbf{S}, \mathbf{v}_1). \tag{4.31}$$

If this condition is satisfied then the vector $\mathbf{u} \in \mathcal{K}_k(\mathbf{S}, \mathbf{v}_1)$ is called a Ritz vector and the value θ is called a Ritz value. Then, the Arnoldi factorization is computed after k steps

$$\mathbf{S}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T, \tag{4.32}$$

where $\mathbf{V}_k^T\mathbf{V}_k = \mathbf{I}_k$, the matrix $\mathbf{H}_k \in \mathbb{R}^{k\times k}$ is an upper Hessenberg matrix, and the vector $\mathbf{f}_k = \tilde{p}(\mathbf{S})\mathbf{v}_1$ is the residual with $\tilde{p}(\mathbf{S})$ as a polynomial of degree not exceeding $k-1$. The matrix $\mathbf{H}_k = \mathbf{V}_k^T\mathbf{S}\mathbf{V}_k$ forms the orthogonal projection of $\mathbf{S}$ onto the Range of $\mathbf{V}_k$. The orthonormal basis $\mathbf{V}_k$ of the Krylov subspace $\mathcal{K}_k$ can be obtained by various procedures, one such is shown in Algorithm 3. After that, the approximate eigenvalues and eigenvectors can be obtained from this factorization. The Arnoldi factorization can also be represented in an alternative way

$$\mathbf{S}\mathbf{V}_k = (\mathbf{V}_k, \mathbf{v}_{k+1}) \begin{pmatrix} \mathbf{H}_k \\ \beta_k\mathbf{e}_k^T \end{pmatrix}, \tag{4.33}$$

where $\beta_k = \|\mathbf{f}_k\|$ and $\mathbf{v}_{k+1} = \frac{1}{\beta_k}\mathbf{f}_k$, then the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ form an orthonomal basis $\mathbf{V}_k$ of the Krylov subspace $\mathcal{K}_k$ (4.30).

If $\mathbf{H}_k\mathbf{s} = \mathbf{s}\theta$ then the vector $\mathbf{u} = \mathbf{V}_k\mathbf{s}$ satisfies the following relation

$$\|\mathbf{S}\mathbf{u} - \mathbf{u}\theta\| = \|(\mathbf{S}\mathbf{V}_k - \mathbf{V}_k\mathbf{H}_k)\mathbf{s}\| = |\beta_k\mathbf{e}_k^T\mathbf{s}|, \tag{4.34}$$

where $|\beta_k\mathbf{e}_k^T\mathbf{s}|$ is called the Ritz estimate for the Ritz pair $(\mathbf{u}, \theta)$. Obviously, the smaller this estimate the better the approximations of the desired eigenvalues are. The use of the Ritz estimate allows us to extract the information about the numerical accuracy of an approximate eigenpair without explicitly computing the residual $\|\mathbf{S}\mathbf{u} - \mathbf{u}\theta\|$.

Algorithm 3 The k-step Arnoldi Factorization [64]

1: Input: $(\mathbf{S}, \mathbf{v}_1)$
2: $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}_1\|$, $\mathbf{w} = \mathbf{S}\mathbf{v}_1$, $\alpha_1 = \mathbf{v}_1^H\mathbf{w}$;
3: $\mathbf{f}_1 \leftarrow \mathbf{w} - \alpha_1\mathbf{v}_1$; $\mathbf{V}_0 \leftarrow (\mathbf{v}_1)$, $\mathbf{H}_1 \leftarrow (\alpha_1)$;
4: **for** $j = 1, 2, 3, \ldots, k-1$ **do**
5: $\quad\beta_j = \|\mathbf{f}_j\|$, $\mathbf{v}_{j+1} \leftarrow \mathbf{f}_j/\beta_j$;
6: $\quad\mathbf{V}_{j+1} \leftarrow (\mathbf{V}_j, \mathbf{v}_{j+1})$, $\mathbf{H}_j \leftarrow \begin{pmatrix} \mathbf{H}_j \\ \beta_j\mathbf{e}_j^T \end{pmatrix}$;
7: $\quad\mathbf{w} \leftarrow \mathbf{A}\mathbf{v}_{j+1}$;
8: $\quad\mathbf{h} \leftarrow \mathbf{V}_{j+1}^H\mathbf{w}$, $\mathbf{f}_{j+1} \leftarrow \mathbf{w} - \mathbf{V}_{j+1}\mathbf{h}$;
9: $\quad\mathbf{H}_{j+1} \leftarrow (\mathbf{H}_j, \mathbf{h})$;
10: **end for**

In finite precision arithmetic, the computed columns $\mathbf{V}_j$ do not form the orthogonal basis exactly and as such, special techniques should be considered when dealing with it.

Another issue of the Arnoldi method is its memory cost as it requires to store all the basis vectors. In addition, the cost of finding eigenvalues of the Hessenberg matrix is $O(k^3)$ at the kth step. An alternative approach was proposed by Lehoucq [64] in 1995 which mitigates both the computation and storage problems. In this method the k-step Arnoldi factorization is restarted using the implicit QR scheme. An implicitly restarted Arnoldi method requires only matrix-vector products and the solution of the mass matrix at each iteration compared to the Rayleigh quotient iteration where at each iteration it is necessary to solve two nonsymmetric matrices.

In the implicitly restarted Arnoldi method, an Arnoldi factorization of length k is extended to a length m by additional p steps to obtain

$$\mathbf{S}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T, \tag{4.35}$$

then p shifted QR steps are applied implicitly on $\mathbf{H}_{k+p}$ to obtain

$$\mathbf{S}\mathbf{V}_m^+ = \mathbf{V}_m^+\mathbf{H}_m^+ + \mathbf{f}_m\mathbf{e}_m^T\mathbf{Q}, \tag{4.36}$$

where $\mathbf{V}_m^+ = \mathbf{V}_m\mathbf{Q}$, $\mathbf{H}_m^+ = \mathbf{Q}^H\mathbf{H}_m\mathbf{Q}$, and $\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2\ldots\mathbf{Q}_p$. The matrices Q_j are orthogonal and associated with the shifts μ_j each. Moreover, the first $k-1$ entries of the vector $\mathbf{e}_m^T\mathbf{Q}$ are zero. After that, the last p columns are discarded delivering a k-step Arnoldi factorization as all necessary information about the desired eigenvalues is contained in this k-step factorization:

$$\mathbf{S}\mathbf{V}_k^+ = \mathbf{V}_k^+\mathbf{H}_k^+ + \mathbf{f}_k^+\mathbf{e}_k^T, \tag{4.37}$$

with the resulting residual $\mathbf{f}_k^+ = \mathbf{V}_m^+\mathbf{e}_{k+1}\hat{\beta}_k + \mathbf{f}_m\sigma$.

The starting vector $\mathbf{v}_1$ is replaced by $(\mathbf{S}-\mu_j\mathbf{I})\mathbf{v}_1$ after applying an implicit shift μ_j

$$\mathbf{v}_1 \leftarrow \psi(\mathbf{S})\mathbf{v}_1 \quad \text{with} \quad \psi(\lambda) = \prod_{j=1}^{p}(\lambda-\mu_j). \tag{4.38}$$

Thus, the polynomial $\psi(\lambda)$ should filter unwanted information from the starting vector $\mathbf{v}_1$ by either damping of unwanted eigenvectors or amplifying the wanted eigenvectors. As such, various techniques exist for choosing these p shifts. One of the most successful schemes is the use of exact shifts in which the eigenvalues of the Hessenberg matrix $\mathbf{H}_m$ are divided into two disjoint sets. The spectrum of the Hessenberg matrix $\mathbf{H}_k$ will be the eigenvalues of interest, $\sigma(\mathbf{H}_k) = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$.

The implicitly restarted Arnoldi iteration stops when

$$\|\mathbf{f}_m\||\mathbf{e}_m^T s| \leq \max(\varepsilon_M\|\mathbf{H}_m\|, \varepsilon\cdot|\theta|), \tag{4.39}$$

indicating that the Ritz pair $(\hat{\mathbf{u}}, \theta)$ is converged where ε_M is machine precision.

Unfortunately, the convergence rate is difficult to determine for a generalized eigen-

Algorithm 4 The Implicitly Restarted Arnoldi Algorithm [64]

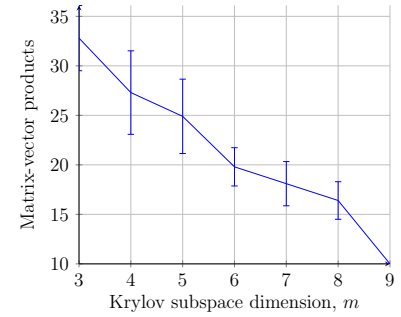
Input: $(\mathbf{S}, \mathbf{V}, \mathbf{H}, \mathbf{f})$ with an m-step Arnoldi factorization $\mathbf{S}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$
for $l = 1, 2, 3, \ldots$ until convergence **do**
  Compute $\sigma(\mathbf{H}_m)$ and select set of $m-k = p$ shifts $\mu_1, \mu_2, \ldots, \mu_p$ based upon $\sigma(\mathbf{H}_m)$ or other information;
  $\mathbf{q}^H \leftarrow \mathbf{e}_m^T$;
  for $j = 1, 2, \ldots, p$ apply implicitly a QR step: **do**
    Factor $[\mathbf{Q}, \mathbf{R}] = qr(\mathbf{H}_m - \mu_j\mathbf{I})$
    $\mathbf{H}_m \leftarrow \mathbf{Q}^H\mathbf{H}_m\mathbf{Q}$
    $\mathbf{V}_m \leftarrow \mathbf{V}_m Q$
    $\mathbf{q} \leftarrow \mathbf{q}^H\mathbf{Q}$
  end for
  $\mathbf{f}_k \leftarrow \mathbf{v}_{k+1}\tilde{\beta}_k + \mathbf{f}_m\sigma_k$
  $\mathbf{V}_k \leftarrow \mathbf{V}_m(1:n, 1:k)$
  $\mathbf{H}_k \leftarrow \mathbf{H}_m(1:k, 1:k)$
  Beginning with the k-step Arnoldi factorization $\mathbf{S}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$ apply p additional steps of the Arnoldi process to obtain a new m-step Arnoldi factorization $\mathbf{S}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$.
end for

value problem. Moreover, the rate depends not only on the structure of the eigenvalue problem but also on such parameters as the dimension of the Krylov subspace and the number of the eigenvalues of interest.

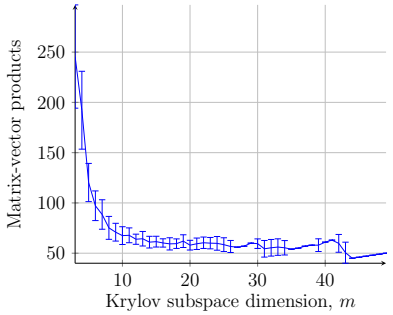
Figures 4.6-4.8 show the behaviour of the Arnoldi method for the discretized convection-diffusion operator (4.1) on the unit domain with the velocity $\mathbf{a} = [20, 0]^T$ and a random conductivity κ. We use ARPACK [65] as an implementation of the implicitly restarted Arnoldi method which is based on Algorithm 4. The numerical results were obtained on the grid sequence from $h = 2^{-2}$ to $h = 2^{-5}$ using 100 Monte Carlo samples. Figure 4.6 shows that the larger the Krylov subspace dimension, the less the number of matrix-vector products is required to find the smallest eigenvalue for the mesh discretization of $h = 2^{-2}$ and $h = 2^{-3}$. But for the finer grid discretizations, $h = 2^{-4}$ and $h = 2^{-5}$, the number of matrix-vector products is no longer monotonically decreasing as for the coarser meshes. Similarly, the number of Arnoldi iterations $\mathbf{S}\mathbf{v}$ which also includes the number of solving linear systems as $\mathbf{S}\mathbf{v} = \mathbf{M}^{-1}\mathbf{A}\mathbf{v}$ is decreasing with the increase of the Krylov subspace size as shown in Figure 4.7. As a result, the overall computational time has almost the same behaviour (Figure 4.8) as the plot of matrix-vector products.

4.6 Homotopy method

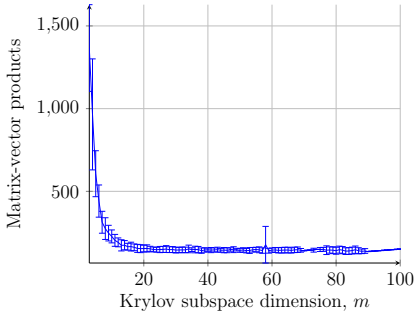
Many approaches exist to produce a solution without non-physical oscillations at a cheaper cost compared to the use of a very fine grid. For example, in adaptive finite element methods a mesh is refined only in the regions of high Peclet number and of the internal and boundary layers. Another technique is the use of a test space different from the trial



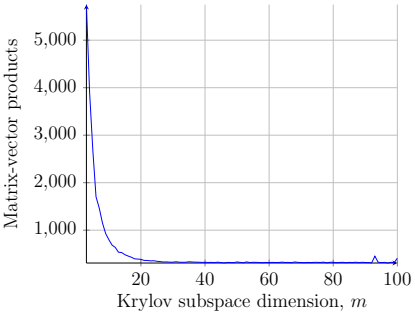
(a) $h = 2^{-2}$



(b) $h = 2^{-3}$

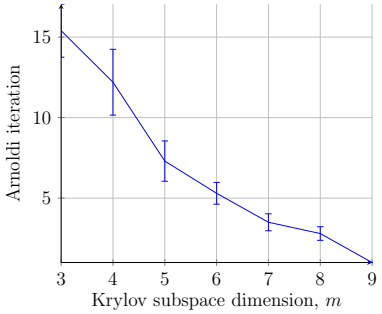


(c) $h = 2^{-4}$

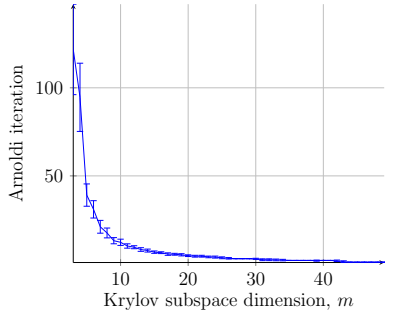


(d) $h = 2^{-5}$

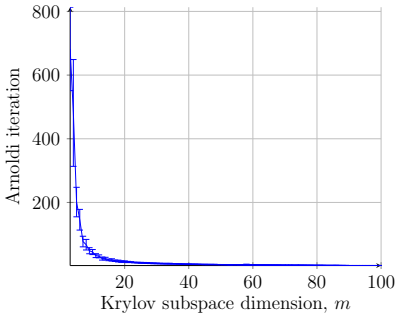
Figure 4.6 – Average number of matrix-vector products $\mathbf{Sv}$ of the implicitly restarted Arnoldi method as a function of Krylov subspace dimension m using the FE approximation for the convection-diffusion problem (4.1) with $\mathbf{a} = [20; 0]^T$ on the unit domain using 10^2 Monte Carlo samples.



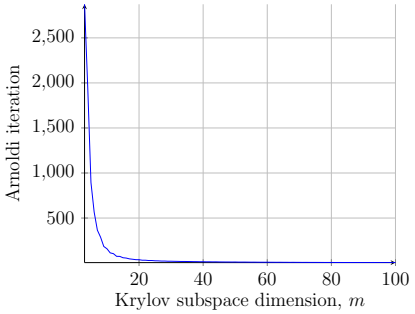
(a) $h = 2^{-2}$



(b) $h = 2^{-3}$

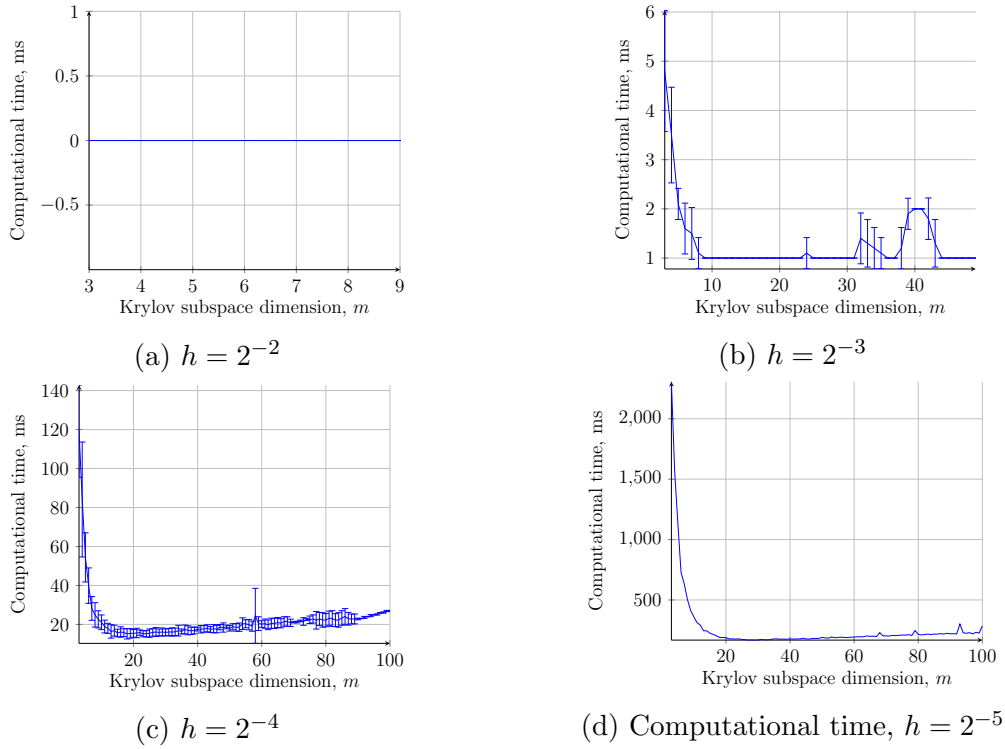


(c) $h = 2^{-4}$



(d) $h = 2^{-5}$

Figure 4.7 – Average number of Arnoldi iterations of the implicitly restarted Arnoldi method as a function of Krylov subspace dimension m using the FE approximation for the convection-diffusion problem (4.1) with $\mathbf{a} = [20; 0]^T$ on the unit domain using 10^2 Monte Carlo samples.



(a) $h = 2^{-2}$ (b) $h = 2^{-3}$

(c) $h = 2^{-4}$ (d) Computational time, $h = 2^{-5}$

Figure 4.8 – Average computational time in ms of the implicitly restarted Arnoldi method as a function of Krylov subspace dimension m using the FE approximation with $\mathbf{a} = [20; 0]^T$ on the unit domain using 10^2 Monte Carlo samples.

space such as in Petrov-Galerkin finite element formulations, e.g. the Galerkin/Least-squares method [30], the Streamline/Upwind Petrov-Galerkin method (SUPG) [16], etc. In the discontinuous Galerkin method [9, 27] non-physical oscillations are eliminated via discontinuous basis functions. Another approach is to decompose the solution into two components: a coarse-scale component and a fine-scale component, as in the variational multiscale method [54].

Here we employ a different methodology which is based on a homotopy continuation method. In [19] the homotopy method was adapted to the convection-diffusion eigenvalue problem with the use of adaptive finite element methods. The authors also provided the estimates on the convergence rate for the homotopy parameter to compute the smallest eigenvalue. As such, we aim to investigate its usefulness in the elimination of non-physical solutions in the context of the multi-level Monte Carlo method on coarse levels.

In general, the homotopy method is formed by the use of some initial operator $\mathcal{L}_0$ with spectrum close enough to the original operator $\mathcal{L}$ and then, a continuation is applied [71]

$$\mathcal{H}(t) = (1 - f(t))\mathcal{L}_0 + f(t)\mathcal{L} \quad \text{for} \quad 0 \leq t \leq 1, \tag{4.40}$$

with a function $f : [0; 1] \to [0; 1]$ and $f(0) = 0$, $f(1) = 1$. For our convection-diffusion operator (4.1) for $t = 0$ we have

$$\mathcal{H}(0) = \mathcal{L}_0 = -\nabla\kappa(x; \omega)\nabla u(x; \omega), \tag{4.41}$$

which represents the case of the elliptic problem (3.1) and

$$\mathcal{H}(1) = \mathcal{L} = \mathbf{a}(x;\omega)\cdot\nabla u(x;\omega) - \nabla\cdot\kappa(x;\omega)\nabla u(x;\omega), \tag{4.42}$$

is the operator of interest.

As for the function $f(t)$, we use a simple linear function $f(t) = t$, although other choices are possible, and it is also discretized at different values of t, so the homotopy (4.40 in this case becomes

$$\mathcal{H}(t) = -\nabla\cdot\kappa(x;\omega)\nabla u(x;\omega) + t_i\mathbf{a}(x;\omega)\cdot\nabla u(x;\omega) = \lambda(\omega,t)u(x;\omega) \text{ in } D. \tag{4.43}$$

The following lemma with proof is from [19] and aims to establish the error of the homotopy (4.40).

Lemma 5. *The homotopy error of the exact eigenvalues $\lambda(\omega, t=1)$ of the homotopy $\mathcal{H}(t)$ for problem (4.1) with divergence-free $\mathbf{a}$ a.s. is*

$$|\lambda(\omega,1) - \lambda(\omega,t)| \le (1-t)\|\mathbf{a}(x;\omega)\|_\infty(|u(\omega,t)|_{H^1} + |u^*(\omega,t)|_{H^1}) \quad \forall t\in[0;1], \tag{4.44}$$

where $|\cdot|_{H^1}$ is the Sobolev semi-norm and $u^*(\omega,t)$ is the dual solution.

Proof. Denote $u(\cdot) := u(\omega,\cdot)$ and $\lambda(\cdot) := \lambda(\omega,\cdot)$, then

$$\begin{aligned}
&(\lambda(1)-\lambda(t))(\mathcal{B}(u(1),u^*(1)) + \mathcal{B}(u(t),u^*(t)) - \mathcal{B}(u(1)-u(t),u^*(1)-u^*(t)))\\
=&(\lambda(1)-\lambda(t))\left(\mathcal{B}(u(1),u^*(t)) + \mathcal{B}(u(t),u^*(1))\right)\\
=&\lambda(1)\mathcal{B}(u(1),u^*(t)) + \overline{\lambda^*(1)}\mathcal{B}(u(t),u^*(1)) - \overline{\lambda^*(t)}\mathcal{B}(u(1),u^*(t)) - \lambda(t)\mathcal{B}(u(t),u^*(1))\\
=&(1-t)\mathcal{C}(u(1),u^*(t)) + (1-t)\mathcal{C}(u(t),u^*(1)).
\end{aligned}$$

Because velocity $\mathbf{a}(\omega)$ is divergence-free a.s. we have

$$\mathcal{C}(u(1),u^*(t)) = -\mathcal{C}(u^*(t),u(1)).$$

Using the Hölder inequality $\|fg\|_1 \le \|f\|_p\|g\|_q$ with $1/p+1/q=1$ we obtain the following result

$$\begin{aligned}
\mathcal{C}(u(t),u^*(1)) - \mathcal{C}(u^*(t),u(1)) &\le \|\mathbf{a}(\omega)\cdot\nabla u(t)\|\|u^*(1)\| + \|\mathbf{a}(\omega)\cdot\nabla u^*(t)\|\|u(1)\|\\
&\le \|\mathbf{a}(\omega)\|_\infty(|u(t)|_{H^1} + |u^*(t)|_{H^1}).
\end{aligned}$$

□

Figure 4.9 shows the eigenpath of the smallest eigenvalue for the homotopy parameter from 0 to 1 using triangular elements with mesh size $h = 2^{-7}$ for the problem with velocity $\mathbf{a} = [100,0]^T$ and the conductivity $\kappa = 1$ on the unit domain. As it can be seen, the eigenvalue changes exponentially with the homotopy parameter t. Therefore,

the homotopy step $\Delta t = t_\ell - t_{\ell-1}$ should become increasingly smaller towards the last level in a multi-level sequence, so that the multi-level difference becomes smaller as well. At the same time, the homotopy parameter t should satisfy the stability condition of the mesh Peclet number, $|\mathbf{Pe}| = t|\mathbf{a}|h/2\kappa < 1$.

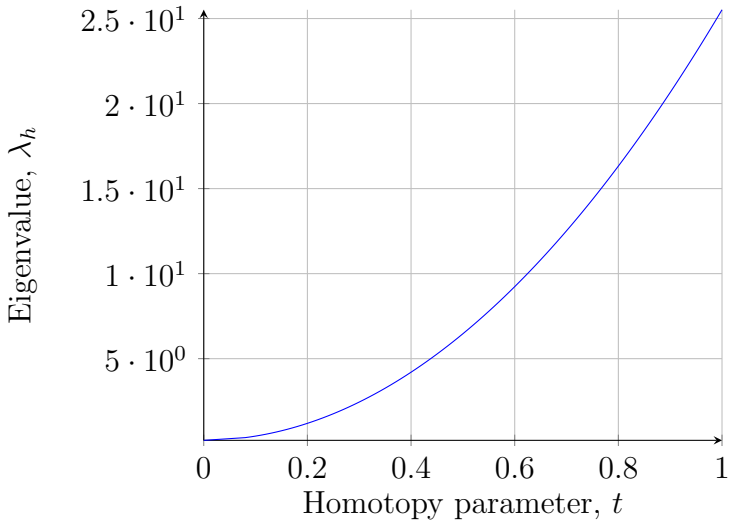


Figure 4.9 – Eigenpaths of the smallest eigenvalue for the homotopy parameter $t \in [0; 1]$ obtained using the finite element method with mesh size $h = 2^{-7}$ for $\mathbf{a} = [100, 0]^T$ and $\kappa = 1$ on the unit domain.

4.7 Homotopy multi-level Monte Carlo method

We explore several strategies of the multi-level Monte Carlo method related to the approximation errors and computational costs. The first error comes from the finite element discretization, ε_{FEM}. The second error and the computational cost are of the eigenvalue iterative solvers, ε_s, C_{RQ}, C_{IRA}. And the last approximation error is of the homotopy method, ε_t.

The simplest strategy is the usual geometric multi-level Monte Carlo using a finite element sequence $\{\Xi_h\}$ of (quasi)-uniform, shape-regular, conforming meshes on the spatial domain D, in which we double the grid resolution with level ℓ

$$\mathbb{E}[\lambda_L] = \mathbb{E}[\lambda_{h_0}] + \sum_{\ell=1}^{L} \mathbb{E}[\lambda_{h_\ell} - \lambda_{h_{\ell-1}}], \tag{4.45}$$

where λ_{h_0} is an approximate eigenvalue obtained on the coarse mesh with grid size $h = h_0$ using finite element discretization, λ_{h_ℓ} is an approximate eigenvalue obtained using mesh with size h_i. For simplicity, we write λ_{h_ℓ} as λ_ℓ omitting index h, $\lambda_\ell \equiv \lambda_{h_\ell}$.

Corollary 1 (Order fo convergence)**.** *For $\omega \in \Omega$, let $h > 0$ be sufficiently small then let $\lambda_\ell(\omega) := \lambda_{h/2}(\omega)$ and $\lambda_{\ell-1}(\omega) := \lambda_h(\omega)$ be the Galerkin (4.13) approximation to the*

convection-diffusion operator (4.1) of the smallest eigenvalue $\lambda(\omega)$. The expectation of their difference is bounded by

$$\left|\mathbb{E}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)]\right| \leq C_{4,1} 2^{-2\ell}, \tag{4.46}$$

while the variance is bounded by

$$\mathbb{V}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)] \leq C_{4,2} 2^{-4\ell}. \tag{4.47}$$

Here C_{41}, C_{42} are independent of ω, h and ℓ.

Proof. From Theorem 2 we have

$$\mathbb{E}[|\lambda(\omega) - \lambda_\ell(\omega)|] \leq C_{4,1} h^2,$$

therefore

$$\begin{aligned}
\left|\mathbb{E}[\lambda(\omega) - \lambda_{\ell-1}(\omega)]\right| &= \left|\mathbb{E}[\lambda_\ell(\omega) - \lambda(\omega) + \lambda(\omega) - \lambda_{\ell-1}(\omega)]\right| \\
&\leq \mathbb{E}[|\lambda(\omega) - \lambda_\ell(\omega)|] + \mathbb{E}[|\lambda(\omega) - \lambda_{\ell-1}(\omega)|] \\
&\leq \tilde{C}_{4,1}(2^{-2\ell} + 2^{-2(\ell-1)}) = C_{4,1} 2^{-2\ell}.
\end{aligned}$$

The variance reduction rate is

$$\mathbb{V}[\lambda_\ell(\omega) - \lambda_{\ell-1}(\omega)] \leq \mathbb{E}[\big(\lambda_\ell(\omega) - \lambda_\ell(\omega)\big)^2] \leq C_{4,1}^2 2^{-4\ell}.$$

□

The second strategy is to employ a different homotopy parameter at each mesh discretization level to satisfy the stability condition on the finite element approximation. The multi-level sequence in this case is $\{(t = 0, h = h_0), (t = t_1, h = h_1), \ldots, (t = 1, h = h_L)\}$, where (t_ℓ, h_ℓ) are model parameters for level ℓ and t is the homotopy parameter (4.40). The resulting MLMC estimator is

$$\mathbb{E}[\lambda_L] = \mathbb{E}[\lambda_{t_0,h_0}] + \sum_{i=1}^{L} \mathbb{E}[\lambda_{t_i,h_i} - \lambda_{t_{i-1},h_{i-1}}]. \tag{4.48}$$

Both of these methods are employed with the Rayleigh quotient and implicitly restarted Arnoldi eigenvalue solvers alongside the finite element method using triangular elements with linear basis functions.

4.8 Numerical results

In this section, we consider two numerical examples to investigate the properties of the multi-level Monte Carlo (MLMC) methods. In both cases, our quantity of interest is the smallest eigenvalue of the stochastic convection-diffusion operator (4.1) in the unit domain $D = [0; 1] \times [0; 1]$. The diffusion $\kappa(x; \omega)$ is a random variable modelled as a log-uniform random field constructed through the convolution of 25 i.i.d. uniform random variables

$$\log \kappa(x; \omega) = \sum_{i=1}^{25} \omega_i k(x - c_i), \tag{4.49}$$

with exponential smoothing kernels $k(x - c_i) = \exp[-\frac{25}{2}\|x - c_i\|]$ where c_i are the kernel centers placed uniformly on a 5×5 grid in the domain D. As in Chapter 3, the random seed is the same for all numerical experiments presented in this section. A realization of such field κ is shown in Figure 4.10.

For each example we apply four variants of the multi-level Monte Carlo method. The first strategy is the use of MLMC with the implicitly restarted Arnoldi method. Then we change the eigenvalue solver to the Rayleigh quotient iteration and compare both methods in terms of the computational cost. After that, we perform the same tests but utilizing the homotopy continuation method for each eigenvalue solver, to see if any improvements arise. Finally, we compare the efficiency between all these strategies including the classic Monte Carlo method.

ARPACK [65] is used as an implementation of the implicitly restarted Arnoldi method while the Rayleigh quotient method was developed in C++ with the use of the Eigen library [42] as the solver of linear systems for non-symmetric matrices. We use the UmfPackLU function which is included in the SuiteSparse library [26] to perform the LU decomposition with permutations for solving linear systems of equations. As a method of generating pseudo-random samples, we use the std::mt19937 function of the standard C++ library.

In Table 4.3 the matrix sizes $\tilde{n}$ are given for the mesh sequence, $h^{-1} = 2^2, 2^3, \ldots, 2^7$, resulting from the finite element discretization using triangular elements with linear basis functions. The matrix dimensions are the same for all numerical examples. The number of nodes $\tilde{n}$ increases approximately quadruply with level. Table 4.4 shows optimal Krylov subspace dimensions for the Arnoldi method for the matrices obtained from the finite element approximation. The parameters are chosen based on the performance for each grid discretization shown at the end of Section 4.5 (Figures 4.6-4.8).

4.8.1 Problem I

Let $\mathbf{a} = [20; 0]^T$. Then the mesh sequence starts from $h = 2^{-3}$ in order to satisfy the stability condition and ends at $h = 2^{-7}$. The stopping criteria in the Rayleigh quotient

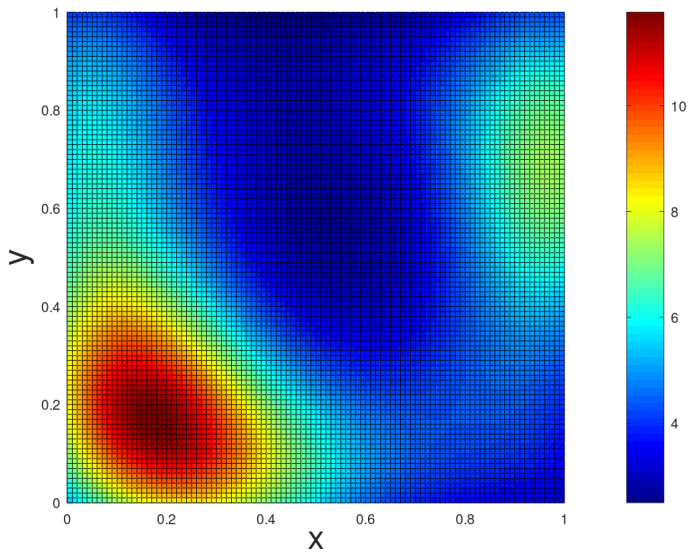


Figure 4.10 – A log-uniform random field $\kappa(x, \omega)$ for a single realization ω with 25 exponential kernels placed uniformly as a grid 5×5.

Table 4.3 – Degrees of freedom resulting from the finite element approximation for the mesh sequence used in the simulations.

Level, ℓ	h^{-1}	Number of nodes, $\tilde{n}$
0	2^3	49
1	2^4	225
2	2^5	961
3	2^6	3969
4	2^7	16129

Table 4.4 – Krylov subspace dimensions for the matrices used in the simulations coupled with the Arnoldi method.

Level, ℓ	h^{-1}	Krylov subspace size, m
0	2^3	20
1	2^4	40
2	2^5	70
3	2^6	70
4	2^7	100

iteration is until the residual $\|\mathbf{Au} - \lambda\mathbf{Mu}\| \leq 10^{-12}$. The implicitly restarted Arnoldi method stops when the Ritz vectors $\|\mathbf{f}_m\||\mathbf{e}_m^T s| \leq \max(10^{-12}\|\mathbf{H}_m\|, 10^{-12} \cdot |\theta|)$, where $\mathbf{H}_m$ is a Hessenberg matrix $\mathbf{H}_m \in \mathbb{R}^{m \times m}$ at step m, θ is the approximate eigenvalue.

Table 4.5 shows various output parameters such as the average number of matrix-vector products, of the number of iterations, and of the computational time, for the implicitly restarted Arnoldi method in the MLMC setting using 10^4 samples at each level ℓ. As can be seen, the average number of matrix-vector products as well as Arnoldi iterations increases with the matrix size. As is shown in Figure 4.11, the rate of increase of matrix-vector products (Figure 4.11c) is almost linear whereas the rate of increase of

number of Arnoldi iterations (Figure 4.11d) is harder to determine as it is about 0.4 from the level 0 to 2 and then the rate becomes linear. On the other hand, the rate of increase of the computational time itself (Figure 4.11e) is easier to understand and is asymptotically $\gamma \approx 3$ (see Table 4.6) indicating an $O(h^{-3})$ cost. From both Table 4.6 and Figure 4.11a we observe that the convergence rate of the mean α corresponds to the theoretical estimate (Lemma 2) which is approximately 2 (blue line). As a result, the variance reduction rate β (Figure 4.11b and Table 4.6) is about 4 which is approximately the square of the mean convergence rate. That way, we have that the variance reduction rate, $O(h^4)$, is larger than the cost increasing rate, $O(h^{-3})$. This corresponds to the best case scenario (Theorem 1 and Equation (2.15)) in which the use of the MLMC method is justified for finding the smallest eigenvalue of the convection-diffusion operator (4.1).

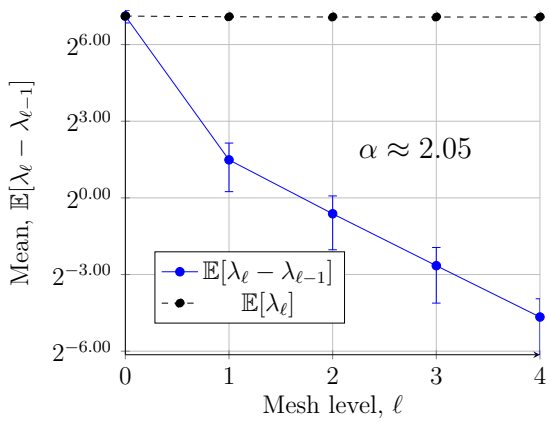
Table 4.5 – Average values of the Arnoldi method: matrix-vector products **Sv**, number of iterations, and computational time in ms using multi-level Monte Carlo simulations for 10^4 samples at each level ℓ for Problem I with $\mathbf{a} = [20; 0]^T$.

Level, ℓ	h^{-1}	**Sv**	Number of iterations	Computational time
0	2^3	101.6	9	2.0113 ms
1	2^4	255.8	11.7	28.865 ms
2	2^5	583.2	15.6	351.19 ms
3	2^6	1195.8	33.0	2 925 ms
4	2^7	2455.3	57.9	29 144 ms

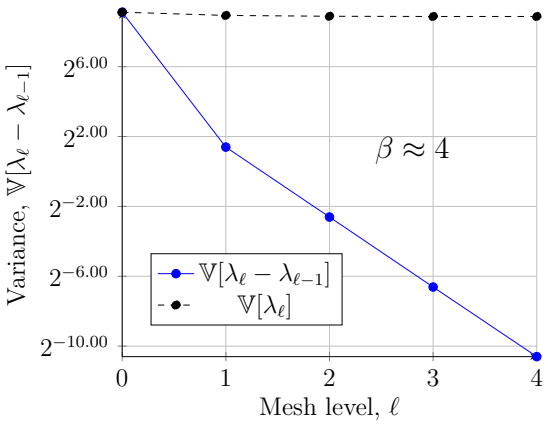
Table 4.6 – Multi-level Monte Carlo results using 10^4 samples on each level ℓ for Problem I using the implicitly restarted Arnoldi method with $\mathbf{a} = [20; 0]^T$, showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

Level, ℓ	h^{-1}	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^3	1.386e+02	5.561e+02	20 113 ms			
1	2^4	2.804e+00	2.632e+00	288 653 ms	5.63	7.72	3.84
2	2^5	6.498e-01	1.648e-01	3 511 929 ms	2.11	4.00	3.60
3	2^6	1.590e-01	1.030e-02	29 251 296 ms	2.03	4.00	3.05
4	2^7	3.952e-02	6.437e-04	291 448 142 ms	2.00	4.00	3.32

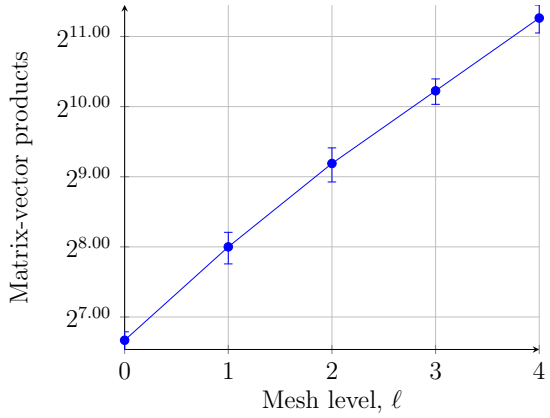
Similar results were obtained by using the Rayleigh quotient iteration as the eigenvalue solver as shown in Table 4.7. Obviously, the variance and mean decay rates, α and β, are the same as in the case of the implicitly restarted Arnoldi method while the cost increase rate is slightly different having on average $\gamma_{\text{RQ}} \approx 3$ compared to $\gamma_{\text{IRA}} \approx 3.5$ using the Arnoldi method. As in the previous case the use of MLMC is fully justified according to the Theorem 1. Although the average $\gamma_{\text{RQ}} < \gamma_{\text{IRA}}$ and the computational time of each level for the Rayleigh quotient iteration is lower except on the zeroth level $\ell = 0$, the use of the implicitly restarted Arnoldi method would be more preferable as the main contribution in the overall computational cost in both methods comes from the coarsest level $\ell = 0$ and the Arnoldi method is twice cheaper in that case as can be seen from Figure 4.15.



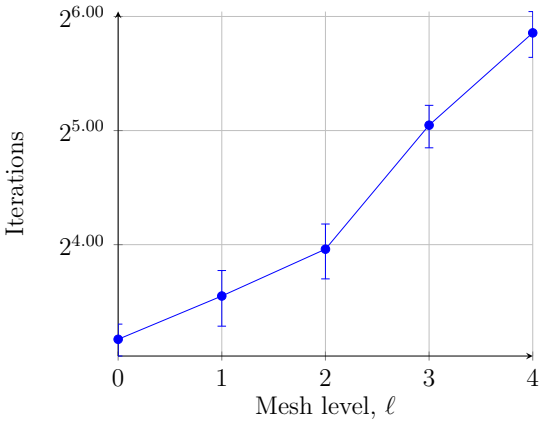
(a) Mean of difference.



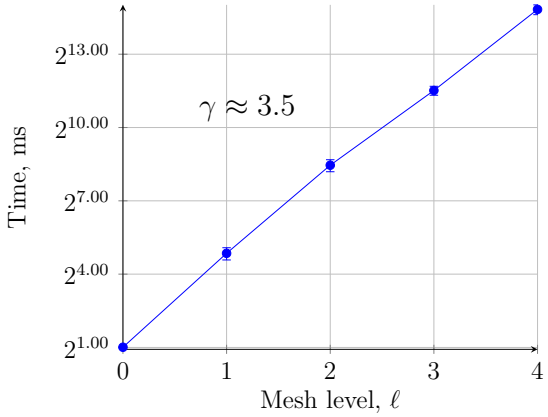
(b) Variance of difference.



(c) Matrix-vector products.



(d) Arnoldi iterations.



(e) Computational time.

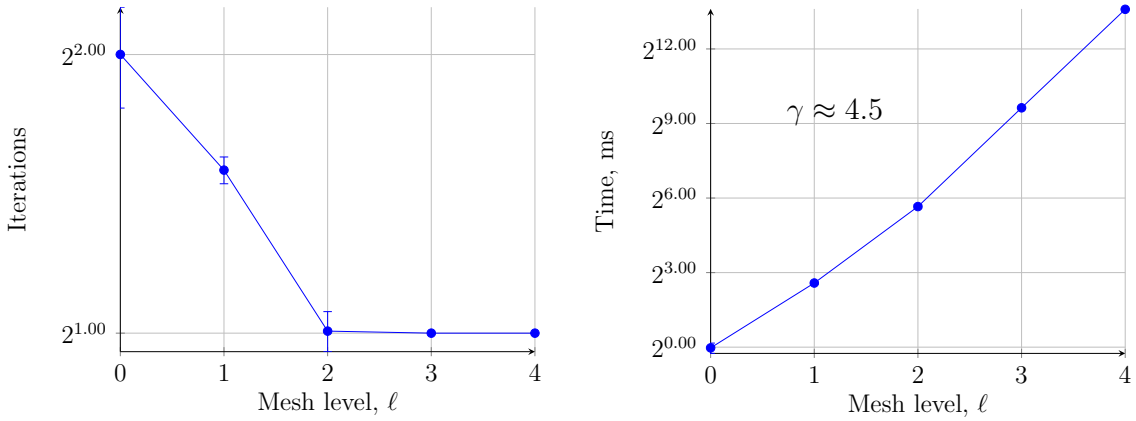
Figure 4.11 – MLMC using 10^4 samples at each level to find the smallest eigenvalue of Problem I with $\mathbf{a} = [20; 0]^T$ using the finite element approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Arnoldi method as the eigenvalue solver. (a) expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b) variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c) average number of matrix-vector products of computing the expectation of differences $\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]$. (d) average number of Arnoldi iterations of computing the expectation of differences. (e) average computational time of one sample.

Obviously, the combination of these two methods in the multi-level Monte Carlo setting would yield even a smaller computational cost but the improvement will be by a constant

only.

Table 4.7 – Multi-level Monte Carlo results using 10^4 samples on each level ℓ for Problem I with $\mathbf{a} = [20; 0]^T$ using the Rayleigh quotient method, showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

Level, ℓ	h^{-1}	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^3	1.386e+02	5.561e+02	44 460 ms			
1	2^4	2.804e+00	2.632e+00	274 194 ms	5.63	7.72	2.62
2	2^5	6.498e-01	1.648e-01	1 545 742 ms	2.11	4.00	2.50
3	2^6	1.590e-01	1.030e-02	13 894 066 ms	2.03	4.00	3.17
4	2^7	3.952e-02	6.437e-04	174 273 148 ms	2.00	4.00	3.64



(a) Rayleigh quotient iterations.

(b) Computational time.

Figure 4.12 – Multi-level Monte Carlo method using 10^4 samples at each level to find the smallest eigenvalue of Problem I with $\mathbf{a} = [20; 0]^T$ using the FE approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. *Left:* Average number of Rayleigh quotient iterations used to obtain the difference $\lambda_\ell - \lambda_{\ell-1}$ for one sample at each level ℓ. *Right:* Average computational time to solve the problem for one sample.

Table 4.8 shows the results of the homotopy multi-level Monte Carlo simulations with the Rayleigh quotient iteration. The homotopy sequence was generated by the formula $t_\ell = 1 - 1/4^\ell$ with the final homotopy parameter $t_{\ell=4} = 1$. That way the distance between two homotopy parameters $t_{\ell-1}$ and t_ℓ is becoming smaller towards the last level, so the convergence of the mean of differences $\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]$ and variances $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ can be ensured (Figure 4.13a and Figure 4.13b). The actual convergence rates of both the mean and variance experience a drop from the first level to the second one but after that the difference monotonically increases with the level, while the averages of their rates are $\alpha \approx 2.01$ for the mean of the difference and $\beta \approx 3.65$ for the variance of the difference. Compared to Figure 4.12a (the Rayleigh quotient iteration without the homotopy), the number of iterations across all levels except the last one is greater and the method also experiences greater variance. Because the initial guess on level ℓ comes from the final

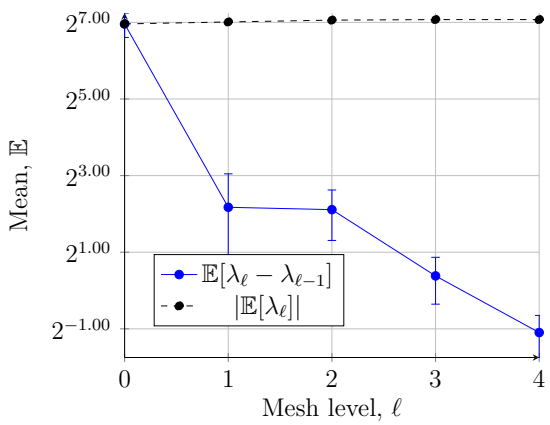
output of level $\ell - 1$ with a different homotopy parameter t, the amount of iterations required to solve the problem increases compared to the same setup but without the use of the homotopy. The computational cost increase rate is similar across all levels with $\gamma \approx 3$ (Figure 4.14b). Overall, the variance reduction rate is larger than the cost increase rate except between the first and the second levels and between the last two levels. This means that the main computational cost will be spent on the zeroth and second levels. But the actual computational complexity depends on the length of the mesh sequence: the more mesh levels, the lengthier the homotopy sequence is itself.

Table 4.8 – Homotopy multi-level Monte Carlo results using 10^4 samples on each level ℓ for Problem I with $\mathbf{a} = [20; 0]^T$ using the Rayleigh quotient method showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

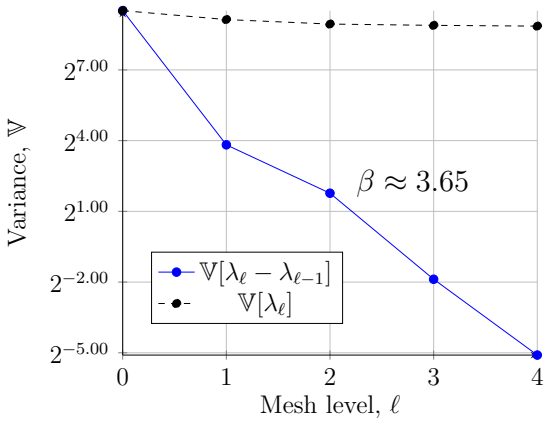
ℓ	h^{-1}	t	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^3	0	1.244e+02	7.310e+02	38 984 ms			
1	2^4	0.75	4.509e+00	1.413e+01	278 354 ms	4.78	5.69	2.84
2	2^5	0.9375	4.322e+00	3.409e+00	1 857 123 ms	0.06	2.05	2.73
3	2^6	0.984375	1.303e+00	2.718e-01	16 223 628 ms	1.25	3.64	3.12
4	2^7	1	4.678e-01	2.925e-02	205 043 525 ms	1.95	3.21	3.66

Next, the same homotopy sequence was used but now in pair with the implicitly restarted Arnoldi method (Figure 4.13) instead of the Rayleigh quotient iteration. As expected, the mean and variance exhibits the same inconsistency with both eigenvalue solvers. On the other hand, the computational cost rate is more stable with the average $\gamma \approx 3.12$ as can be seen from Figure 4.13e which is less than for the case without the use of the homotopy and with the same eigenvalue solver where $\gamma \approx 3.5$. Although the mean and variance reduction rates, Figure 4.13a and Figure 4.13b, respectively, are significantly different from the case without the homotopy method, the behavior of the plots of the number of matrix-vector (Figure 4.13c) products as well as of the number of Arnoldi iterations (Figure 4.13d) is similar to the case without the homotopy (Figures 4.11c and 4.11d).

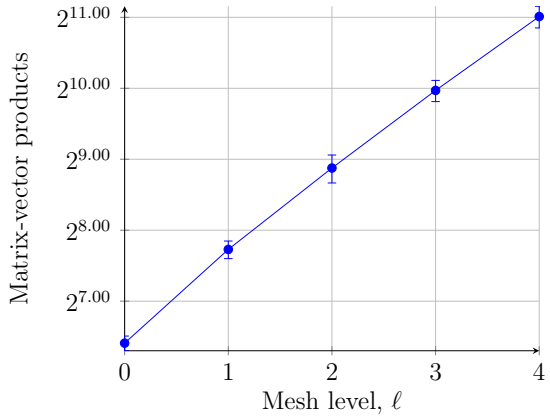
Finally, Figure 4.15 plots CPU time vs. mean square error for all presented MLMC methods. The lines plotted in blue and red indicate the use of the implicitly restarted Arnoldi method and the Rayleigh quotient iteration, respectively, while the lines plotted with squared marks indicate the use of the homotopy method and the lines plotted with triangular marks indicate no homotopy. Overall, MLMC with the Arnoldi method outperforms all other methods as shown in Figure 4.15. MLMC with the Rayleigh quotient iteration works just slightly worse. The computational cost of the homotopy MLMC method, on the other hand, grows worse because the variance of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ is far greater for the last two levels compared to the simple MLMC. Nevertheless, all presented methods give a computational advantage having an $O(\varepsilon^{-2})$ complexity in comparison to the standard Monte Carlo method which has an $O(\varepsilon^{-5}) = O(\varepsilon^{-3} \times \varepsilon^{-2})$ (cost of one sample $O(\varepsilon^{-3})$ multiplied by $O(\varepsilon^{-2})$ samples required to achieve the targeted error) complexity.



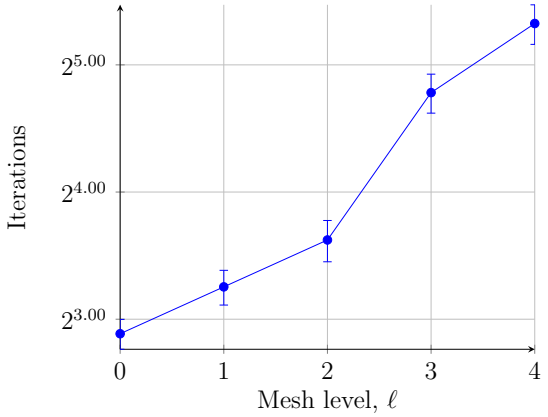
(a) Mean of difference.



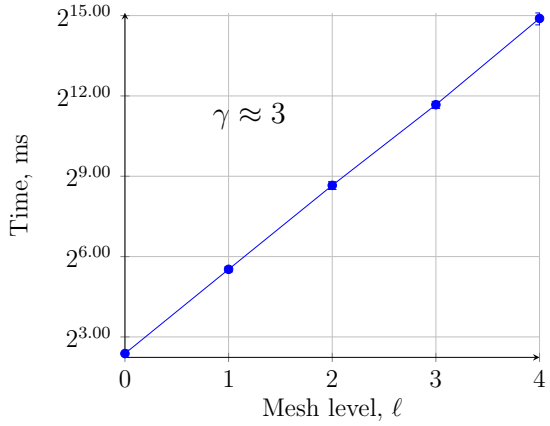
(b) Variance of difference.



(c) Matrix-vector products.

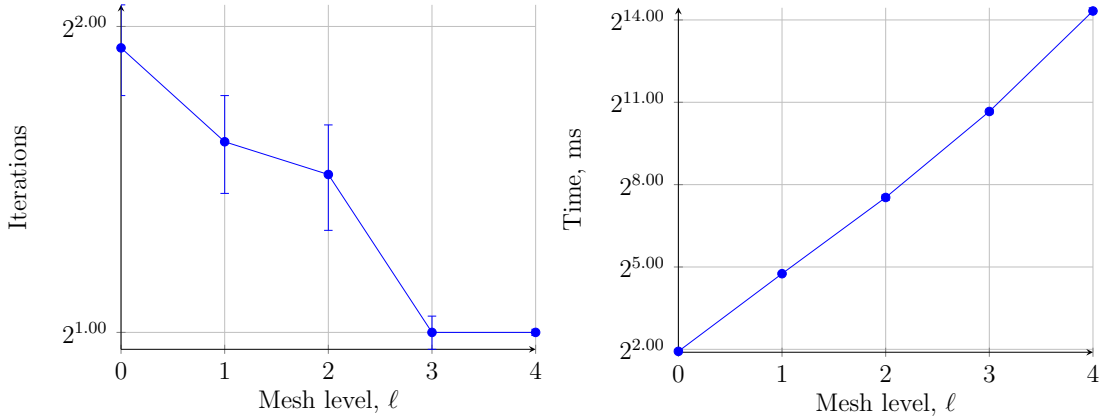


(d) Arnoldi iterations.



(e) Computational time.

Figure 4.13 – Homotopy multi-level Monte Carlo method using 10^4 samples at each level ℓ to find the smallest eigenvalue of Problem I with $\mathbf{a} = [20; 0]^T$ using the FE approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Arnoldi method as the eigenvalue solver. (a) expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b) variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c) average number of matrix-vector products of computing the expectation of differences $\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]$. (d) average number of Arnoldi iterations of computing the expectation of differences. (e) average computational time of one sample.



(a) Rayleigh quotient iterations.

(b) Computational time.

Figure 4.14 – Homotopy MLMC using 10^4 samples at each level ℓ to find the smallest eigenvalue of Problem I with $\mathbf{a} = [20; 0]^T$ using the FE approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. *Left:* Average number of Rayleigh quotient iterations used to obtain the difference $\lambda_\ell - \lambda_{\ell-1}$ for one sample at each level ℓ. *Right:* Average computational time to solve the problem for one sample.

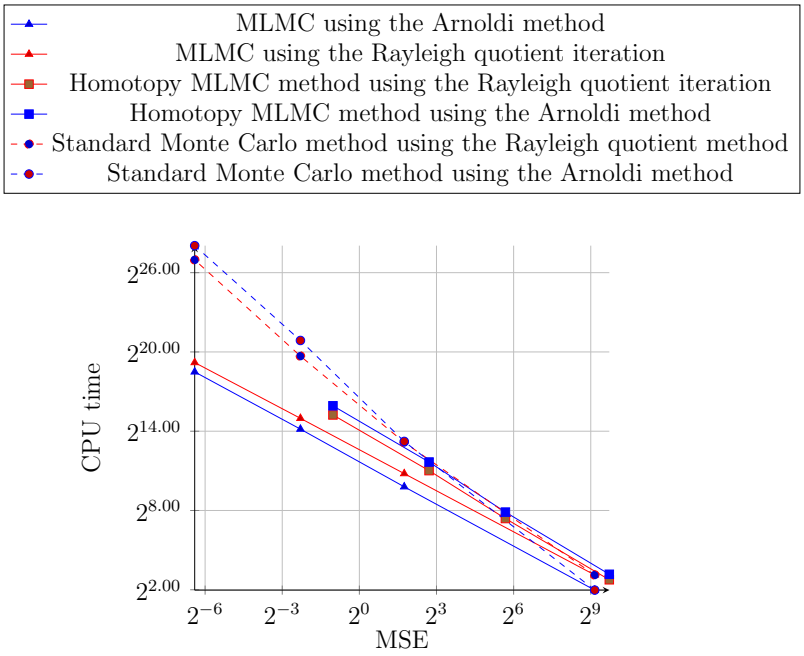


Figure 4.15 – CPU time vs. mean square error of MLMC for Problem I using two different eigenvalue solvers.

4.8.2 Problem II

Now we investigate the behaviour of the previously developed multi-level Monte Carlo methods in the context of high velocity. Setting the velocity $\mathbf{a} = [50; 0]^T$ will make the first two levels unusable as the solution on these levels will exhibit non-physical oscillations. Thus, the mesh sequence for the MLMC method begins with the mesh size $h = 2^{-5}$ and ends with $h = 2^{-7}$, in total using only three levels compared to the previous sequence with the total of five levels. In tests with the homotopy MLMC method we use only the Rayleigh quotient iteration as the eigenvalue solver because, as shown in the previous section, the Rayleigh quotient iteration performs slightly faster than the implicitly restarted Arnoldi method for all levels. As the condition number of the generated matrices becomes worse with a higher velocity, we increase the tolerance $\varepsilon_{rq} = 10^{-9}$, to reduce the number of Rayleigh quotient iterations needed to solve the eigenvalue problem.

Table 4.9 demonstrates the results for the MLMC method with 10^4 samples using the Arnoldi iteration as the eigenvalue solver. As in the case of the lower velocity, the variance reduction rate which is $\beta \approx 4$ is higher than the cost increase rate which is $\gamma \approx 3.2$. The difference between the mean and variance reduction rates on the first two levels (Figure 4.16) is even greater in comparison with the use of the same eigenvalue solver for Problem I (Table 4.6) which is $\alpha_I \approx 5.63$ vs. $\alpha_{II} \approx 9.91$ and $\beta_I \approx 7.72$ vs. $\beta_{II} \approx 13.7$. Therefore, the complexity of the method is $O(\varepsilon^{-2})$, although it becomes slower compared to the problems with lower velocity.

Table 4.9 – Multi-level Monte Carlo results using 10^4 samples on each level ℓ for Problem II with $\mathbf{a} = [50; 0]^T$ using the implicitly restarted Arnoldi method showing expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

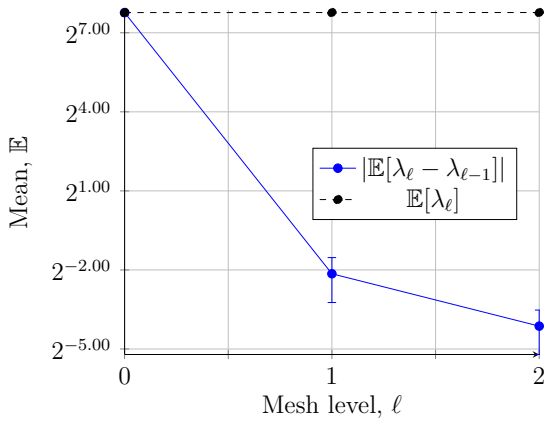
Level, ℓ	h^{-1}	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^5	2.176e+02	2.303e+02	2 901 605 ms	–	–	–
1	2^6	2.261e-01	1.686e-02	26 308 271 ms	9.91	13.7	3.18
2	2^7	5.705e-02	1.040e-03	247 639 813 ms	1.99	4.02	3.23

Similar results were obtained with the use of the MLMC method coupled with the Rayleigh quotient iteration. Here we observe from Table 4.10 that the average cost increase rate $\gamma \approx 3.18$ which is less than the variance reduction rate as well. As in the previous case, the complexity is $O(\varepsilon^{-2})$. Overall, the total computational cost of the MLMC method with the Rayleigh iteration is better than the cost of the MLMC method with the Arnoldi method, although this difference is only by a constant. We also note that for Problem I the result was reversed, with MLMC using the Arnoldi iteration yielding a better computational cost.

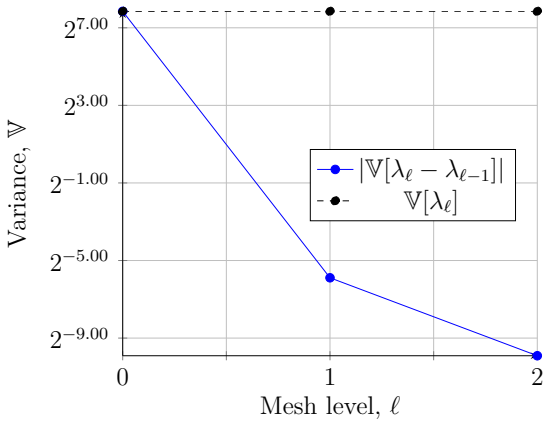
Now we use the homotopy method with the MLMC in hope of reducing the computational cost by introducing an additional coarse level. First, we use the same generating sequence of homotopy parameters as in Problem I, $t = 1 - 1/4^\ell$. This allows us to include

Table 4.10 – MLMC results using 10^4 samples on each level ℓ for Problem II with $\mathbf{a} = [50; 0]^T$ using the Rayleigh quotient method showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

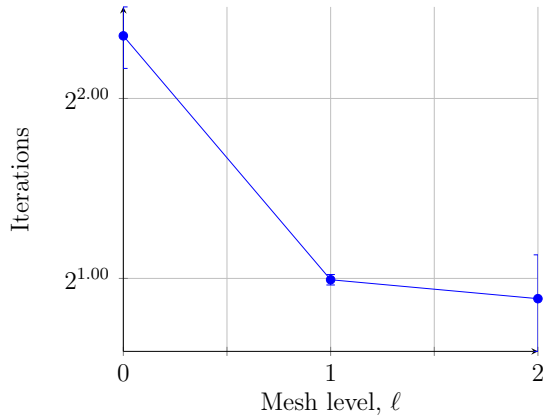
Level, ℓ	h^{-1}	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^5	2.176e+02	2.303e+02	2 048 706 ms	–	–	–
1	2^6	2.261e-01	1.686e-02	14 592 552 ms	9.91	13.7	2.83
2	2^7	5.705e-02	1.040e-03	175 312 392 ms	1.99	4.02	3.59



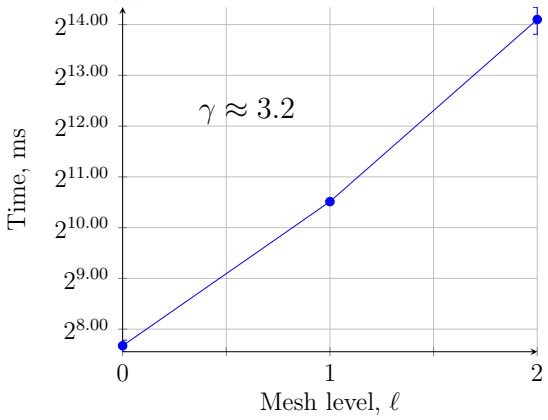
(a) Mean of difference.



(b) Variance of difference.



(c) Rayleigh quotient iterations.



(d) Computational time.

Figure 4.16 – Multi-level Monte Carlo method using 10^4 samples at each level to find the smallest eigenvalue of Problem I with $\mathbf{a} = [50; 0]^T$ using the FE approximation for the sequence of meshes, $h = 2^{-4} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. (a) expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b) variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c) average number of Rayleigh quotient iterations of computing the differences. (d) average computational time to find the difference of one sample.

the level with mesh size $h = 2^{-4}$ (but not a coarser one) while satisfying the stability condition. Compared to Problem I, each mesh discretization has a different homotopy parameter as a result of starting MLMC with the mesh size $h = 2^{-4}$ instead of $h = 2^{-3}$. Table 4.11 illustrates the convergence of the mean and variance of homotopy MLMC with

the Rayleigh quotient iteration. The variance reduction rate across the first two levels is less than the cost increase rate, $\beta \approx 1.15$ and $\gamma \approx 3.30$, and on the last level $\beta \approx 3.78$ and $\gamma \approx 3.56$. According to Theorem 1, the overall cost of homotopy MLMC is, then, $O(\varepsilon^{-2-(\gamma-\beta)/\alpha}) = O(\varepsilon^{-4.88})$ which is almost of the same cost as the standard Monte Carlo method ($O(\varepsilon^{-5})$).

Table 4.11 – Homotopy MLMC results using 10^4 samples on each level for Problem II with $\mathbf{a} = [50; 0]^T$ using the Rayleigh quotient method showing expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

ℓ	h^{-1}	t	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^4	0	1.199e+02	6.689e+02	223 361 ms	–	–	–
1	2^5	0.75	5.543e+01	2.703e+02	2 295 934 ms	1.11	1.30	3.36
2	2^6	0.9375	3.088e+01	7.218e+01	21 604 532 ms	0.38	1.01	3.23
3	2^7	1	1.166e+01	9.770e+00	255 320 523 ms	1.86	3.78	3.56

Next, we try to utilize the possibility of using the solution of the pure diffusion problem with $t = 0$. For that, we start with solving the problem with zero convection on the zeroth level. Then, on the subsequent levels we solve the full convection-diffusion problem with $t = 1$. The multi-level sequence is $\{(t = 0, h = 2^{-4}), (t = 1, h = 2^{-5}), (t = 1, h = 2^{-6}), (t = 1, h = 2^{-7})\}$. Table 4.12 shows the results for such sequence of levels. We see from this table that the mean of the difference becomes smaller with each level, but the jump in the variance of the difference from the zeroth to the first level indicates the absence of the convergence in variance for these two levels. Because the difference between the solution of the diffusion and convection-diffusion problems becomes too significant, the method is not applicable for the use with high velocities.

Table 4.12 – MLMC results using 10^4 samples on each level l for Problem II with $\mathbf{a} = [50; 0]^T$ using the Rayleigh quotient method showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

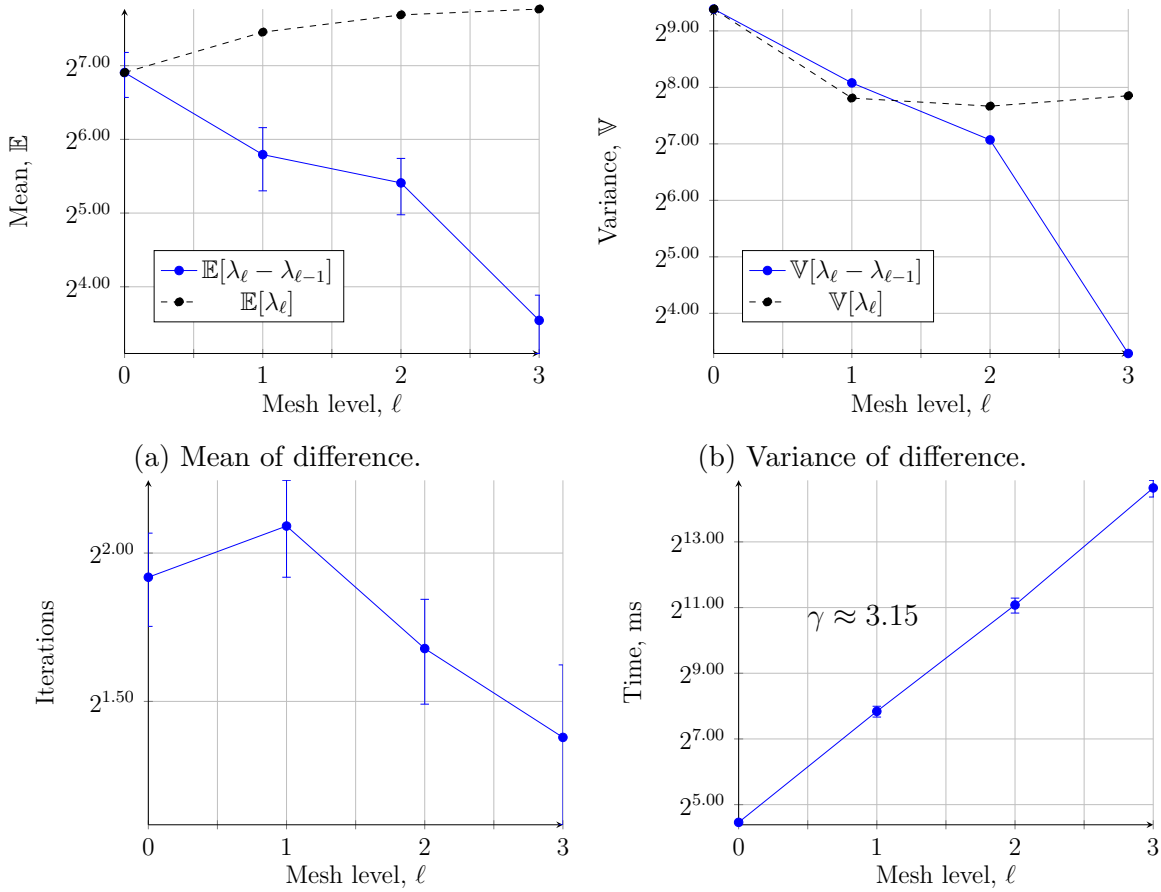
ℓ	h^{-1}	t	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^4	0	1.199e+02	6.689e+02	223 361 ms	–	–	–
1	2^5	1	9.769e+01	7.762e+02	2 048 706 ms	0.29	-0.21	3.20
2	2^6	1	2.261e-01	1.686e-02	14 592 552 ms	8.75	15.49	2.83
3	2^7	1	5.705e-02	1.040e-03	175 312 392 ms	1.99	4.01	3.59

In the next test, we remove the level containing the mesh size $h = 2^{-4}$ and introduce the pure diffusion problem on the level with the mesh size $h = 2^{-5}$. As the pure diffusion operator is self-adjoint and the derived matrix is symmetric, the time spent on solving the eigenvalue problem is approximately twice smaller compared to the full convection-diffusion eigenvalue problem. The resulting sequence of levels is $\{(t = 0, h = 2^{-5}), (t = 1, h = 2^{-5}), (t = 1, h = 2^{-6}), (t = 1, h = 2^{-7})\}$. Table 4.13 shows that the mean difference as in the previous case decreases across all levels but the variance of the difference still

shows no convergence on the first two levels. As in the previous case, unfortunately, the application of the pure diffusion problem is not beneficial.

Table 4.13 – MLMC results using 10^4 samples on each level ℓ for Problem II with $\mathbf{a} = [50; 0]^T$ using the Rayleigh quotient method showing the expectation value, $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$, variance of the difference, $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$, total computational time in ms, and their convergence rates, α, β, γ (see Theorem 1).

ℓ	h^{-1}	t	$\lvert\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]\rvert$	$\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$	Time	α	β	γ
0	2^5	0	1.187e+02	6.541e+02	1 391 730 ms	–	–	–
1	2^5	1	9.880e+01	7.617e+02	2 048 706 ms	0.26	-0.21	0.55
2	2^6	1	2.261e-01	1.686e-02	14 592 552 ms	8.77	15.46	2.83
3	2^7	1	5.705e-02	1.040e-03	175 312 392 ms	1.99	4.02	3.59



(a) Mean of difference.

(b) Variance of difference.

(c) Rayleigh quotient iterations.

(d) Computational time.

Figure 4.17 – Homotopy MLMC method using 10^4 samples at each level to find the smallest eigenvalue of Problem I with $\mathbf{a} = [50; 0]^T$ using the FE approximation for the sequence of meshes, $h = 2^{-4} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. The homotopy sequence is $\{0, 0.75, 0.9385, 1\}$. (a) expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b) variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c) average number of Rayleigh quotient iterations of computing the differences. (d) average computational time to find the difference of one sample.

Figure 4.18 summarizes the total computational cost vs. mean square error for MLMC

with the Rayleigh quotient iteration, MLMC with the implicitly restarted Arnoldi method, and homotopy MLMC with the Rayleigh quotient iteration presented in Table 4.11. This time the difference between the MLMC with the Arnoldi and Rayleigh quotient methods is insignificant compared to the results presented in Figure 4.15 with a lower velocity. The computational complexity of the geometric MLMC with both eigenvalue solvers is $O(\varepsilon^{-2})$. On the other hand, the total cost of the homotopy MLMC is unclear and it seems that for higher velocities the convergence becomes worse than for lower velocities. As the mesh sequence starts with a much finer mesh for high velocities, the overall cost of the multi-level Monte Carlo method increases and is equal to the computational cost of the coarsest level, although it outperforms the standard Monte Carlo method which depends only on the finest mesh for the targeted MSE.

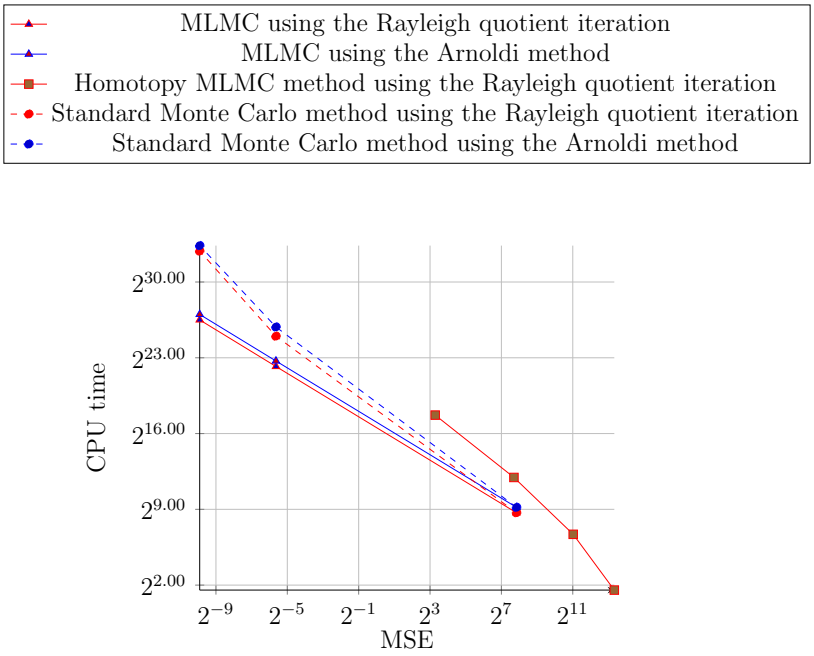


Figure 4.18 – CPU time vs. mean square error of MLMC for Problem II with $\mathbf{a} = [50, 0]^T$.

In the next chapter, we explore another possibility of introducing coarse mesh models.

Chapter 5

Multi-level Monte Carlo method with the streamline-upwind Petrov-Galerkin method

In the previous chapter we considered incorporating the standard Galerkin approximation into the multi-level Monte Carlo setting for computing the smallest eigenvalue of the convection-diffusion problem with a random conductivity. The developed methods showed good approximation properties when solving the problem with low velocity. However, for problems with high velocities, the finite element method requires a much finer mesh, as the solutions obtained on coarser meshes exhibit non-physical oscillations. As a result, the starting level of the MLMC method contains a very fine model which takes a significant amount of time to solve. As such, various remedy strategies based on the homotopy continuation method were proposed, which later deemed unsuitable as showed by the numerical simulations.

In this chapter, we consider an alternative approach which is based on the Petrov-Galerkin finite element approximation. Compared to the standard Galerkin method, a stabilization parameter is used to resolve the oscillations properly. While the trial and test spaces are the same in the standard FE scheme, in the Petrov-Galerkin method they are different. The stabilization parameter used in the Petrov-Galerkin method usually considers a measure of the local element as well as the local Peclet number. Here we use the streamline-upwind/Petrov-Galerkin (SUPG) formulation in the multi-level Monte Carlo simulations.

The chapter itself is rather short as the main preliminaries, such as the stochastic convection-diffusion problem, the eigenvalue solver, and the multi-level Monte Carlo method were introduced in the previous chapter. We introduce the SUPG scheme with its function spaces in Section 5.1. Then, we present the numerical results of the SUPG MLMC method for high velocities and show the superiority of the developed method compared to the FE MLMC method.

5.1 Streamline-upwind Petrov-Galerkin method

In this section we introduce the streamline-upwind/Petrov-Galerkin method. We impose the same assumptions on the random field $\kappa(\omega; x)$ that are described (A1 and A2) in Section 4.2.

The streamline-upwind/Petrov-Galerkin method was introduced by Brooks and Hughes in 1982 [16] as a means to stabilize the finite element solution. Since then, the method has been a subject of extensive research and been used in various applications [12, 23, 45, 50, 52, 59]. The SUPG method can be derived in several ways. Here, we consider its formulation through adding a stabilization term to the bilinear form. An equivalent weak formulation can be obtained by defining a test space with additional test functions in the form $\hat{v}(x) = v(x) + p(x)$, where $v(x)$ is a standard test function in the finite element method and $p(x)$ is an additional discontinuous function.

5.1.1 Weak formulation

The residual of the convection-diffusion equation is

$$\mathcal{R}(u(\omega);\omega) = \mathbf{a}(x;\omega)\cdot\nabla u(\omega) - \nabla\cdot\kappa(\omega)\nabla u(\omega) - \lambda(\omega)u(\omega) = \mathcal{L}(u(\omega);\omega) - \lambda(\omega)u(\omega), \quad (5.1)$$

where $\mathcal{L}(\cdot;\omega)$ is the differential operator. The stabilization techniques are applied for each element interior only, because the residual $\mathcal{R}(u(\omega);\omega)$ is computed only on the finite elements. A general formulation of the stabilized finite element methods can be defined in the following form

$$\int_D \mathbf{a}(\omega)\cdot\nabla u(x)v(x)\,\mathrm{d}x - \int_D \kappa(x;\omega)\nabla u(x)\nabla v(x)\,\mathrm{d}x + \sum_k \int_{D_k} \tau_k \mathcal{R}(u(\omega);\omega)\mathcal{P}(v;\omega)\,\mathrm{d}x = \lambda(\omega)\int_D u(x)v(x)\,\mathrm{d}x, \quad (5.2)$$

where $\mathcal{P}(v;\omega)$ is some operator and τ_k is the stabilization parameter acting in the kth finite element.

Various definitions exist for the operator $\mathcal{P}(v;\omega)$, such as the Galerkin/Least squares (GLS) method [51], the Streamline-Upwind/Petrov-Galerkin (SUPG) method [15, 16, 30], the Unusual Stabilized Finite Element (USFEM) method [7], etc.

For the SUPG method the stabilization operator $\mathcal{P}(v;\omega)$ defined as

$$\mathcal{P}(v;\omega) = \mathbf{a}(\omega)\cdot\nabla v. \quad (5.3)$$

Substituting Equations (5.1) and (5.3) into (5.2) gives the streamline-upwind/Petrov-

Galerkin weighted residual formulation

$$\int_D \mathbf{a}(\omega)\cdot\nabla u(\omega)v\,\mathrm{d}x - \int_D \nabla\cdot\kappa(\omega)\nabla u(\omega)\nabla v\,\mathrm{d}x + \sum_k \int_{D_k} \tau_k(\mathbf{a}(\omega)\cdot\nabla u(\omega) - \kappa(\omega)\nabla u(\omega) - \lambda(\omega)u(\omega))(\mathbf{a}(\omega)\cdot\nabla v)\,\mathrm{d}x = \lambda(\omega)\int_D u(\omega)v\,\mathrm{d}x. \tag{5.4}$$

5.1.2 Finite element matrices

After approximation of the weak form (Eq. 5.4) by usual finite-dimensional subspaces, we obtain the discrete variational problem: find non-trivial primal and dual eigenpairs $(\lambda(\omega), u_h(\omega)) \in \mathbb{C}\times V_g^h$ and $(\lambda^*(\omega), u_h^*(\omega)) \in \mathbb{C}\times V_g^h$ such that

$$\mathcal{A}(u_h, v_h;\omega) + \mathcal{C}(u_h(\omega), v_h;\omega) + \sum_k \int_{D_k} \tau_k(\mathbf{a}(\omega)\cdot\nabla u_h - \nabla\cdot\kappa(\omega)\nabla u_h - \lambda(\omega)u_h)(\mathbf{a}(\omega)\cdot\nabla v_h)\mathrm{d}x = \lambda(\omega)b(u_h, v_h;\omega), \tag{5.5}$$

$$\mathcal{A}(w_h, u_h^*(\omega);\omega) + \mathcal{C}(w_h, u_h^*(\omega);\omega) + \sum_k \int_{D_k} \tau_k(\mathbf{a}(\omega)\cdot\nabla w_h - \nabla\cdot\kappa(\omega)\nabla w_h - \overline{\lambda^*}(\omega)w_h)(\mathbf{a}(\omega)\cdot\nabla u_h^*)\mathrm{d}x = \overline{\lambda^*}(\omega)b(w_h, u_h^*;\omega). \tag{5.6}$$

The discrete primal and dual formulations for the right and left eigenfunctions in the matrix form after substituting the eigenfunctions u_h, w_h and test functions v_h, u_h^* with their corresponding linear combinations (Eq. (4.15), Sec. 4.2.2) in the finite spaces V_g^h and V_0^h are

$$\tilde{\mathbf{A}}(\omega)\mathbf{q} = \lambda(\omega)\tilde{\mathbf{M}}(\omega)\mathbf{q}, \qquad \mathbf{q}^H\tilde{\mathbf{A}}(\omega) = \overline{\lambda}(\omega)\mathbf{q}^H\tilde{\mathbf{M}}(\omega), \tag{5.7}$$

with matrix elements defined as

$$\tilde{A}_{ij} = \sum_k \int_{D_k} \kappa(\omega)\nabla\psi_j\nabla\psi_i\,\mathrm{d}x + \sum_k \int_{D_k} \mathbf{a}(\omega)\cdot\psi_j\nabla\psi_i\,\mathrm{d}x, + \sum_k \int_{D_k} \tau_k\mathbf{a}(\omega)\cdot\psi_i\nabla\psi_j\,\mathrm{d}x, \tag{5.8}$$

$$\tilde{M}_{ij} = \sum_k \int_{D_k} \psi_j\psi_i + \tau_k\mathbf{a}(\omega)\cdot\nabla\psi_j\psi_i\,\mathrm{d}x, \quad i,j = \overline{1\ldots n}. \tag{5.9}$$

In general, the problem of finding an optimal stabilization parameter τ_K is open and can be defined by multiple ways [59]. We employ the following stabilization parameter [12,

45]

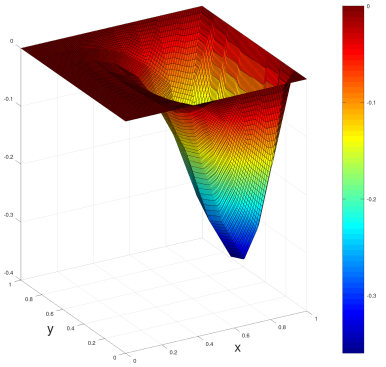
$$\tau_k(\omega) = \frac{h_k}{2|\mathbf{a}(\omega)|}(\coth|\mathbf{Pe}_k|(\omega) - \frac{1}{|\mathbf{Pe}_k|(\omega)}), \tag{5.10}$$

where Pe_k denotes the local Peclet mesh number. Though, in practical simulations the asymptotic expressions are used

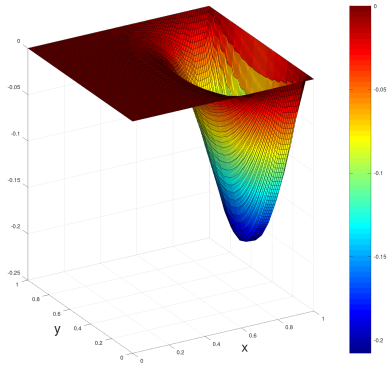
$$\tau_k = \begin{cases} \frac{h_k}{2|a|(\omega)}, & Pe_k(\omega) \geq 1, \\ \frac{h_k^2}{12\kappa(\omega)}, & Pe_k(\omega) < 1, \end{cases} \tag{5.11}$$

where $Pe_k \equiv |\mathbf{Pe}_k|$. It follows that the right-hand side matrix $\tilde{\mathbf{M}}$ is no longer symmetric and is random compared to the mass matrix in the standard Galerkin method.

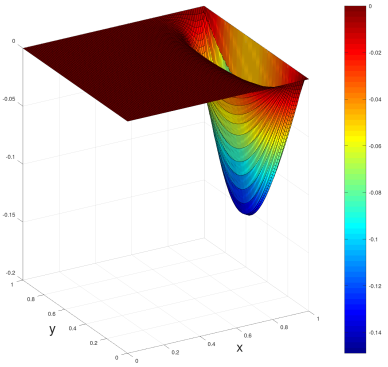
Figure 5.1 shows numerical results for SUPG with velocity $\mathbf{a} = [50, 0]^T$ and conductivity $\kappa = 1$ on the mesh sequence starting from $h = 2^{-3}$ to $h = 2^{-6}$. From Figure 5.1 it can be seen that the solutions obtained on the grids (Figures 5.1a and 5.1b) with Peclet numbers $|\mathbf{Pe}| \geq 1$ are smooth and do not have any non-physical oscillations as it is in the case with the standard Galerkin method.



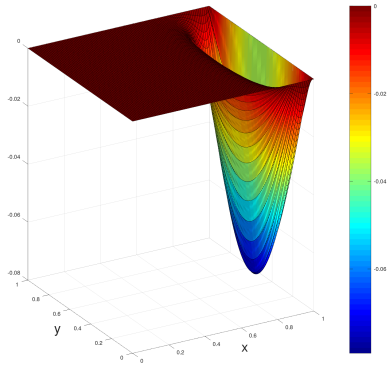
(a) $h = 2^{-3}, Pe = 3.125$.



(b) $h = 2^{-4}, Pe = 1.5625$.



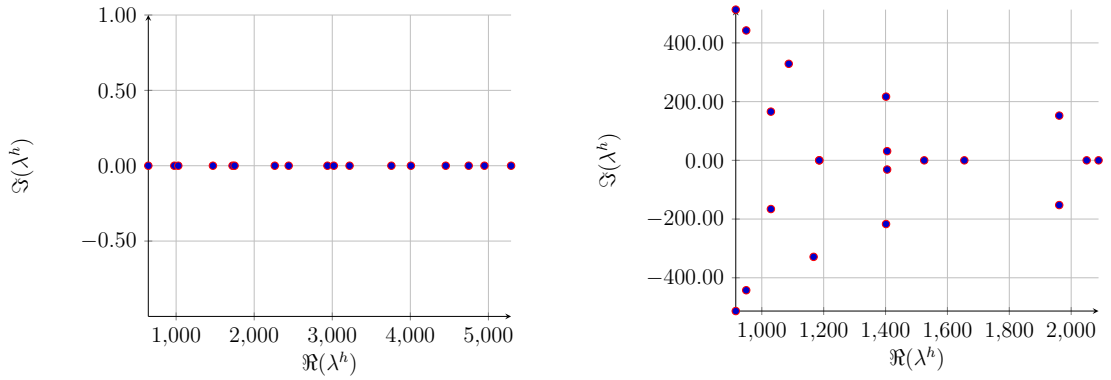
(c) $h = 2^{-5}, Pe = 0.78125$.



(d) $h = 2^{-6}, Pe = 0.390625$.

Figure 5.1 – SUPG eigenfunction approximations u_h with homogeneous boundary conditions for different mesh sizes, $\mathbf{a}(x) = [50, 0]^T$ and $\kappa(x) = 1$ where Pe is the Peclet number.

Figure 5.2 shows the first 20 smallest eigenvalues for a single realization of random field $\kappa(x)$ with velocity $\mathbf{a} = [100, 0]^T$ on the mesh with size $h = 2^{-3}$. The standard Galerkin method has non-physical oscillations in the solution for such coarse mesh and its smallest eigenvalue is a complex pair, as mentioned in Section 4.2.3. The SUPG method, on the other hand, has a real smallest eigenvalue, indicating a stable solution.



(a) SUPG eigenvalues.

(b) Finite element eigenvalues.

Figure 5.2 – Eigenvalue spectrum (the first 20) of the convection-diffusion operator for a single realization of random field $\kappa(\mathbf{x})$ with $\mathbf{a} = [100, 0]^T$ approximated by SUPG (*Left*) and FEM (*Right*) with mesh size $h = 2^{-3}$.

5.2 Numerical results

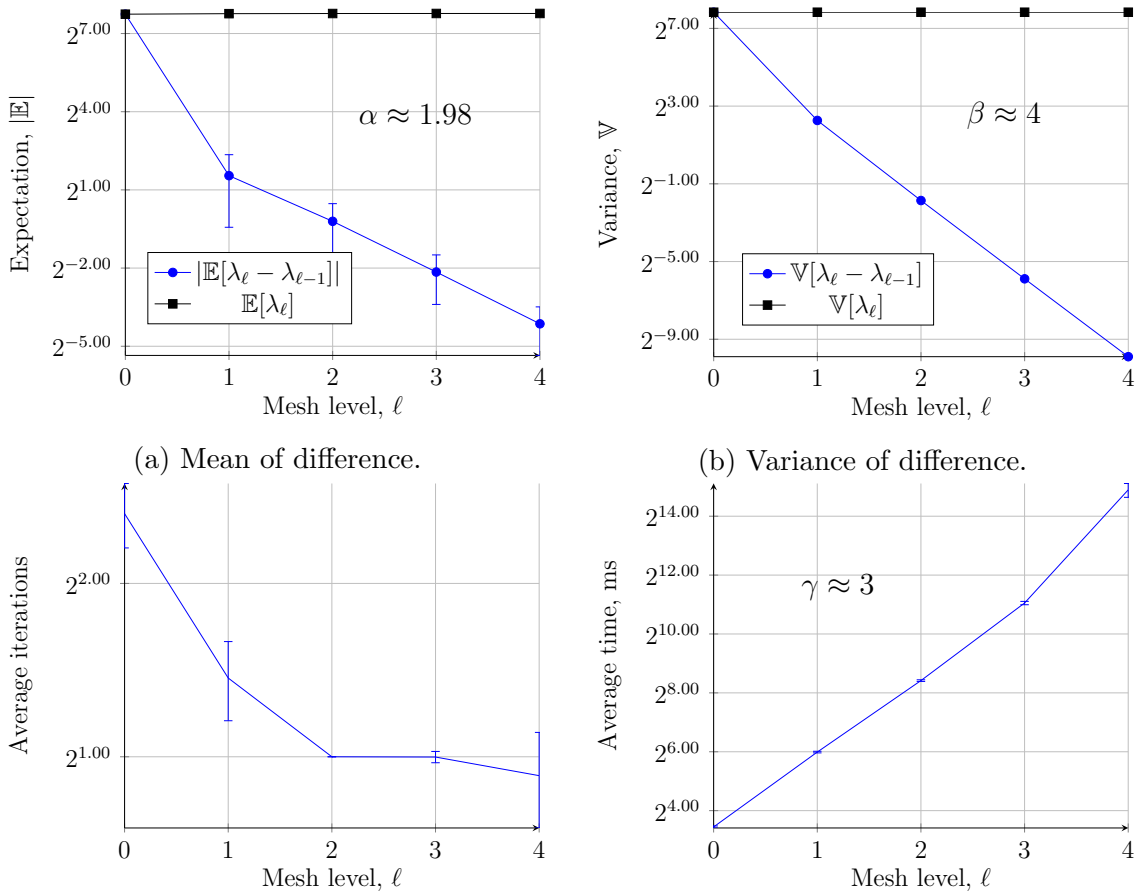
We present two numerical examples, in which we demonstrate the application of the streamline-upwind/Petrov-Galerkin method for cases with high velocities. As usual, the domain $D = [0; 1] \times [0; 1]$ is the square unit and the conductivity $\kappa(x; \omega)$ is modelled as a log-uniform random field described in Section 3.2.

First, we consider Problem II presented in the previous chapter in section 4.8.2. Then, we increase velocity and change its direction, so that the standard finite element approximation would require a very fine mesh in order to produce a solution without spurious oscillations. As shown by numerical results in Chapter 4, the Rayleigh quotient and implicitly restarted Arnoldi iterations have a similar computational complexity $O(h^{-3})$ but the Rayleigh quotient iteration was slightly faster for both Problem I and II. This is why we employ here only the Rayleigh quotient solver.

5.2.1 Problem II

The problem setup is described in Chapter 4 in Section 4.8.2. Figure 5.3 illustrates the numerical results for the streamline-upwind/Petrov-Galerkin method in the multi-level-Monte Carlo setting using the Rayleigh quotient as the eigenvalue solver using 10^4 random samples at each level. Figure 5.3c reports the average number of the Rayleigh quotient

iterations at each level. The number of iterations required to solve the problem is decreasing with level, almost obtaining the solution at two or less iterations on the last levels with little or no variance. As it can be seen from Figures 5.3a and 5.3b, the mean and variance are decreasing steadily. The mean reduction rate is $\alpha \approx 2$ while the variance reduction rate is $\beta \approx 4$. At the same time, the cost increasing rate is $\gamma \approx 2.8$ on average, with the last level having $\gamma \approx 3.8$. Therefore, the computational complexity of the SUPG MLMC is $O(\varepsilon^{-2})$ because $\beta > \gamma$ (Theorem 1) across all levels.



(a) Mean of difference.

(b) Variance of difference.

(c) Rayleigh quotient iterations.

(d) Computational time.

Figure 5.3 – Multi-level Monte Carlo method using 10^4 samples at each level to find the smallest eigenvalue of Problem II with $\mathbf{a} = [50; 0]^T$ using the SUPG approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. (a): Expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b): Variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c): Average number of Rayleigh quotient iterations for computing the differences. (d): Average computational time for the difference of one sample.

Figure 5.4 shows the CPU time vs. MSE. The black lines indicate SUPG MLMC, while the red lines correspond to the standard finite element (FE) MLMC method. We observe that the complexity of FE MLMC and SUPG MLMC has the same rate $O(\varepsilon^{-2})$. However, they differ by a constant and SUPG MLMC is approximately ten times cheaper. This is because the SUPG MLMC method allows one to form a multi-level sequence with much

coarser levels than FE MLMC.

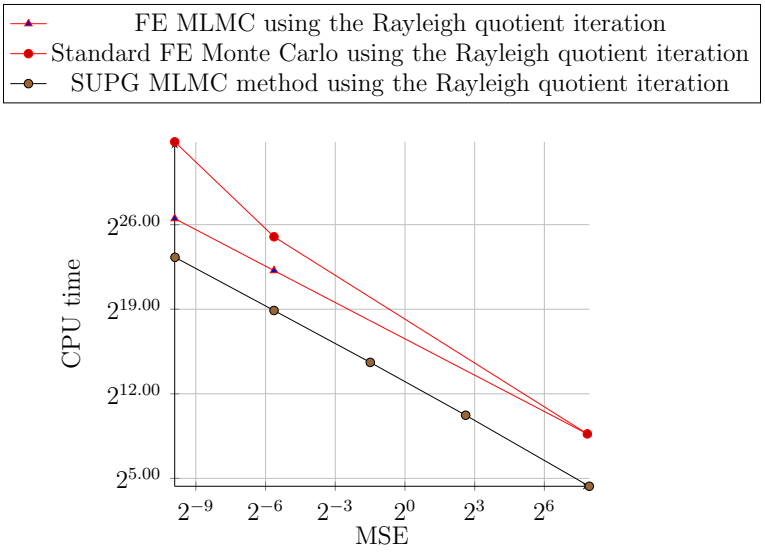
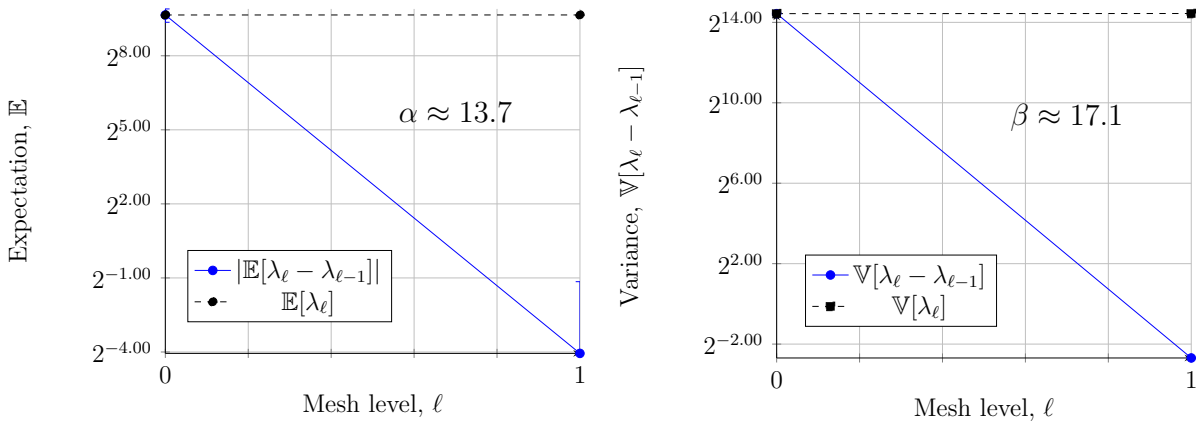


Figure 5.4 – CPU time vs. mean square error of FE MLMC and SUPG MLMC for Problem II with $\mathbf{a} = [50, 0]^T$.

5.2.2 Problem III

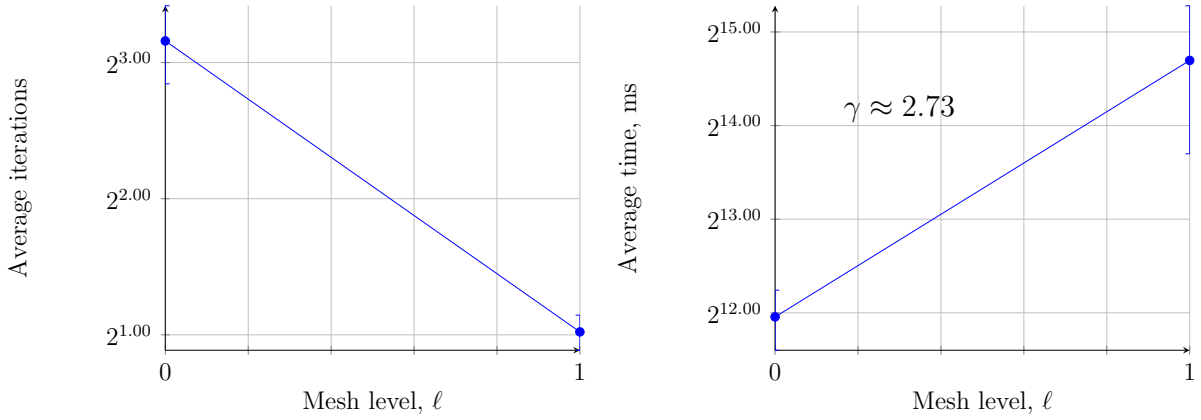
In this experiment we increase the velocity and change its direction to have $\mathbf{a} = [100, 100]^T$. It is also known as convection skew to the mesh. As a consequence, the solution may experience cross-wind diffusion, allowing to form some discontinuities in the direction of flow [103]. Both our methods are based on the continuous Galerkin method, as such they are not able to model actual discontinuities in the solution. The standard Galerkin may have non-physical oscillations everywhere in the solution on coarse meshes. SUPG, on the other hand, may have spurious oscillations localized in narrow regions on the same coarse meshes.

With limited capacity resources, we can compute FE MLMC with only two levels of meshes, essentially making it two-level Monte Carlo with the following sequence $\{h_0 = 2^{-6}, h_1 = 2^{-7}\}$. Figure 5.5 shows the numerical results for FE MLMC using 10^4 samples at each level ℓ. The absolute value of mean of difference $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ as well as the expectation $\mathbb{E}[\lambda_\ell]$ are shown by blue and black lines in Figure 5.5a, respectively. Similarly, their variance is shown in Figure 5.5b. The average number of Rayleigh quotient iterations is illustrated in Figure 5.5c with average cost of a sample shown in Figure 5.5d. The slope of the mean of difference α is approximately 13.7, while the variance reduction rate is $\beta \approx 17.1$. The cost increase rate is $\gamma \approx 2.73$. The average number of iterations on the level zero is about 8 with some deviation, but it decreases rapidly to approximately 2 on the next level. The overall computational complexity of FE MLMC, therefore, is $O(\varepsilon^{-2})$ asymptotically as $\beta > \gamma$, though the mesh sequence starts with a finer mesh compared to Problem I and II.



(a) Mean of difference.

(b) Variance of difference.



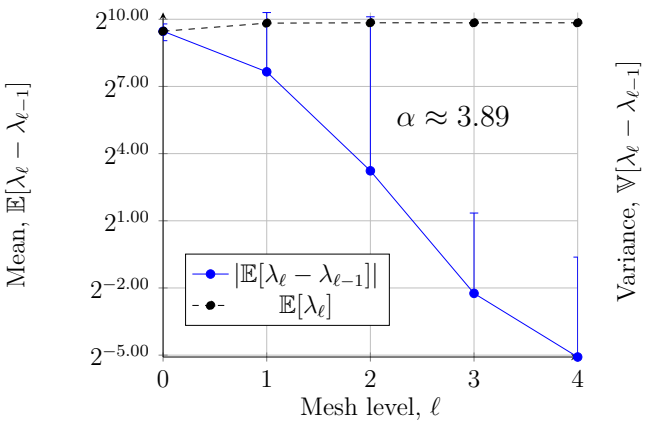
(c) Rayleigh quotient iterations.

(d) Computational time.

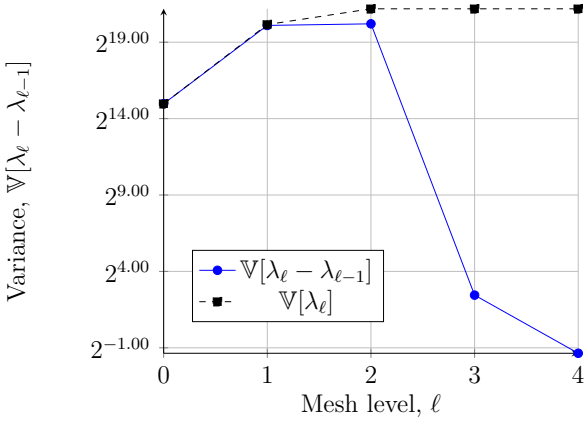
Figure 5.5 – Two-level Monte Carlo method using 10^4 samples at each level l to find the smallest eigenvalue of Problem III with $\mathbf{a} = [100; 100]^T$ using the finite element approximation for the sequence of meshes, $h = 2^{-6}, 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. (a): Expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b): Variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c): Average number of Rayleigh quotient iterations for computing the differences. (d): Average computational time for the difference of one sample.

We now examine the behaviour of the SUPG MLMC method with the Rayleigh quotient iteration for the same case. The SUPG MLMC mesh sequence is the following $\{h_0 = 2^{-3}, h_1 = 2^{-4}, h_2 = 2^{-5}, h_3 = 2^{-6}, h_4 = 2^{-7}\}$. Figure 5.6 presents the numerics for SUPG MLMC with the Rayleigh quotient iteration using 10^4 samples at each mesh level. Figure 5.6a shows the expectation of difference $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ at each level. The mean decreases with the rate $\alpha \approx 3.89$ which is larger than for the previous case ($\alpha_{II} \approx 1.98$). However, Figure 5.6b illustrates that the variance of difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ has a different behaviour compared to in Problem II (Figure 5.3b). The variance increases from level zero to level one and then it remains almost the same on the level two. After that, the variance $\mathbb{V}[\lambda_3 - \lambda_2]$ decreases abruptly from $\approx 2^{19}$ to $\approx 2^3$. Therefore, the SUPG MLMC sequence should start with level three ($h = 2^{-5}$) in order to get benefits of the MLMC method. In Figure 5.6c the average number of Rayleigh quotient iterations is shown at

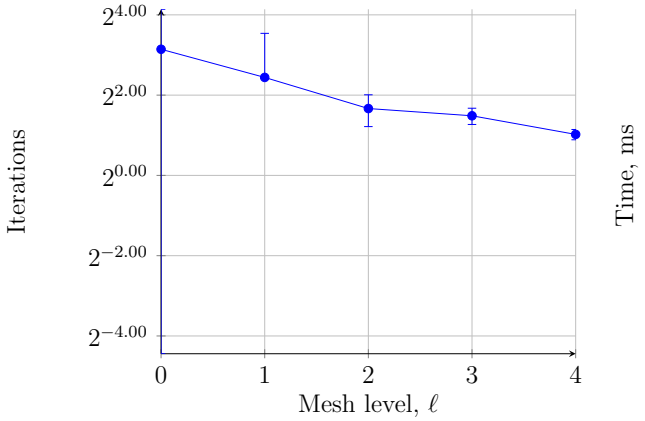
each level ℓ. Only the lowest level experiences large deviation in the number of iterations while other levels have small variance. The average computational time (Figure 5.6d) changes at different rates having on average $\gamma \approx 3$.



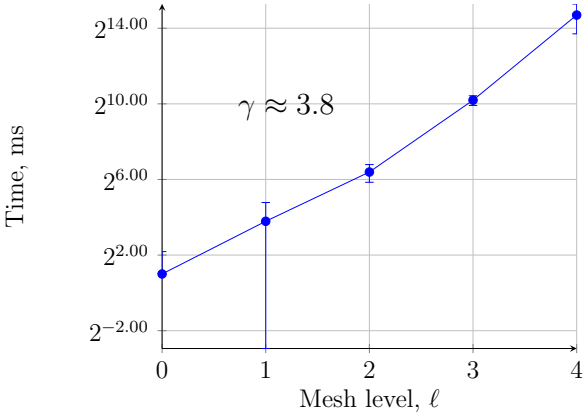
(a) Mean of difference.



(b) Variance of difference.



(c) Rayleigh quotient iterations.



(d) Computational time.

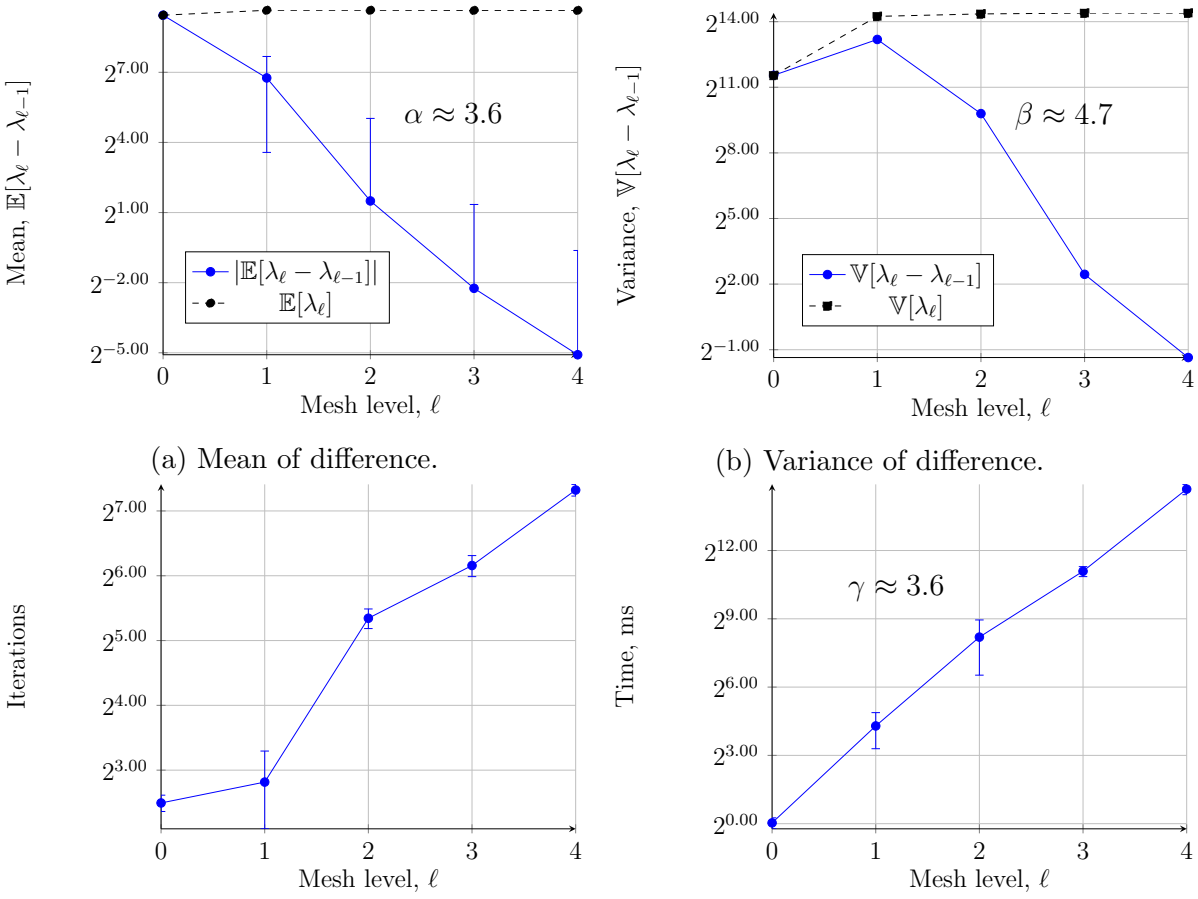
Figure 5.6 – Multi-level Monte Carlo method using 10^4 samples at each level to find the smallest eigenvalue of Problem III with $\mathbf{a} = [100; 100]^T$ using the SUPG approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the Rayleigh quotient iteration as the eigenvalue solver. (a): Expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b): Variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c): Average number of Rayleigh quotient iterations for computing the differences. (d): Average computational time for the difference of one sample.

To further investigate the issue of the variance of difference $\mathbb{V}[\lambda_3 - \lambda_2]$, we perform the same test but with the implicitly restarted Arnoldi method instead. The numerical results for SUPG MLMC with Arnoldi method using 10^4 samples at each level are presented in Figure 5.7. Figure 5.7a shows the expectation of difference $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ at each level. The mean decreases with a slightly different rate $\alpha \approx 3.6$ compared to case with the Rayleigh quotient iteration ($\alpha \approx 3.89$). However, Figure 5.7b illustrates that the variance of difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ has a different behaviour compared to the previous case with the Rayleigh quotient iteration (Figure 5.6b). Similarly, the variance increases from level zero to level one. However, after that, the variance $\mathbb{V}[\lambda_2 - \lambda_1]$ decreases from $\approx 2^{13}$ to $\approx 2^9$.

Figure 5.7c shows the average number of Arnoldi iterations while Figure 5.7d show the average computational time of the Arnoldi method at each level ℓ. The cost increase rate $\gamma \approx 3.6$ is lower than the variance reduction rate $\beta \approx 4.7$ yielding an $O(\varepsilon^{-2})$ complexity (Theorem 1).

The difference in the behaviour of the Rayleigh quotient and Arnoldi iterations is because the coarsest mesh $h = 2^{-3}$ in this case contains complex eigenvalues for some realizations of the random field $\kappa(x;\omega)$ due to cross-diffusion in Problem III and we have no complex arithmetic in our current implementation of the iterative solver. As a consequence, the Rayleigh quotient iteration converges to the smallest *real* eigenvalue while the Arnoldi method converges to the smallest eigenvalue which is *complex* for some samples on the mesh with $h = 2^{-3}$. Moreover, as we employ a two-grid method for the Rayleigh quotient iteration to obtain the solution on the adjacent mesh levels, the approximate eigenvalue obtained on the coarse mesh propagates into the solution on the next mesh level. As a result, the Rayleigh quotient iteration converges to the closest to this eigenvalue on the finer mesh instead of to the actual smallest one.

CPU time vs. MSE plot for Problem III is shown in Figure 5.8. All three methods give an $O(\varepsilon^{-2})$ complexity. However, SUPG MLMC and FEM MLMC with the Rayleigh quotient iteration perform almost similarly in terms of the computational complexity. SUPG MLMC with the implicitly restarted Arnoldi method, on the other hand, is more robust when computing the eigenvalues of the convection-diffusion operator. As a result, the difference in the computational cost is about 2^8 between SUPG MLMC with the Arnoldi method and SUPG / FE MLMC with the Rayleigh quotient method.



(a) Mean of difference.

(b) Variance of difference.

(c) Arnoldi iterations.

(d) Computational time.

Figure 5.7 – Multi-level Monte Carlo method using 10^4 samples at each level to find the smallest eigenvalue of Problem III with convection skew to mesh $\mathbf{a} = [100; 100]^T$ using the SUPG approximation for the sequence of meshes, $h = 2^{-3} \dots 2^{-7}$ and the implicitly restarted Arnoldi method as the eigenvalue solver. (a): Expectation of the eigenvalue $\mathbb{E}[\lambda_\ell]$ (black line) and of the difference between two levels $|\mathbb{E}[\lambda_\ell - \lambda_{\ell-1}]|$ (blue line). (b): Variance of the eigenvalue $\mathbb{V}[\lambda_\ell]$ (black line) and of the difference $\mathbb{V}[\lambda_\ell - \lambda_{\ell-1}]$ (blue line). (c): Average number of Arnoldi iterations for computing the differences. (d): Average computational time for the difference of one sample.

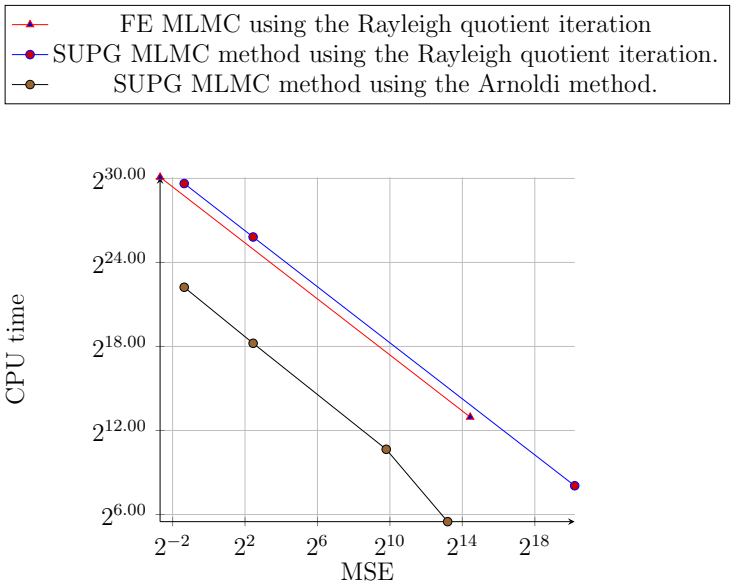


Figure 5.8 – CPU time vs. mean square error for FE MLMC, SUPG MLMC with the Rayleigh quotient and Arnoldi methods for Problem III with convection skew to mesh.

Chapter 6

Conclusion and Future work

6.1 Thesis summary

In this thesis we considered and developed various strategies for the multi-level and multi-index Monte Carlo (MLMC and MIMC) methods for solving elliptic problems in 2D with uncertainties related to the conductivity. In Chapter 2, we introduced the general concepts of MLMC and MIMC on which the other chapters are based. We established that under certain circumstances MLMC and MIMC outperform the standard Monte Carlo method for quantities of interest (QoI) that are computed as discretized functionals. For such quantities we required a construction of a convergent sequence of discretized models to our targeted quantity. It was shown that if the sequence converges faster than its cost increase then MLMC and MIMC have a better complexity than the standard Monte Carlo method. Chapter 3 investigated and developed four MLMC and two MIMC methods to compute two quantities of interest based on the solution of an elliptic PDE; one QoI is the average solution in a given volume and the other is the average flux at given locations. The first MLMC method was geometric MLMC in which the sequence of discretized elliptic problems was constructed by halving the mesh size with level. The second MLMC scheme was the geometric MLMC method but with the use of second-order basis functions at each level. The third MLMC method was hp-MLMC in which the approximate sequence was constructed by simultaneously increasing the polynomial order of basis functions and refining the mesh with level. The last MLMC was p-MLMC in which the sequence was obtained by only increasing the order of basis functions without any mesh refinement. Next, we introduced $h_x h_y$-MIMC which was the geometric multi-index Monte Carlo method utilizing directional refinement in h_x and h_y. Finally, we developed a new MIMC method ($h_x p_x, h_y p_y$-MIMC) based on incomplete basis functions in 2D. All presented methods yielded the same optimal complexity $O(\varepsilon^{-2})$ for both QoIs.

In Chapter 4 we considered the geometric and homotopy MLMC alongside the finite element approximation for finding the smallest eigenvalue of the convection-diffusion operator with randomness in conductivity for cases with low and high velocity. The newly developed homotopy MLMC method was based on the homotopy continuation method,

utilizing solutions with lower velocities. Rayleigh quotient and implicitly restarted Arnoldi iterations were incorporated in both MLMC methods. For the case with low velocity, the developed MLMC methods showed $O(\varepsilon^{-2})$ complexity compared to the standard Monte Carlo method which has $O(\varepsilon^{-5})$ complexity. Geometric MLMC with the Arnoldi method, overall, performed better than all other presented methods, because the main contribution in the computational cost comes from the coarsest level and the Arnoldi method had the cheapest cost on this level, although the Rayleigh quotient iteration was cheaper on all other levels. For the case with high velocity, geometric MLMC with the Arnoldi and Rayleigh quotient iterations performed similar to each other having $O(\varepsilon^{-2})$ complexity but the overall computational cost was higher than for the case with low velocity as a result of starting the multi-level sequence with a finer mesh in order to obtain a stable solution. However, homotopy MLMC showed almost the same cost $O(\varepsilon^{-4.9})$ as the standard Monte Carlo method $O(\varepsilon^{-5})$.

Chapter 5 proposed an alternative approach in dealing with spurious oscillations in convection-dominated regions. The geometric MLMC was coupled together with the streamline-upwind/Petrov-Galerkin method (SUPG) to obtain an efficient method.

6.2 Future work

In order to better understand the presented methods, further research is required in several directions. First, one can consider the extension to higher-dimensional PDEs, such as 3D elliptic problems. The variance reduction rate of the geometric MLMC is the same for a continuous QoI and does not depend on the dimension of the underlying PDE with the use of the finite element discretization with linear basis functions, which results in a variance reduction rate of $O(h^{-4})$. On the other hand, the cost increase rate depends on the dimension of the PDE. In case of 2D elliptic problems, the cost of solving the discretized problem is $O(h^{-2})$ using iterative solvers with naive implementation or $O(h^{-3/2})$ using direct solvers with permutations. In 3D, the situation is different and the direct solvers have a poor performance. As such, MIMC may perform significantly better compared to MLMC, because MIMC complexity depends on the directional variance and cost, although there is an additional constraint on the mixed regularity of the QoI for some MIMC methods. There is another direction in which the geometric MLMC can be improved. Multigrid and multilevel solvers utilize in a similar fashion a sequence of discretized grids or a sequence of function spaces. Because of that, they can be coupled with the geometric MLMC when computing the difference between two adjacent levels for one sample.

Second, the proposed $h_x p_x, h_y p_y$-MIMC showed good performance, even in the case of the discontinuous QoI where $h_x h_y$-MIMC experienced oscillations in variance.

Third, further investigation is required in case of non-self-adjoint eigenvalue problems. As previously stated, more research is needed for 3D problems. The Rayleigh quotient

iteration may perform poorly for higher-order problems because it requires the solution of ill-conditioned linear systems; as such direct solvers may not be applied because they poorly scale with the dimension of the problem and iterative solvers may not converge, so special care should be taken. On the other hand, the Arnoldi method requires only matrix-vector products and solving a system of linear equations with mass matrix. Moreover, cases with random velocity and inhomogeneous boundary conditions should be considered as well. Finally, the SUPG MLMC method and its possible extension SUPG MIMC should be investigated for cases with conductivity modelled as a log-normal random field. For cases with convection skew to mesh, various stabilization parameters should be considered to deal with spurious oscillations properly.

Bibliography

[1] R. J. Adler. *The Geometry of Random Fields*. Society for Industrial and Applied Mathematics, 2010.

[2] J. Akin. *Finite Element Analysis with Error Estimators: An Introduction to the FEM and Adaptive Error Analysis for Engineering Students*. Elsevier Science, 2005.

[3] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[4] M. N. Avramova and K. N. Ivanov. Verification, validation and uncertainty quantification in multi-physics modeling for nuclear reactor design and safety analysis. *Progress in Nuclear Energy*, 52:601–614, 2010.

[5] D. Ayres, M. Eaton, A. Hagues, and M. Williams. Uncertainty quantification in neutron transport with generalized polynomial chaos using the method of characteristics. *Annals of Nuclear Energy*, 45:1428, 07 2012.

[6] J. M. Bardsley, T. Cui, Y. M. Marzouk, and Z. Wang. Scalable optimization-based sampling on function space. *SIAM Journal on Scientific Computing*, 42(2):A1317–A1347, 2020.

[7] V. F. Barrenechea, G. An unusual stabilized finite element method for a generalized stokes problem. *Numerische Mathematik*, 92:653–677, 2002.

[8] A. Barth, C. Schwab, and N. Zollinger. Multi-level monte carlo finite element method for elliptic pdes with stochastic coefficients. *Numerische Mathematik*, 119:123–161, 2011.

[9] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175:311–341, 1999.

[10] A. Beck, J. Dürrwächter, T. Kuhn, F. Meyer, C.-D. Munz, and C. Rohde. hp-Multilevel Monte Carlo Methods for Uncertainty Quantification of Compressible Flows. *ArXiv e-prints*, aug 2018.

[11] M. Bennani and T. Braconnier. Stopping criteria for eigensolvers. Technical report, 1994.

[12] P. B. Bochev, M. D. Gunzburger, and J. N. Shadid. Stability of the supg finite element method for transient advectiondiffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 193(23):2301–2323, 2004.

[13] P. Boyle, M. Broadie, and P. Glasserman. Monte carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21:1267–1321, 1997.

[14] J. H. Bramble and J. E. Osborn. Rate of convergence estimates for nonselfadjoint eigenvalue approximations. *Mathematics of Computation*, 27(123):525–549, 1973.

[15] D. Broersen and R. Stevenson. A robust petrov–galerkin discretisation of convectiondiffusion equations. *Computers & Mathematics with Applications*, 68(11):1605–1618, 2014. Minimum Residual and Least Squares Finite Element Methods.

[16] A. N. Brooks and T. J. R. 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*, 32:199–259, Sept. 1982.

[17] S. Brooks, A. Gelman, G. Jones, and X. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2011.

[18] E. G. Carnoy and M. Geradin. On the practical use of the lanczos algorithm in finite element applications to vibration and bifurcation problems. In B. Kågström and A. Ruhe, editors, *Matrix Pencils*, pages 156–176, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

[19] C. Carstensen, J. Gedicke, V. Mehrmann, and A. Miedlar. An adaptive homotopy approach for non-selfadjoint eigenvalue problems. *Numerische Mathematik*, 119:557–583, 11 2011.

[20] J. Charrier, R. Scheichl, and A. L. Teckentrup. Finite element error analysis of elliptic pdes with random coefficients and its application to multilevel monte carlo methods. *SIAM Journal on Numerical Analysis*, 51(1):322–352, 2013.

[21] P. G. Ciarlet. *The Fintie Element Method for Elliptic Problems*. North-Holland, AMsterdam, 1979.

[22] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14:3–15, 2011.

[23] A. Cohen, W. Dahmen, and G. Preprint. Adaptivity and variational stabilization for convection-diffusion equations. *European Series in Applied and Industrial Mathematics (ESAIM): Mathematical Modelling and Numerical Analysis*, 46, 09 2012.

[24] S. H. Crandall. Iterative procedures related to relaxation methods for eigenvalue problems. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 207:416–423, 1951.

[25] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11), 10 2014.

[26] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006.

[27] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous petrovgalerkin methods. part I: The transport equation. *Computer Methods in Applied Mechanics and Engineering*, 199(23):1558–1572, 2010.

[28] D. Dobson, J. Gopalakrishnan, and J. Pasciak. An efficient method for band structure calculations in 3d photonic crystals. *Journal of Computational Physics*, 161:668–679, 07 2000.

[29] T. Dodwell, C. Ketelsen, R. Scheichl, and A. Teckentrup. A Hierarchical Multilevel Markov Chain Monte Carlo Algorithm with Applications to Uncertainty Quantification in Subsurface Flow. *SIAM/ASA J. Uncertainty Quantification*, 3:1075–1108, 2014.

[30] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Finite Element Methods for Flow Problems. Wiley, 2003.

[31] I. T. Drummond, S. Duane, and R. R. Horgan. Scalar diffusion in simulated helical turbulence with molecular diffusivity. *Journal of Fluid Mechanics*, 138:7591, 1984.

[32] J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis*. John Wiley & Sons, Inc., 1976.

[33] A. Ern and J. Guermond. *Theory and Practice of Finite Elements*. Applied Mathematical Sciences. Springer New York, 2004.

[34] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, Berlin, Heidelberg, 1991.

[35] S. Giani and I. Graham. Adaptive finite element methods for computing band gaps in photonic crystals. *Numerische Mathematik*, 121, 05 2012.

[36] A. D. Gilbert, I. G. Graham, F. Y. Kuo, R. Scheichl, and I. H. Sloan. Analysis of quasi-monte carlo methods for elliptic eigenvalue problems with stochastic coefficients, 2019.

[37] A. D. Gilbert and R. Scheichl. Multilevel quasi-monte carlo for random elliptic eigenvalue problems I: Regularity and error analysis. *ArXiv*, abs/2010.01044, 2020.

[38] A. D. Gilbert and R. Scheichl. Multilevel quasi-monte carlo for random elliptic eigenvalue problems II: Efficient algorithms and numerical results. 03 2021.

[39] M. B. Giles. Multilevel monte carlo path simulation. *Oper. Res.*, 56(3):607617, May 2008.

[40] M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[41] I. Graham, M. Parkinson, and R. Scheichl. Error analysis and uncertainty quantification for the heterogeneous transport equation in slab geometry. *IMA Journal of Numerical Analysis*, 41(4):23312361, Oct. 2021.

[42] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[43] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132:767–806, 2016.

[44] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[45] G. Hauke. A simple subgrid scale stabilized method for the advection-diffusion-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 191:2925–2947, 04 2002.

[46] S. Heinrich. Multilevel monte carlo methods. In S. Margenov, J. Waśniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, pages 58–67, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[47] A. Henrot. Extremum problems for eigenvalues of elliptic operators. Birkhäuser Verlag, Basel, Switzerland, 2006.

[48] J. Heyvaerts, J. M. Lasry, M. Schatzman, and P. Witomski. Solar flares: A non linear eigenvalue problem in an unbounded domain. In C. Bardos, J. M. Lasry, and M. Schatzman, editors, *Bifurcation and Nonlinear Eigenvalue Problems*, pages 160–191, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.

[49] D. Higdon. Space and space-time modeling using process convolutions. In C. W. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi, editors, *Quantitative Methods for Current Environmental Issues*, pages 37–56, London, 2002. Springer London.

[50] T. Hughes and T. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45(1):217–284, 1984.

[51] T. J. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173–189, 1989.

[52] T. J. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. the generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58(3):305–328, 1986.

[53] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. Englewood Cliffs, N.J: Prentice-Hall, 1987.

[54] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method - a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.

[55] D. Hutton. *Fundamentals of Finite Element Analysis*. McGraw-Hill series in mechanical engineering. McGraw-Hill, 2004.

[56] E. Jamelot and P. Ciarlet. Fast non-overlapping schwarz domain decomposition methods for solving the neutron diffusion equation. *Journal of Computational Physics*, 241:445–463, 2013.

[57] A. A. Kana. *Enabling Decision Insight by Applying Monte Carlo Simulations and Eigenvalue Spectral Analysis to the Ship-Centric Markov Decision Process Framework*. PhD thesis, Univeristy of Michigan, Ann Arbor, Michigan, 2016.

[58] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[59] P. Knobloch. On the definition of the supg parameter. *Electronic Transactions on Numerical Analysis*, 32:76–89, 01 2008.

[60] R. H. Kraichnan. Diffusion by a random velocity field. *The Physics of Fluids*, 13(1):22–31, 1970.

[61] P. Lancaster. A generalised rayleigh quotient iteration for lambda-matrices. *Archive for Rational Mechanics and Analysis*, 8(1):309–322, Jan. 1961.

[62] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. B*, 45:255–282, 1950.

[63] O. P. Le Matre and O. M. Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Scientific computation. Springer, Dordrecht, 2010.

[64] R. B. Lehoucq. *Analysis and implementation of an implicitly restarted Arnoldi iteration*. PhD thesis, Rice University, Houston, Texas, May 1995. Also available as Technical Report TR95-13, Dept. of Computational and Applied Mathematics.

[65] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1998.

[66] C. Lester, C. A. Yates, M. B. Giles, and R. E. Baker. An adaptive multi-level simulation algorithm for stochastic biological systems. *Journal of Chemical Physics*, 142(2):024113, 2015.

[67] R. Lewis, P. Nithiarasu, and K. Seetharamu. *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. 01 2004.

[68] B. Q. Li. *Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer*. Computational Fluid and Solid Mechanics. Springer London, 2006.

[69] J. S. Liu. *Monte Carlo strategies in Scientific Computing*. Springer, New York, 2001.

[70] M. Loève. *Probability Theory*. Springer, New York, 1978.

[71] S. H. Lui, H. B. Keller, and T. W. C. Kwok. Homotopy method for the large, sparse, real nonsymmetric eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications*, 18(2):312–333, 1997.

[72] R. Mannella. Absorbing boundaries and optimal stopping in a stochastic differential equation. *Physics Letter A*, 254:257–262, 1999.

[73] B. P. McGrail, S. Ahmed, H. T. Schaef, A. T. Owen, P. F. Martin, and T. Zhu. Gas hydrate property measurements in porous sediments with resonant ultrasonic spectroscopy. *Journal of Geophysical Research: Solid Earth*, 112, 2007.

[74] A. Migliori. Resonant ultrasound spectroscopy. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2016.

[75] A. Migliori and J. Sarrao. Resonant ultrasound spectroscopy : applications to physics, materials measurements, and nondestructive evaluation. 1997.

[76] S. Mishra, C. Schwab, and J. Sukys. Multi-level monte carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *Journal of Computational Physics*, 231(8):3365–3388, 2012.

[77] K. W. Morton. *Numerical solution of convection-diffusion problems*, volume 12. CRC Press, 1996.

[78] M. Motamed and D. Appel. A multiorder discontinuous galerkin monte carlo method for hyperbolic problems with stochastic parameters. *SIAM Journal on Numerical Analysis*, 56(1):448–468, 2018.

[79] R. Norton and R. Scheichl. Planewave expansion methods for photonic crystal fibres. *Applied Numerical Mathematics*, 63:88104, 01 2013.

[80] O. O. Ochoa and J. N. Reddy. *Finite Element Analysis of Composite Laminates*, pages 37–109. Springer Netherlands, Dordrecht, 1992.

[81] A. M. Ostrowski. On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors. I. *Archive for Rational Mechanics and Analysis*, 1(1):233–241, Jan. 1957.

[82] C. C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. PhD thesis, University of London, London, England, 1971.

[83] J. Rayleigh. *The Theory of Sound*. Number v. 1 in The Theory of Sound. Macmillan, 1894.

[84] S. C. Reddy and L. N. Trefethen. Pseudospectra of the convection-diffusion operator. *SIAM Journal on Applied Mathematics*, 54(6):1634–1649, 1994.

[85] Y. Saad. Variations on arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and Its Applications*, 34(C):269–295, Dec. 1980.

[86] Y. Saad. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Mathematics of Computation*, 42:567–588, 1984.

[87] R. Scheichl, A. M. Stuart, and A. L. Teckentrup. Quasi-Monte Carlo and Multilevel Monte Carlo Methods for Computing Posterior Expectations in Elliptic Inverse Problems. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):493–518, 2017.

[88] R. B. Schwartz and J. F. Vuorinen. Resonant ultrasound spectroscopy: applications, current status and limitations. *Journal of Alloys and Compounds*, 310:243–250, 2000.

[89] J. A. Scott. An arnoldi code for computing selected eigenvalues of sparse, real, unsymmetric matrices. *ACM Trans. Math. Softw.*, 21:432–475, 1995.

[90] C. Soize. *Uncertainty Quantification* , volume 47 of *Interdisciplinary Applied Mathematics*. Springer, 2017.

[91] D. C. Sorensen. Implicit application of polynomial filters in a k-step arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, 1992.

[92] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, 1973.

[93] A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[94] M. Stynes. Steady-state convection-diffusion problems. *Acta Numerica*, 14:445508, 2005.

[95] D. M. Tartakovsky and S. Broyda. Pdf equations for advectivereactive transport in heterogeneous porous media with uncertain properties. *Journal of Contaminant Hydrology*, 120-121:129–140, 2011. Reactive Transport in the Subsurface: Mixing, Spreading and Reaction in Heterogeneous Media.

[96] A. Teckentrup, R. Scheichl, M. Giles, and E. Ullmann. Further analysis of multilevel monte carlo methods for elliptic pdes with random coefficients. *Numerische Mathematik*, 125, 04 2012.

[97] W. T. Thomson. *The Theory of Vibrations with Applications*. Prentice–Hall, NJ, USA, 1981.

[98] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[99] M. Williams. A method for solving stochastic eigenvalue problems ii. *Applied Mathematics and Computation*, 219:3906–3928, 02 2010.

[100] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, USA, 2010.

[101] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping With Uncertainties*. 01 2002.

[102] O. Zienkiewicz and R. Taylor. *Finite Element Method: Volume 1 - The Basis*. Butterworth-Heinemann, Oxford, 5th edition, 2000.

[103] O. Zienkiewicz and R. Taylor. *Finite Element Method: Volume 3 - Fluid Dynamics*. Butterworth-Heinemann, Oxford, 5th edition, 2000.

[104] B. Zohuri. *Neutronic Analysis For Nuclear Reactor Systems*. Springer, 2019.