On selected efficient numerical methods for multiscale problems with stochastic coefficients

Cornelia Kronsbein

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.) genehmigte Dissertation.

1. Gutachter: Prof. Dr. Oleg Iliev
2. Gutachter: Prof. Victor Ginting, Ph.D.

Datum der Disputation: 13 Dezember 2012

D 386

Acknowledgments

Firstly, I would like to thank my supervisor Professor Oleg Iliev for his support during the entire time of my PhD. Furthermore, I would like to thank Professor Yalchin Efendiev for his support and many fruitful discussions and Frédéric Legoll for the great collaboration.

Additionally, I am grateful for the pleasant working environment provided by the Fraunhofer ITWM. Thank you to all my colleagues of the department SMS.

Last but not least, I would like to thank the Fraunhofer ITWM, the TU Kaiserslautern, the "Deutsche Forschungsgemeinschaft" (DFG Project IL 55/1 − 2) and the "Innovationszentrum Applied System Modeling" for the financial support.

Contents

1. Introduction 1

2. Numerical methods for stochastic computations 2
- 2.1. Overview of the techniques 2
  - 2.1.1. Monte Carlo method 2
  - 2.1.2. Perturbation methods 2
  - 2.1.3. Moment equations 2
  - 2.1.4. Operator based methods 3
  - 2.1.5. Generalized polynomial chaos method 3
- 2.2. Setting 3
- 2.3. Generalized polynomial chaos method 3
  - 2.3.1. Univariate generalized polynomial chaos method 3
  - 2.3.2. Multivariate generalized polynomial chaos method 4
  - 2.3.3. Generalized polynomial chaos approximation 5
  - 2.3.4. Multi-element basis 5
  - 2.3.5. Statistical information 5
- 2.4. Stochastic Galerkin method 6
- 2.5. Stochastic collocation methods 6
  - 2.5.1. Lagrange interpolation approach 6
  - 2.5.2. Pseudo-spectral gPC approach 7
- 2.6. Comparison of Galerkin and collocation methods 8
- 2.7. Random domain problem 9

3. Stationary diffusion equation with a random coefficient 9

4. Homogenization 10
- 4.1. Homogenization for a deterministic diffusion problem 10
  - 4.1.1. Formal asymptotic expansion 10
- 4.2. Stochastic homogenization 12
- 4.3. Numerical homogenization 14
- 4.4. Estimation of the rate of convergence 15

I. Karhunen-Loève expansion for numerical homogenization 17

5. Diffusion with a stochastic coefficient 17
- 5.1. Monte Carlo simulation 18
  - 5.1.1. Cell problem 18
- 5.2. Karhunen-Loève expansion 18
  - 5.2.1. Discrete Karhunen-Loève expansion 18
  - 5.2.2. Cell problem 19
- 5.3. Hierarchical matrix approximation of the covariance matrix 27
  - 5.3.1. Cluster theory 28
  - 5.3.2. Geometric bisection 29
  - 5.3.3. Regular subdivision 30
  - 5.3.4. Block cluster tree 30
  - 5.3.5. Admissibility 31
  - 5.3.6. Low-rank approximation 31

6. **Statement of the problems** 33

7. **Cell-centered finite volume method** 35
  7.1. Scalar coefficient 35
  7.2. Matrix coefficient 36

8. **Generation of random variables** 40

9. **Numerical results** 40
  9.1. Finite volume test 40
  9.2. Deterministic equations 41
  9.3. Stochastic equations 41
    9.3.1. Gaussian distributed random variables 42
    9.3.2. Lognormal distributed random variables 44
  9.4. H-matrix results 44

II. Multi-level Monte Carlo for numerical homogenization 47

10. **Preliminaries** 48
  10.1. Multi-level Monte Carlo method 48
  10.2. Remark on same or independent samples 49
  10.3. Definition of meshes and representative volume sizes 51

11. **Multi-level Monte Carlo for the upscaled coefficients and their properties** 52
  11.1. Various RVE sizes and fixed fine mesh 52
  11.2. Coarse and fine meshes 57

12. **Weighted multi-level Monte Carlo** 59

13. **Multi-level Monte Carlo for the homogenized solution** 60
  13.1. Standard MLMC 60
  13.2. Separable case 61
  13.3. General case. Application of weighted MLMC 64

14. **Numerical results** 69
  14.1. Numerical results of the homogenized coefficient 70
    14.1.1. Numerical study of the convergence rate 70
    14.1.2. One dimensional examples 72
    14.1.3. Two dimensional example 82
  14.2. Numerical results of the homogenized solution 87
    14.2.1. One dimensional example 87
    14.2.2. Two dimensional example 88

III. Multi-level Monte Carlo with two-phase flow and transport 93

15. **Preliminaries** 93
  15.1. Physical quantities 93
  15.2. Two-phase flow and transport model 94

15.3. Mixed multiscale finite element methods 97
    15.3.1. Mixed MsFEM using local boundary conditions 98
    15.3.2. Mixed MsFEM using global boundary conditions 99
15.4. Implicit scheme for solving the saturation equation 99

16. Ensemble level methods for mixed multiscale finite element methods 100
16.1. No-local-solve-online ensemble level multiscale method (NLSO) 101
16.2. Local-solve-online ensemble level multiscale method (LSO) 102
16.3. Comparison of LSO and NLSO approaches 102

17. Multi-level Monte Carlo using ensemble level mixed multiscale FEMs 103
17.1. Computational cost 105

18. Experimental setup 106

19. Numerical results for single-phase flow 108
19.1. Comparison of the NLSO and the LSO approach 108
19.2. Numerical study of the convergence rate 111
19.3. Ensemble level mixed MsFEM 111

20. Numerical results for two-phase flow 119
20.1. Comparison of the NLSO and the LSO approach 119
20.2. Numerical study of the convergence rate 121
20.3. Ensemble level mixed MsFEM 123

21. Conclusions 129

A. Notation 131
A.1. Notation Part I 133
A.2. Notation Part II 135
A.3. Notation Part III 137

B. Tables 141

1. Introduction

Multiple spatial scales occur in most real life problems (e.g., groundwater flow, filtration processes). A direct numerical simulation of these problems is difficult, since one has to deal with huge problems if one resolves the finest scale. In order to overcome these difficulties the idea is to approximate the solutions on a coarser grid with the help of homogenization or multiscale methods. In the homogenization approach effective properties of the media -e.g., the effective permeability of the medium- which contain small scale features are constructed in each coarse grid block. In a multiscale method the fine scale features can come into play due to an appropriate choice of the basis functions. These approaches reduce the computational costs.

Another difficulty arises since these multiscale problems are not purely deterministic. For example, the properties of the soil are known on some snapshots only, where samples of the soil have been pulled by drilling. Due to this incomplete knowledge stochasticity has to be taken into account which increases the complexity of the considered problems. A standard approach to deal with uncertainties is the Monte Carlo method [32]. Here one solves the problems for many realizations of the stochastic quantity -e.g., the permeability. For each realization the stochastic problem reduces to a deterministic one, where the techniques for deterministic equations can be applied. Another popular approach is the polynomial chaos method [56]. In this approach one tries to approximate a random function by orthogonal polynomials in a finite dimensional random space.

Our objective is to compute the expectation of some functionals of the effective coefficient or of the macroscopic solution. The goal is to combine approaches to construct effective properties or multiscale basis functions with stochastic algorithms to increase the accuracy in comparison to the Monte Carlo method by keeping the numerical costs fixed.

As a first step we focus on numerical homogenization. In this case we consider a stationary diffusion equation with a stochastic coefficient. Here we deal with the uncertainties with the Karhunen-Loève expansion in combination with a polynomial chaos expansion or a multi-level Monte Carlo method. Secondly, we use mixed multiscale finite element methods to cope with the hierarchy of spatial scales and a multi-level Monte Carlo method to handle the stochasticity. We apply it for multi-phase flow and transport equations.

The work is organized as follows. After an overview of numerical methods for stochastic computations (cf. Section 2) we state a stationary diffusion problem with a stochastic coefficient (cf. Section 3) which we consider in Part I and Part II and introduce the homogenization theory (cf. Section 4).

In Part I we combine numerical homogenization with the Karhunen-Loève expansion in combination with a polynomial chaos expansion. With the help of the Karhunen-Loève expansion in combination with a polynomial chaos expansion we reduce the stochastic problem to a high dimensional deterministic one. By choosing the basis of the stochastic space appropriately we are able to decouple the high dimensional problem to many d-dimensional deterministic problems, where d denotes the space dimension. This idea was introduced in [33]. We combine this idea with homogenization and construct an approximation of the expectation of the effective coefficient with solutions of the decoupled deterministic equations. To construct the Karhunen-Loève expansion of a random field it is essential to compute the eigenpairs of a given covariance operator. In general this results in a large problem. In Section 5.3 we use hierarchical matrices (cf. [37]) to approximate the eigenpairs similar to the approach in [44]. So we can reduce the computational time. To solve the equations we use a cell-centered finite volume method, which we describe in Section 7. We close the first Part and as well the other parts with numerical results.

In Part II we use a multi-level Monte Carlo approach (cf. [38]) to deal with the randomness of the problem. We apply it for both to approximate the expectation of the effective coefficient (cf.

Section 11) and of the homogenized coarse solution (cf. Section 13). To approximate the solution we introduce a weighted multi-level Monte Carlo method in Section 12. The idea of the multi-level Monte Carlo approach is to consider the quantity of interest -in our case the effective coefficient or the coarse solution- on different levels. As levels we consider different coarse mesh and domain sizes. The largest level (e.g., the finest mesh or the largest computational domain) is the level where we want to approximate the expectation of the quantity of interest. It is the computationally most expensive and accurate one. All levels are used to approximate the expectation of the quantity of interest at the largest level. In multi-level Monte Carlo methods the choice of realizations per level is essential. One straightforward condition is that the number of realizations must decrease if the level increases to solve only a few problems if they are computationally expensive and compensate the high stochastic error with many problem solves in less accurate cases (lower levels).

In Part III we apply the in Part II introduced multi-level Monte Carlo method to multi-phase flow and transport problems (cf. Section 15.2). In this case we are interested in the expectation of the water saturation. To solve the system of equations we use a mixed multiscale finite element method (cf.[29]) for the pressure equation and a standard implicit scheme to solve for the saturation (cf. Section 15.4). To deal with the uncertainties -here we consider a random permeability- we use ensemble level mixed multiscale finite element approaches (cf. Section 16). Here a different level denotes a different velocity approximation space in the mixed multiscale finite element method. The higher the level the more accurate is the approximation space.

2. Numerical methods for stochastic computations

In the following section we present an overview of numerical methods for stochastic computations based on the review of D. Xiu (cf. [54]). First we give a short introduction on recently available techniques. Later on we have a closer look on the generalized polynomial chaos method.

2.1. Overview of the techniques

2.1.1. Monte Carlo method

One very popular method is the Monte Carlo method. Here one generates realizations of random inputs on their prescribed probability distribution. For each realization the data is fixed and the problem becomes deterministic. Statistical information, such as the mean of the solution, can be extracted from a collected ensemble of solutions. The method is straightforward to apply, but typically a large number of realization is needed.

2.1.2. Perturbation methods

In the most commonly used non-sampling method random fields are expanded via Taylor series around their mean and truncated at a certain order. The resulting system of equations becomes extremely complicated. Therefore typically at most a second-order expansion is employed. A limitation of this method is that the magnitude of uncertainties cannot be large (less than 10%).

2.1.3. Moment equations

Here one attempts to compute the moments of the random solution directly. The moments are the unknowns of the equations which are derived by averaging the original stochastic equations. Problems can arise, because the deviation of a moment almost always requires the information of higher moments.

2.1.4. Operator based methods

This approach is based on manipulation of the stochastic operators in the considered equations. However, like the perturbation method this method is restricted to small uncertainties.

2.1.5. Generalized polynomial chaos method

A generalization of the classical polynomial chaos method has become one of the most popular methods. Here stochastic solutions are expressed as orthogonal polynomials of the random parameters. The convergence depends on the choice of type of orthogonal polynomials. In the classical polynomial chaos Hermite polynomials are used. The coefficients of the generalized polynomial chaos expansion are the unknown quantities.

2.2. Setting

In this section we present a more general setting in which we give an introduction to the generalized polynomial chaos method. We consider the following partial differential equations

$$\begin{aligned} L(x,u;W) &= 0, \quad \text{in } D, \\ B(x,u;W) &= 0, \quad \text{on } \partial D, \end{aligned} \tag{2.1}$$

where L is a differential operator and B is a boundary operator. On Dirichlet segments B is the identity operator, for example. $W = (W_1, \dots, W_N)$ is a N-variate random vector with independent components in a properly defined probability space (Ω, F, P). Let $\tilde{\rho}_i : \Upsilon_i \mapsto \mathbb{R}^+$ be the probability function of the random variable $W_i(\omega)$, $\omega \in \Omega$, whose range is $\Upsilon_i = W_i(\Omega) \subseteq \mathbb{R}$ for $i = 1, \dots, N$. Then $\tilde{\rho}(W) = \prod_{i=1}^N \tilde{\rho}_i(W_i)$ is the joint probability density of the random vector $W = (W_1, \dots, W_N)$ with the support $\Upsilon = \prod_{i=1}^N \Upsilon_i \subseteq \mathbb{R}$. So we can replace the infinite space Ω and seek a solution $u(x,W) : \bar{D} \times \Upsilon \mapsto \mathbb{R}$. Thus the key issue is to parameterize the input uncertainty by a set of finite number random variables.

2.3. Generalized polynomial chaos method

With a generalized polynomial chaos (gPC) expansion one tries to approximate a random function by orthogonal polynomials in the finite dimensional random space Υ.

2.3.1. Univariate generalized polynomial chaos method

First we consider the one-dimensional case. We define the one-dimensional orthogonal polynomial space w.r.t. the measure $\tilde{\rho}_i(W_i)\, dW$ in Υ_i via

$$\mathcal{W}^{i,r_i} = \{v : \Upsilon_i \mapsto \mathbb{R} : \, v \in \operatorname{span}\{\theta_m(W_i)\}_{m=0}^{r_i}\}, \quad i = 1, \dots, N, \tag{2.2}$$

where $\{\theta_m(W_i)\}$ denotes a set of orthogonal polynomials with

$$\int_{\Upsilon_i} \tilde{\rho}_i(W_i)\theta_m(W_i)\theta_n(W_i)\, dW_i = H_m^2 \delta_{mn}$$

and

$$H_m^2 = \int_{\Upsilon_i} \tilde{\rho}_i(W_i)\theta_m^2(W_i)\, dW_i.$$

In the following we assume, that the polynomials are normalized by an appropriate scaling, i.e., we have $H_m^2 = 1$ for all m. The scaling depends on the probability density function $\tilde{\rho}_i$. For example

	Distribution	**GPC Basis Polynomials**	**Support**
Continuous	Gaussian	Hermite	$(-\infty, \infty)$
	Gamma	Laguerre	$[0, \infty)$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	$[a, b]$
Discrete	Poisson	Charlier	$\{0, 1, 2, \dots\}$
	Binomial	Krawtchouk	$\{0, 1, \dots, N\}$
	Negative Binomial	Meixner	$\{0, 1, 2, \dots\}$
	Hypergeometric	Hahn	$\{0, 1, \dots, N\}$

Table 1: Correspondence between the probability distribution and the type of gPC polynomial basis.

for Gaussian distributed random variable W_i we have Hermite polynomials as basis which is the classical polynomial chaos method. In Table 1 we present different probability distributions with their corresponding gPC basis polynomials.

2.3.2. Multivariate generalized polynomial chaos method

We define the corresponding N-variate orthogonal polynomial space of total degree at most R in Υ as a tensor product of the one-dimensional spaces (2.2), i.e.,

$$\mathcal{W}_N^R = \bigotimes_{|\mathbf{r}| \leq R} \mathcal{W}^{i, r_i},$$

where the tensor product is over all possible combinations of the multi index $\mathbf{r} = (r_1, \dots, r_N) \in \mathbb{N}_0^N$. We denote the N-variate orthonormal polynomials from $\mathcal{W}_N^R$ with $\{\Theta_m(W)\}$, they are constructed as products of a sequence of one-dimensional orthonormal polynomials in each direction, i.e.,

$$\Theta_m(W) = \theta_{m_1}(W_1) \cdots \theta_{m_N}(W_n), \quad m_1 + \cdots + m_n \leq R,$$

where m_i denotes the order of the univariate polynomials of $\theta(W_i)$ in the W_i direction for $1 \leq i \leq N$. The number of basis functions is

$$\dim\left(\mathcal{W}_N^R\right) = \binom{N+R}{N}.$$

From the ortho-normality conditions of the univariate polynomials we get for the expected value

$$E[\Theta_m(W)\Theta_n(W)] = \int \Theta_m(W)\Theta_n(W)\tilde{\rho}(W)\, dW = \delta_{mn}, \quad \forall 1 \leq m, n \leq \dim\left(\mathcal{W}_N^R\right).$$

2.3.3. Generalized polynomial chaos approximation

The R^{th}-order gPC approximation of the solution $u(x,W)$ of (2.1) can be obtained by projection u onto the space $\mathcal{W}_N^R$, i.e., $\forall x \in D$

$$\mathbb{P}_N^R u := u_N^R(x,W) = \sum_{m=1}^{M} \hat{u}_m(x)\Theta_m(W), \quad M = \binom{N+R}{N}, \tag{2.3}$$

where $\mathbb{P}_n^R u$ denotes the orthogonal projection operator from $L^2_{\tilde{\rho}}(\Upsilon)$ onto $\mathcal{W}_N^R$ and $\{\hat{u}_m\}$ are the Fourier coefficients defined as

$$\hat{u}_m := \int u(x,W)\Theta_m(W)\tilde{\rho}(W)\,dW = E[u(x,W)\Theta_m(W)], \quad 1 \leq m \leq M. \tag{2.4}$$

From classical approximation arguments follows that $\mathbb{P}_N^R u$ is the best-approximation in $\mathcal{P}_N^R$, the linear polynomial space of N-variate polynomials of degree up to R, i.e., for any $x \in D$ and $u \in L^2_{\tilde{\rho}}(\Upsilon)$

$$\| u - \mathbb{P}_N^R u \|_{L^2_{\tilde{\rho}}(\Upsilon)} = \inf_{\tilde{\Theta} \in \mathcal{P}_N^R} \| u - \tilde{\Theta} \|_{L^2_{\tilde{\rho}}(\Upsilon)} .$$

The error of this finite-order projection we define as

$$\epsilon_G(x) := \| u - \mathbb{P}_N^R u \|_{L^2_{\tilde{\rho}}(\Upsilon)} = \Big(E\big[\big(u(x,W) - u_N^R(x,W)\big)^2\big]\Big)^{\frac{1}{2}} \quad \forall x \in D.$$

It will converge to zero as $R \to \infty$.

2.3.4. Multi-element basis

The basis does not need to be globally smooth. In fact gPC basis of piecewise polynomials should be used to avoid accuracy loss, if the stochastic solution exhibits discontinuities in random space.

2.3.5. Statistical information

We can obtain statistical information of the solution from the gPC approximation. For example, the mean solution is

$$\begin{aligned}
E[u] &\approx E[u_N^R] \\
&= \int \left(\sum_{m=1}^{M} \hat{u}_m \Theta_m(W) \right) \tilde{\rho}(W)\,dW \\
&= \hat{u}_1 \underbrace{\int \Theta_1(W)\tilde{\rho}(W)\,dW}_{=1,\ (\Theta_1 \equiv 1, \tilde{\rho}\ density)} + \sum_{m=2}^{M} \hat{u}_m \underbrace{\int \Theta_m(W)\tilde{\rho}(W)\,dW}_{=0} \\
&= \hat{u}_1
\end{aligned}$$

and for the covariance function we get

$$\begin{aligned}
\mathrm{cov}_u(x_1,x_2) &= E[(u(x_1,W) - E[u(x_1,W)])\,(u(x_2,W) - E[u(x_2,W)])] \\
&\approx E[\big(u_N^R(x_1,W) - E[u_N^R(x_1,W)]\big)\,\big(u_N^R(x_2,W) - E[u_N^R(x_2,W)]\big)] \\
&= E[u_N^R(x_1,W)u_N^R(x_2,W)] - E[u(x_1,W)]E[u(x_2,W)]
\end{aligned}$$

$$= \sum_{m,n=1}^{M} \hat{u}_m(x_1)\hat{u}_n(x_2) \underbrace{\int \Theta_m(W)\Theta_n(W)\tilde{\rho}(W)\,dW}_{=\delta_{mn}} - \hat{u}_1(x_1)\hat{u}_1(x_2)$$

$$= \sum_{m=2}^{M} \hat{u}_m(x_1)\hat{u}_m(x_2).$$

Therefore the variance can be approximated as

$$\mathrm{Var}(u(x)) = E[(u(x,W) - E[u(x,W)])]^2 \approx \sum_{m=2}^{M} \hat{u}_m^2(x).$$

2.4. Stochastic Galerkin method

The mean issue in using the gPC approximation (2.3) is to determine the coefficient $\hat{u}_m$ of the expansion. Thereby the definition via Fourier coefficients (2.4) is not helpful, since it requires the exact unknown solution $u(x,W)$. Consequently, we need an alternative way to estimate the coefficients. A typical approach is to employ a stochastic Galerkin approach. Here we seek an approximate gPC solution in the form of (2.3). The coefficients $\{\hat{u}_m\}$ are obtained by satisfying the following weak form, for all $v \in \mathcal{W}_N^R$,

$$\int L(x, u_N^R; W)v(W)\tilde{\rho}(W)\,dW = 0, \quad \text{in } D,$$

$$\int B(x, u_N^R; W)v(W)\tilde{\rho}(W)\,dW = 0, \quad \text{on } \partial D.$$

This is a set of coupled deterministic PDEs for $\{\hat{u}_m\}$, and standard numerical techniques can be applied. However one should keep in mind that if the equation (2.1) takes a complicated form, the derivation of Galerkin equations for the coefficients can become highly nontrivial, sometimes impossible.

2.5. Stochastic collocation methods

In collocation methods one seeks to satisfy the equations (2.1) at a discrete set of points in the corresponding random space. These points are called “nodes”. The selection of nodes is the key ingredient to all stochastic collocation methods. If the space is one-dimensional the optimal choice is usually the Gauss quadrature. The challenge is in multi-dimensional spaces, especially for large dimensions. A natural generalization is to choose tensor products of the one-dimensional nodes. However, for larger dimensions this results in a huge number of nodes. Therefore one uses this approach mostly at lower dimensions, e.g., $N \leq 5$.

Another ansatz is based on Smolyaks algorithm. It uses only a subset of the full tensor product grids -a sparse grid. This sparse grid consists of much smaller number of nodes. The subset is chosen strategically in such a way that the approximation properties for $N = 1$ are preserved for $N > 1$ as good as possible.

Two of the major approaches of high-order stochastic collocation methods are the Lagrange interpolation approach and the pseudo-spectral gPC approach.

2.5.1. Lagrange interpolation approach

With $\Xi_N = \{W^{(i)}\}_{i=1}^{Q} \in \Upsilon$ we denote a set of prescribed nodes in the N dimensional random space Υ and with Q the number of nodes. Let $L_i(W^{(j)}) = \delta_{ij}$, $1 \leq i,j \leq Q$ be the Lagrange polynomials and

$$\tilde{u}_k = u(x, W^{(k)}), \quad 1 \le k \le Q,$$

then we can write a Lagrange interpolation of the solution $u(x,W)$ of (2.1) as

$$\mathcal{I}u(x,W) = \sum_{k=1}^{Q} \tilde{u}_k(x) L_k(W), \quad \forall x \in D.$$

By requiring the equation (2.1) to be satisfied at each of the nodes, we obtain: $\forall k = 1, \dots, Q$

$$\begin{aligned} L(x, \tilde{u}_k; W^{(k)}) &= 0, \quad \text{in } D \\ B(x, \tilde{u}_k; W^{(k)}) &= 0, \quad \text{on } \partial D \end{aligned}$$

Thus we have to solve Q deterministic problems with realizations of the random vector. There one can apply existing deterministic solvers in comparison with the stochastic Galerkin method, where the resulting equations are generally coupled. Again we obtain statistics of the solution, e.g.,

$$E[u(x,W)] \approx E[\mathcal{I}u(x,W)] = \sum_{k=1}^{Q} \tilde{u}_k(x) \int L_k(W) \tilde{\rho}(W)\, dW.$$

The integrals $\int L_k(W)\tilde{\rho}(W)\, dW$ correspond to weights in the discrete sum. In multi-dimensional random spaces these weights are not readily available and the formula is of little use.

2.5.2. Pseudo-spectral gPC approach

In this approach we seek an approximate solution of (2.1) in the form of gPC expansion, i.e., for any $x \in D$

$$\mathbb{I}_N^R u := v_N^R = \sum_{m=1}^{M} \hat{v}_m(x) \Theta_m(W), \quad M = \binom{N+R}{N} \tag{2.5}$$

where $\mathbb{I}_N^R u$ is another projector from $L^2_{\tilde{\rho}}$ to $\mathcal{W}_N^R$ and the expansion coefficients are determined as

$$\hat{v}_m(x) = \sum_{j=1}^{Q} u(x, W^{(j)}) \Theta_m(W^{(j)}) \alpha^{(j)}, \quad m = 1, \dots, M$$

where $\{W^{(j)}, \alpha^{(j)}\}_{j=1}^{Q}$ are a set of nodes and weights. These nodes and weights should be chosen in such a way that

$$\sum_{j=1}^{Q} f(W^{(j)}) \alpha^{(j)} \approx \int f(W) \tilde{\rho}(W)\, dW = E[f(W)] \tag{2.6}$$

for sufficiently smooth functions $f(W)$. Hence it follows

$$\hat{v}_m(x) = \sum_{j=1}^{Q} u(x, W^{(j)}) \Theta_m(W^{(j)}) \alpha^{(j)} \approx E[u(x,W)\Theta_m(W)] = \hat{u}_m(x)$$

and consequently $\mathbb{I}_N^R u$ of (2.5) becomes an approximation of the exact gPC expansion $\mathbb{P}_N^R u$ of (2.3). The arising error

$$\epsilon_Q := \|\mathbb{I}_N^R u - \mathbb{P}_N^R u\|_{L^2_{\tilde{\rho}}(\Upsilon)} = \left(E[\left(\mathbb{I}_N^R - \mathbb{P}_N^R\right) u]^2\right)^{\frac{1}{2}}$$

	Galerkin Method	**Collocation Method**
Pros	the gPC method offers the most accurate solutions involving least number of equations in multi-dimensional random spaces	requires only repetitive executions of existing deterministic solvers
		applicability of stochastic collocation is not affected by the complexity or nonlinearity of the original problem, as long as one can develop a reliable deterministic solver
Cons	it is relatively more cumbersome to implement	the aliasing error can be significant, especially for higher dimensional random spaces
	when the original problem takes highly complex form, the explicit derivation of the gPC equations may not be possible	all of the existing collocation methods require solutions of (much) larger numbers of equations than that of gPC Galerkin, especially for higher dimensional random spaces
	resulting system of equations is generally coupled	

Table 2: Advantages and drawbacks of the stochastic collocation method and the stochastic Galerkin method.

we call “aliasing error”. It is caused by the integration error from (2.6). This error can become a significant source of errors in multi-dimensional random spaces.
The pseudo-spectral gPC method also requires only repetitive deterministic solutions with fixed realizations of the random inputs. In comparison to the gPC Galerkin method the evaluation of the approximate gPC expansion coefficients is completely independent. This approach has not got the drawback of unavailable weights like the Lagrange interpolation approach mentioned before.

2.6. Comparison of Galerkin and collocation methods

The question arises which method one should choose. In Table 2 we list advantages and drawbacks of the stochastic collocation method and the stochastic Galerkin method. When a single deterministic computation is already time consuming the gPC Galerkin method should be preferred (because of the smaller number of equations), if either

1. efficient solvers can be developed to decouple the gPC system effectively, or
2. the coupling of gPC Galerkin equations does not incur much additional computational costs.

2.7. Random domain problem

Up to now we have assumed a deterministic domain D. However, in practice it can be a source of uncertainty as in many applications the physical domain cannot be determined precisely. Here we consider an uncertain domain and the problem can be formulated as

$$\begin{aligned} L(x,u) &= 0, \quad \text{in } D(W) \\ B(x,u) &= 0, \quad \text{in } \partial D(W). \end{aligned} \tag{2.7}$$

For simplicity we assume the only source of uncertainty to be in the definition of the boundary. The idea is to use an one-to-one mapping to transform the random domain into a deterministic one. Let

$$\zeta = \zeta(x,W), \quad x = x(\zeta,W), \quad \forall\, W \in \Upsilon,$$

be this mapping and its inverse, such that the random domain $D(W)$ is transformed into a deterministic domain $D_{det} \subseteq \mathbb{R}^d$ with the coordinates $\zeta = (\zeta_1, \ldots, \zeta_d)$. Then (2.7) is tranformed into the following problem: for all $W \in \Upsilon$, find $u = u(\zeta, W); \bar{D}_{det} \times \Upsilon \mapsto \mathbb{R}$ such that

$$\begin{aligned} \mathcal{L}(\zeta,u;W) &= 0, \quad \text{in } D_{det} \\ \mathcal{B}(\zeta,u;W) &= 0, \quad \text{in } \partial D_{det}, \end{aligned} \tag{2.8}$$

where the operators L and B are transformed to $\mathcal{L}$ and $\mathcal{B}$, respectively. The problem (2.8) is a stochastic PDE in a fixed domain and we can apply all the techniques mentioned before.

3. Stationary diffusion equation with a random coefficient

We will consider a case with random coefficients that become de-correlated in distances much larger than ϵ and give the estimate for the first order corrector.

Consider the following elliptic problem

$$\begin{aligned} -\text{div}\,(K_\epsilon(x,\omega)\nabla u_\epsilon) &= f \quad \text{in} \quad D \\ u_\epsilon &= g \quad \text{on} \quad \partial D \end{aligned} \tag{3.1}$$

where $K_\epsilon(x,\omega)$ is a homogeneous random field, f and g are non-random functions and ϵ is a small parameter. We define, for all $\omega \in \Omega$,

$$k_{\min}(\omega) = \min_{x \in \bar{D}} K_\epsilon(x,\omega) \quad \text{and} \quad k_{\max}(\omega) = \max_{x \in \bar{D}} K_\epsilon(x,\omega).$$

Here we denote as in [18] with $\mathcal{C}^t(\bar{D})$ the space of Hölder continuous functions and with $H^{-t}(D)$ the dual space of the Sobolev space $H^t_0(D)$, for $0 < t \leq 1$. We assume $K_\epsilon(x,\omega) \in L^p(\Omega, \mathcal{C}^t(\bar{D}))$, for some $0 < t < 1$ and for all $p \in (0,\infty)$, $g \in H^1(D)$ and $f \in H^{t-1}(D)$. Furthermore we assume $k_{\min} > 0$ almost surely and $k^{-1}_{\min}, k_{\max} \in L^p(\Omega)$, for all $p \in (0,\infty)$. Then (3.1) has a unique solution u_ϵ, which belongs to $L^p(\Omega, H^1_0(D))$, for all p (cf.[18]). Note that the assumptions on the coefficient are fulfilled for a log-normal or Gaussian random field.

4. Homogenization

4.1. Homogenization for a deterministic diffusion problem

Let D be a bounded domain in $\mathbb{R}^d$ with smooth boundary ∂D. Then a stationary diffusion problem reads

$$\begin{aligned} -\operatorname{div}\left(K\left(\frac{x}{\epsilon}\right)\nabla u_\epsilon\right) &= f(x), \quad x \in D \\ u_\epsilon &= g(x), \quad x \in \partial D \end{aligned} \tag{4.1}$$

where the coercive and bounded coefficient K is Y-periodic in $\mathbb{R}^d$ with periodicity cell

$$Y = \{y = (y_1, \ldots, y_d) : 0 < y_i < 1 \text{ for } i = 1, \ldots, d\}$$

and ϵ a scale parameter. If ϵ decreases, the oscillation rate of the coefficient increases. Thus it is natural to ask about the behavior of the solution u_ϵ if $\epsilon \to 0$.

4.1.1. Formal asymptotic expansion

To derive the limit problem in a formal way, one starts from the ansatz (cf.[26, 42]) that the unknown function u_ϵ has an asymptotic expansion with respect to ϵ of the form

$$u_\epsilon(x) = \sum_{i=0}^{\infty} \epsilon^i u_i(x, y)$$

where the coefficient functions $u_i(x, y)$ are Y-periodic with respect to the variable $y = \frac{x}{\epsilon}$. The derivative obeys the law

$$\operatorname{div} = \operatorname{div}_x + \frac{1}{\epsilon}\operatorname{div}_y,$$

where the subscripts denote the partial derivatives with respect to x and y, respectively. If we plug in the asymptotic expansion in equation (4.1) and use the law above, we get the formula

$$\begin{aligned} &-\epsilon^{-2}\operatorname{div}_y\left(K(y)\nabla_y u_0(x, y)\right) \\ &-\epsilon^{-1}[\operatorname{div}_y\left(K(y)[\nabla_y u_1(x, y) + \nabla_x u_0(x, y)]\right) + \operatorname{div}_x\left(K(y)\nabla_y u_0(x, y)\right)] \\ &-\epsilon^{0}\ [\operatorname{div}_y\left(K(y)[\nabla_y u_2(x, y) + \nabla_x u_1(x, y)]\right) + \operatorname{div}_x\left(K(y)[\nabla_y u_1(x, y) + \nabla_x u_0(x, y)]\right)] \\ &-\sum_{i=1}^{\infty}\epsilon^i[\ \operatorname{div}_y\left(K(y)[\nabla_y u_{i+2}(x, y) + \nabla_x u_{i+1}(x, y)]\right) \\ &\qquad\qquad +\operatorname{div}_x\left(K(y)[\nabla_y u_{i+1}(x, y) + \nabla_x u_i(x, y)]\right)] \\ &= f(x) \end{aligned}$$

The next step consists of comparing the coefficients of the different ϵ powers on both sides of this equation. The term with ϵ^{-2} gives

$$\operatorname{div}_y\left(K(y)\nabla_y u_0(x, y)\right) = 0 \text{ for } (x, y) \in D \times Y.$$

Because of the Y-periodicity the weak formulation with $u_0(x, y)$ as test function reads as follows

$$\int_Y K(y)\nabla_y u_0(x, y) \cdot \nabla_y u_0(x, y)\, dy = 0.$$

With the coercivity of the coefficient $K(y)$ we get

$$\|\nabla_y u_0(x,y)\|^2_{L^2(Y)} \leq 0$$

and therefore $u_0(x,y) = u_0(x)$ is a function of x alone, independent of y. Using this and the term with ϵ^{-1}, we obtain

$$\mathrm{div}_y\left(K(y)\nabla_y u_1(x,y)\right) = -\mathrm{div}_y\left(K(y)\nabla_x u_0(x)\right) \text{ for } y \in Y.$$

If we use the identity

$$\nabla_x u_0(x) = \sum_{j=1}^{d} e_j \frac{\partial u_0}{\partial x_j}(x)$$

we can write

$$\mathrm{div}_y\left(K(y)\nabla_y u_1(x,y)\right) = -\sum_{j=1}^{d} \frac{\partial K}{\partial y_j}(y)\frac{\partial u_0}{\partial x_j}(x) \text{ for } y \in Y.$$

Now, for $j = 1,\ldots,d$, we introduce the cell problem with the Y-periodic solution χ_j

$$-\,\mathrm{div}_y\left(K(y)(e_j + \nabla_y \chi_j(y))\right) = 0 \quad \text{in } Y. \tag{4.2}$$

Using these functions $\chi_j(y)$, we find

$$u_1(x,y) = \sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\chi_j(y) + u_1(x),$$

where $u_1(x)$ is some function independent of y, since

$$\nabla_y u_1(x,y) = \sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\nabla_y \chi_j(y) \tag{4.3}$$

and

$$\begin{aligned}
\mathrm{div}_y\left(K(y)\nabla_y u_1(x,y)\right) &= \mathrm{div}_y\left(K(y)\sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\nabla_y \chi_j(y)\right) \\
&= \sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\mathrm{div}_y\left(K(y)\nabla_y \chi_j(y)\right) \\
&\overset{eq.(4.2)}{=} -\sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\mathrm{div}_y\left(K(y)e_j\right) \\
&= -\sum_{j=1}^{d} \frac{\partial u_0}{\partial x_j}(x)\frac{\partial K}{\partial y_j}(y).
\end{aligned}$$

We proceed with the ϵ^0 term in the above equation. We get

$$-\mathrm{div}_y\left(K(y)[\nabla_y u_2(x,y) + \nabla_x u_1(x,y)]\right) - \mathrm{div}_x\left(K(y)[\nabla_y u_1(x,y) + \nabla_x u_0(x)]\right) = f(x)$$

for $y \in Y$, which is an equation for u_2. After integration over Y we obtain

$$\begin{aligned}&-\int_Y \mathrm{div}_y \left(K(y)[\nabla_y u_2(x,y) + \nabla_x u_1(x,y)]\right)\, dy\\ &-\int_Y K(y)\mathrm{div}_x \nabla_y u_1(x,y)\, dy - \left(\int_Y K(y)\, dy\right) \Delta_x u_0(x) = f(x)\end{aligned}$$

because the volume over Y is 1. The divergence theorem applied to the first integral leads to

$$\begin{aligned}&\int_Y \mathrm{div} \left(K(y)[\nabla_y u_2(x,y) + \nabla_x u_1(x,y)]\right)\, dy\\ =\ &\int_{\partial Y} \nu \cdot \left(K(y)\nabla_y u_2(x,y) + K(y)\nabla_x u_1(x,y)\right)\, d\Gamma(y)\end{aligned}$$

where ν is the normal vector on ∂Y. This boundary integral vanishes because of the Y-periodicity of the functions $K(y)$, $u_1(x,y)$ and $u_2(x,y)$. For the second term we use equation (4.3) and get

$$\mathrm{div}_x \nabla_y u_1(x,y) = \sum_{i,j=1}^{d} \frac{\partial \chi_j}{\partial y_i}(y) \frac{\partial^2 u_0(x)}{\partial x_i \partial x_j}(x).$$

In this way it follows

$$-\sum_{i,j=1}^{d} \int_Y K(y) \frac{\partial \chi_j}{\partial y_i}(y) \frac{\partial^2 u_0}{\partial x_i \partial x_j}(x) - \left(\int_Y K(y)\, dy\right) \Delta_x u_0(x) = f(x).$$

With the abbreviation

$$K^*_{ij} = \int_Y K(y) \left(\delta_{ij} + \frac{\partial \chi_j}{\partial y_i}(y)\right)\, dy$$

we get the final result

$$-\sum_{ij}^{d} K^*_{ij} \frac{\partial^2 u_0}{\partial x_i \partial x_j}(x) = f(x).$$

This elliptic differential equation is the homogenized limit of the equation (4.1). We simplify the notation to

$$-\sum_{ij}^{d} K^*_{ij} \partial_{ij} u^*(x) = f(x).$$

4.2. Stochastic homogenization

Here we introduce stochastic homogenization in the classical stationary ergodic setting (cf. [16] and the references there). First we recall the concept of a homogeneous field. With (Ω, F, P) we denote a probability space and let $\tau_x : \Omega \mapsto \Omega$, $x \in \mathbb{R}^d$ be a d-dimensional dynamical system, which satisfies:

1. the group property: $\tau_0 = Id$ and $\tau_x \tau_{x'} = \tau_{x+x'}$, $\forall x, x' \in \mathbb{R}^d$;
2. it preserves the measure μ on Ω, i.e., for every $x \in \mathbb{R}^d$, and every μ-measurable set $A \in \Omega$, we have $\tau_x A$ is measurable, $\mu(\tau_x A) = \mu(A)$;
3. for any measurable function $h(\omega)$ on Ω, the function $h(\tau_x \omega)$ defined on $\Omega \times \mathbb{R}^d$ is also measurable.

We assume all functions on Ω to be F measurable. With equality between random functions we mean equality almost everywhere (a.e.) with respect to the measure P and often we omit the notation a.e.. A function $h(\omega)$ is called invariant if for any fixed $x \in \mathbb{R}^d$ holds $h(\tau_x \omega) = h(\omega)$. We assume that τ is ergodic, that is, if for every measurable function f the following holds

$$[h(\tau_x \omega) = h(\omega) \text{ for all } x \text{ and almost all } \omega] \Rightarrow h = \text{constant a.s.} \tag{4.4}$$

A homogeneous or stationary random field is a function $h \in L^1_{loc}(\mathbb{R}^d, L^1(\Omega))$ which satisfies

$$h(x + x', \omega) = h(x, \tau_{x'} \omega) \quad \forall x' \in \mathbb{R}^d.$$

We consider the following elliptic problem

$$\begin{aligned} -\text{div}\left(K(\frac{x}{\epsilon}, \omega)\nabla u_\epsilon\right) &= f \quad \text{in} \quad D \\ u_\epsilon &= 0 \quad \text{on} \quad D \end{aligned} \tag{4.5}$$

where $K(\frac{x}{\epsilon}, \omega)$ is a homogeneous ergodic random field, $f \in L^2(D)$ is a non-random function and ϵ is a small parameter (cf. (3.1)).

The behavior of the random field $u_\epsilon(x)$ as $\epsilon \to 0$ has been investigated under various assumptions on $K(\frac{x}{\epsilon}, \omega)$ ([42, 50, 57]). If $K(\frac{x}{\epsilon}, \omega)$ is strictly stationary and ergodic it was found that there exists K^* such that if $u^*(x)$ is the solution of the deterministic Dirichlet problem

$$\begin{aligned} -\text{div}\,(K^* \nabla u^*) &= f \quad \text{in} \quad D \\ u^* &= g \quad \text{on} \quad D \end{aligned} \tag{4.6}$$

then

$$\int_D E|u_\epsilon - u^*|^2 dx \to 0 \quad \text{as } \epsilon \to 0 \tag{4.7}$$

where E denotes the expectation. In this case the effective coefficient is defined as

$$[K^*]_{ij} = E\left[(e_i + \nabla \chi_i(y, \cdot))^T K(y, \cdot) e_j\right].$$

Hereby χ_i is the (unique up to a random constant) solution of

$$\begin{aligned} -\text{div}\,(K(y, \omega)\,(\nabla \chi_p(y, \omega) + p)) &= 0 \text{ in } \mathbb{R}^d \\ \nabla \chi_p \text{ is stationary}& \\ E\,[\nabla \chi_p] &= 0 \end{aligned} \tag{4.8}$$

for $p = e_i$. Furthermore is holds $\chi_i \in \{\chi \in L^2_{loc}(\mathbb{R}^d, L^2(\Omega)),\ \nabla \chi \in L^2_{uinf}(\mathbb{R}^d, L^2(\Omega))\}$. L^2_{unif} denotes the space of functions for which the L^2-norm on a unit ball is bounded above independently from the center of the ball (cf. [8]).

According to Birkhoff's theorem the effective coefficient K^* can be approximated by the spatial average

$$[K^*]_{ij} = \lim_{m \to \infty} \frac{1}{(2m)^d} \int_{|y| \leq m} (e_i + \nabla \chi_i(y, \cdot))^T K(y, \cdot) e_j \, dy.$$

This remains valid even if the coefficient is not ergodic then the effective coefficient is a random matrix, in general.

Note that in general in (4.8) the stationarity is fulfilled for the gradient $\nabla \chi_p$ of the solution only, but not for the solution itself.

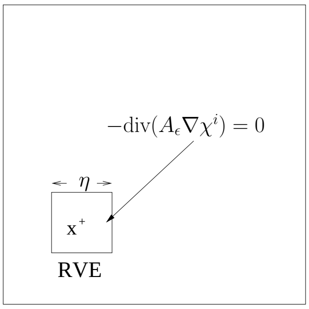


Figure 1: Illustration of numerical homogenization.

4.3. Numerical homogenization

In this section we describe numerical homogenization procedures. For practical reasons we solve (4.8) in a bounded domain Y_ρ^x instead of the abstract space $\mathbb{R}^d$. For a.e. ω we consider a numerical homogenization procedure as follows. Given a representative volume centered at a macroscopic point x with size ρ, $Y_\rho^x = (x - \rho/2, x + \rho/2)^d$, the local problem

$$\mathrm{div}(K(x, y, \omega)\nabla \chi_i) = 0, \text{ in } Y_\rho^x,\ i = 1, ..., d, \tag{4.9}$$

is solved subject to some boundary conditions. The boundary conditions are not very essential if there is a scale separation. For simplicity we will consider Dirichlet boundary conditions for the local problem

$$\chi_i = y_i \text{ on } \partial Y_\rho^x.$$

Additionally it is possible to use Neumann or periodic boundary conditions (see [16, 43]). Then the homogenized coefficients are computed as

$$K_\rho^*(x, \omega)e_i = \frac{1}{\rho^d} \int_{Y_\rho^x} K(x, y, \omega)\nabla \chi_i, \tag{4.10}$$

where e_i $(i = 1, ..., d)$ is a unit vector in the direction i. This procedure is repeated at every macroscopic point (see Figure 1 for illustration). It is known ([53]) that

$$e_j \cdot K_\rho^*(x, \omega)e_i = \frac{1}{\rho^d} \int_{Y_\rho^x} \nabla \chi_j \cdot K(x, y, \omega)\nabla \chi_i.$$

This is a practical way to obtain a converging approximation of the homogenized matrix, in the sense that

$$\lim_{\rho \to \infty} K_\rho^*(x, \omega) = K^*(x), \tag{4.11}$$

almost surely, and for almost all x (cf. [16]).

Note that it is convenient to rescale (4.9), such that

$$\mathrm{div}(K(x, \frac{y}{\epsilon}, \omega)\nabla \chi_i) = 0, \text{ in } Y_\eta^x,\ i = 1, ..., d, \tag{4.12}$$

with a rescaled domain Y_η^x for $\eta = \rho\epsilon$. For a fixed $\eta > 0$ the approximated coefficient K_η^* converges for $\epsilon \to 0$ to K^* a.s. and for $\eta \to \infty$ analogously. We denote the local homogenization procedure by $\mathcal{H}_\eta$, i.e.,

$$K_\eta^*(x, \omega) = \mathcal{H}_\eta(K(x, \frac{y}{\epsilon}, \omega)),$$

i.e., with the solution of (4.12) we have

$$e_j \cdot K_\eta^*(x,\omega) e_i = \frac{1}{\eta^d} \int_{Y_\eta^x} \nabla \chi_j \cdot K(x, \frac{y}{\epsilon}, \omega) \nabla \chi_i. \tag{4.13}$$

To give a precise convergence rate another condition besides the ergodicity is needed. If one assumes that the matrix $K_\epsilon(x,\omega)$ decorrelates at large distances, then one can obtain a convergence rate. This is briefly mentioned in the next section.

4.4. Estimation of the rate of convergence

We briefly mention some results of the rate of convergence if the coefficient is an ergodic random field and satisfies a uniform mixing condition. More details can be found in [16, 57].
First we introduce the uniform mixing condition. Be A_j a subset of $\mathbb{R}^d$. With $\Phi(A_j)$ we denote the σ-algebra generated by the random field $K(\frac{x}{\epsilon}, \cdot)$, $x \in A_j$. The family of σ-algebras $(\Phi(A_j), A_j \subset \mathbb{R}^d)_{j \in \mathbb{N}}$ satisfies the condition of uniformly mixing, if

$$|E\xi\vartheta - E\xi E\vartheta| \leq b(q)(E\xi^2)^{1/2}(E\vartheta^2)^{1/2} \tag{4.14}$$

where the random quantity ξ is $\Phi(A_j)$ measurable and ϑ is $\Phi(A_i)$ measurable, and $q = \inf\{|x - y|,\ x \in A_j,\ y \in A_i\}$ for $i \neq j$. Note we denote the Euclidian norm with $|\cdot|$, as well as the absolute value. In the following we assume in addition to ergodicity that this condition is fulfilled for the coefficient $K(x,\omega)$ with

$$b(q) \leq \frac{C}{q^k} \tag{4.15}$$

for some $k > 0$. That means that the coefficient decorrelates at large distances. We note that (4.14) and (4.15) imply strong mixing for the coefficients K. It has been shown in [16] that

$$E(|||K_\eta^* - K^*|||^2) \leq C \left(\frac{\epsilon}{\eta}\right)^\beta \tag{4.16}$$

for some $\beta > 0$ and $C > 0$ that depend on the correlation rate, but are independent of η and ϵ, and where $|||\cdot|||$ denotes any norm of the $d \times d$ matrices.

To proof this one introduces a 'penalized' or 'disturbed' problem in $\mathbb{R}^d$

$$-\operatorname{div}(K(y,\omega)\nabla\chi^\kappa) + \kappa\chi^\kappa = \operatorname{div} K(y,\omega) \tag{4.17}$$

with $\kappa > 0$. In comparison to (4.8) this problem has a unique solution $\chi^\kappa \in (H^1_{loc}(\mathbb{R}^d))^d$ which is stationary not only in its gradient. With $\chi^{\kappa,\rho}$ we denote the solution of (4.17) in $Y_\rho = (0,\rho)^d$ and define

$$\begin{aligned} K_\kappa^* &= \frac{1}{\rho^d} \int_{Y_\rho} K(y,\omega)(\nabla\chi^\kappa + Id) \\ K_{\rho,\kappa}^* &= \frac{1}{\rho^d} \int_{Y_\rho} K(y,\omega)(\nabla\chi^{\kappa,\rho} + Id). \end{aligned}$$

By choosing κ appropriate and with

$$E(|||K_\rho^* - K^*|||^2) \leq E(|||K_\kappa^* - K^*|||^2) + E(|||K_\kappa^* - K_{\rho,\kappa}^*|||^2) + E(|||K_\rho^* - K_{\rho,\kappa}^*|||^2)$$

one can show (cf. [16])

$$E(|||K_\rho^* - K^*|||^2) \leq C\rho^{-\beta}$$

and with $\rho = \frac{\eta}{\epsilon}$ we have (4.16).

Part I.

Using Karhunen-Loève expansion for numerical homogenization

In this part we consider the Karhunen-Loève expansion (cf. [46]) in a multiscale context. This expansion is widely used to sample random fields with a given distribution (e.g., [4, 47]). We use it in Part II and III to generate realizations of the stochastic coefficient. Here we use the Karhunen-Loève expansion in combination with a polynomial chaos expansion ([55, 54]) as in the work of Frauenfelder et al. ([33]) to reduce the stochastic problem to many deterministic problems. In particular we reduce the stochastic cell problems to deterministic ones to approximate the mean of the effective coefficient. In the next section we introduce two ways to approximate the mean of the homogenized matrix: The first one is based on a standard Monte Carlo approach. In the second one we reduce the high dimensional stochastic equations to many deterministic problems. This reduction is done by a combination of a Karhunen-Loève expansion and a polynomial chaos expansion. In the numerics we compare these two approaches. To compute the Karhunen-Loève expansion of a given random field, the eigenpairs of the covariance operator are required. Because they are in general analytically unknown one has to solve large systems of equations. To overcome this problem we show how to approximate a general covariance matrix with hierarchical matrices ([37]). A similar approach can be found in ([44]). In Section 6 we introduce the problem types, which we consider in the numerics (cf. Sec. 9). We solve the problems with a cell-centered finite volume method introduced in Section 7. We conclude this part with numerical results (cf. Sec. 9).

5. Diffusion with a stochastic coefficient

We consider a microscale problem with a stochastic coefficient $K(\frac{x}{\epsilon},\omega)$:

$$\begin{aligned} -\text{div}\left(K(\frac{x}{\epsilon},\omega)\nabla u_\epsilon(x,\omega)\right) &= f(x), \quad x \in D \\ u_\epsilon &= g, \qquad x \in \partial D \end{aligned}$$

for P a.e. $\omega \in \Omega$, with a σ-finite probability space (Ω, F, P). We assume that the known information about the coefficient includes its expected value ($y = \frac{x}{\epsilon}$)

$$E_K(y) := E[K(y,\omega)] = \int_\Omega K(y,\omega)dP(\omega)$$

and its covariance function

$$\text{cov}(y,y') := E[K(y,\omega)K(y',\omega)] - E[K(y,\omega)]E[K(y',\omega)].$$

In the deterministic case we achieve a macroscale problem with an effective tensor K^* via homogenization. To determine K^* we have to solve appropriate cell problems. In the following sections we present two approaches to approximate the expectation of the tensor K^*. In the first ansatz we compute the upscaled tensor via Monte Carlo simulation. In the second one we use the Karhunen-Loève expansion of the random field $K(y,\omega)$.

5.1. Monte Carlo simulation

For each sample $K_n^{MC}(y) = E_K(y) + X_n$ we determine the upscaled tensor K_n^{*MC}. Here, X_n denotes a random variable with the given covariance and with the expected value equal to zero. As approximation of the homogenized coefficient we use K_N^{*MC}, the empirical expectation of K_n^{*MC}.

5.1.1. Cell problem

In this case we consider a cell problem with $K_n^{MC}(y) = E_K(y) + X_n$ as coefficient. It reads in i^{th} direction $(i = 1, \cdots, d)$:

$$\begin{aligned} -\mathrm{div}\left(K_n^{MC}(y)\nabla\chi_i^{MC,n}(y)\right) &= 0, \quad y \in Y \\ \chi_i^{MC,n} &= y \cdot e_i, \quad y \in \partial Y. \end{aligned} \tag{5.1}$$

As in the deterministic case the upscaled tensor is determined by

$$\left(K_n^{*MC}\right)_{ij} = \int_Y \nabla\chi_i^{MC,n}(y) \cdot K_n^{MC}(y)\nabla\chi_j^{MC,n}(y)\, dy. \tag{5.2}$$

As an upscaled tensor of the stochastic problem we use the arithmetic mean

$$K_N^{*MC} = \frac{1}{N}\sum_{n=1}^{N} K_n^{*MC},$$

where N denotes the number of realizations.

5.2. Karhunen-Loève expansion

The second approach is based on the work of Frauenfelder, Schwab and Todor (cf. [33]). Here we use the Karhunen-Loève expansion of the stochastic coefficient $K(y, \omega)$ in the unit cell Y

$$K^{KL}(y, \omega) = E_K(y) + \sum_{m=1}^{\infty} \sqrt{\lambda_m}\phi_m(y)X_m(\omega), \tag{5.3}$$

where the random variables X_m are centered at 0 and pairwise uncorrelated. In addition we assume independence of the random variables and $\|X_m\|_{L^\infty(\Omega,dP)} \leq c_x \in \mathbb{R}$, $\forall m \in \mathbb{N}$. With $(\lambda_m, \phi_m)_{1\leq m\leq\infty}$ we denote the sequence of eigenpairs of the covariance operator $(\mathcal{K}\phi)(y) = \int_Y \mathrm{cov}(y, y')\phi(y')\, dy'$, i.e., they fulfill

$$\int_{Y\times Y} \mathrm{cov}(y, y')\phi_m(y')v(y)\, dy'dy = \lambda_m \int_Y \phi_m(y)v(y)\, dy \quad \forall v \in L^2(Y).$$

5.2.1. Discrete Karhunen-Loève expansion

Let $\{S_h\}_{h\geq 0} \subseteq L^2(Y)$ denote a family of finite element (FE) spaces with mesh width h. The discrete eigenproblem reads:

$$\int_{Y\times Y} \mathrm{cov}(y, y')\phi_m^h(y')v(y)\, dy'dy = \lambda_m^h \int_Y \phi_m^h(y)v(y)\, dy \quad \forall v \in S_h. \tag{5.4}$$

If $\tilde{N}$ is the dimension of the FE space and $\varphi_n \in S_h,\ 1 \leq n \leq \tilde{N}$, a basis of S_h, (5.4) is equivalent to

$$\int_{Y\times Y} \operatorname{cov}(y,y')\phi_m^h(y')\varphi_n(y)\,dy'dy = \lambda_m^h \int_Y \phi_m^h(y)\varphi_n(y)\,dy, \quad 1 \leq n \leq \tilde{N}.$$

Additionally we write ϕ_m^h as a sum of the basis functions

$$\phi_m^h(y) = \sum_{j=1}^{\tilde{N}} \Phi_m^j \varphi_j(y).$$

Then the problem reads as follows:

$$\sum_{j=1}^{\tilde{N}} \Phi_m^j \int_{Y\times Y} \operatorname{cov}(y,y')\varphi_j(y')\varphi_n(y)\,dy'dy = \lambda_m^h \sum_{j=1}^{\tilde{N}} \Phi_m^j \int_Y \varphi_j(y)\varphi_n(y)\,dy \quad 1 \leq n \leq \tilde{N},$$

thus we require the solution of the dense matrix eigenproblem

$$\mathbf{K}\Phi_m = \lambda_m^h \mathbf{M}\Phi_m, \tag{5.5}$$

with

$$\begin{aligned} (\mathbf{K})_{ij} &= \int_{Y\times Y} \operatorname{cov}(y,y')\varphi_j(y')\varphi_i(y)\,dy'dy, \\ (\mathbf{M})_{ij} &= \int_Y \varphi_j(y)\varphi_i(y)\,dy. \end{aligned}$$

These matrices are symmetric and positive semi-definite. If we choose an orthogonal basis $\varphi_n \in S_h$, $\mathbf{M}$ is diagonal. Then we achieve by multiplying with $\mathbf{M}^{-1/2}$

$$\mathbf{M}^{-1/2}\mathbf{K}\Phi_m = \lambda_m^h \mathbf{M}^{1/2}\Phi_m,$$

and finally

$$\tilde{\mathbf{K}}\tilde{\Phi}_m = \lambda_m^h \tilde{\Phi}_m,$$

with $\tilde{\mathbf{K}} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ and $\tilde{\Phi}_m = \mathbf{M}^{1/2}\Phi_m$.

5.2.2. Cell problem

By

$$K_M^{KL}(y,\omega) = E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y)X_m(\omega) \tag{5.6}$$

we denote the truncated Karhunen-Loève expansion of the stochastic coefficient. With $\tilde{K}_M^{KL}$ we associate the (M, 1) polynomial chaos expansion

$$\tilde{K}_M^{KL}(y,z) = \tilde{K}_M^{KL}(y,z_1,z_2,\ldots,z_M) = E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y)z_m. \tag{5.7}$$

We use this coefficient for the cell problem

$$\begin{aligned} -\operatorname{div}\left(\tilde{K}_M^{KL}(y,z)\nabla\chi_i^{KL}(y,z)\right) &= 0, \quad y \in Y,\, z \in I^M = \left[-\frac{1}{2},\frac{1}{2}\right]^M \\ \chi_i^{KL} &= y \cdot e_i, \quad y \in \partial Y. \end{aligned} \tag{5.8}$$

In the following we consider the equivalent formulation

$$\begin{aligned} -\operatorname{div}\left(\tilde{K}_M^{KL}(y,z)\left(\nabla\hat{\chi}_i^{KL}(y,z)+e_i\right)\right) &= 0, \quad y\in Y,\, z\in I^M=\left[-\frac{1}{2},\frac{1}{2}\right]^M \\ \hat{\chi}_i^{KL} &= 0, \quad y\in\partial Y \end{aligned} \tag{5.9}$$

with $\hat{\chi}_i^{KL}=\chi_i^{KL}-y\cdot e_i$.
This is a high dimensional deterministic problem where we can calculate the upscaled coefficient. However, instead of finite elements or volumes we approximate the solution w.r.t. z via polynomials. Therefore we define, for $r\in\mathbb{N}_0$, the space of polynomials of degree at most r,

$$\mathcal{P}_r := \operatorname{span}\{1,t,t^2,\dots,t^r\}\subseteq L^2(I)$$

and, for $\mathbf{r}=(r_1,r_2,\dots,r_M)\in\mathbb{N}_0^M$, a polynomial space by

$$\mathcal{P}_{\mathbf{r}} := \mathcal{P}_{r_1}\otimes\mathcal{P}_{r_2}\otimes\dots\otimes\mathcal{P}_{r_M}\subseteq L^2(I^M).$$

This semi-discretization can be solved with each polynomial basis of $\mathcal{P}_{\mathbf{r}}$. Because in general it is a coupled system of deterministic equations the computational costs are high. In order to reduce these we make the following ansatz to decouple the problem: We denote for $1\le m\le M$ and $r_m\in\mathbb{N}_0$, by $\left(\mu_j^{r_m},P_j^{r_m}\right)_{0\le j\le r_m}$ the eigenpairs of the symmetric bilinear form

$$(u,v)\to\int_{-1/2}^{1/2}u(t)v(t)t\,d\rho_m(t)$$

over $\mathcal{P}_{r_m}:=\operatorname{span}\{1,t,t^2,\dots,t^{r_m}\}$. Let the eigenpairs be orthonormal w.r.t. to the weight ρ_m. Thereby we denote by ρ_m the law of the random variable X_m,

$$\rho_m(\mathcal{B}) := P(X_m\in\mathcal{B})\quad\text{for any Borel set } \mathcal{B}\subseteq I$$

and, for all $M\ge 1$, we define a probability measure on I^M by

$$\rho := \rho_1\times\rho_2\times\dots\times\rho_M$$

The eigenproblem reads

$$\int_I P(t)\psi_j(t)t\,d\rho_m(t)=\mu\int_I P(t)\psi_j(t)\,d\rho_m(t)\quad\forall 0\le j\le r_m, \tag{5.10}$$

with the solution $(\mu,P(t))\in\mathbb{R}\times\mathcal{P}_{r_m}$ where $\{\psi_j\}_{0\le j\le r_m}$ is a basis of $\mathcal{P}_{r_m}$.
If we write P as a sum of the polynomial basis, i.e.,

$$P(t)=\sum_{i=0}^{r_m}p_i\psi_i(t),$$

we obtain

$$\sum_{i=0}^{r_m}p_i\int_I\psi_i(t)\psi_j(t)t\,d\rho_m(t)=\mu\sum_{i=0}^{r_m}p_i\int_I\psi_i(t)\psi_j(t)\,d\rho_m(t)\quad\forall 0\le j\le r_m.$$

In matrix form the eigenproblem reads

$$\mathbf{A}p=\mu\mathbf{B}p,$$

with the vector of coefficients $p := (p_0, p_1, \dots, p_{r_m})^T$, and the matrices

$$(\mathbf{A})_{ij} := \int_I \psi_i(t)\psi_j(t) t \, d\rho_m(t) \quad \text{and} \quad (\mathbf{B})_{ij} := \int_I \psi_i(t)\psi_j(t) \, d\rho_m(t).$$

On $\mathbb{N}_0^M$ we define the partial ordering

$$\mathbf{j} \leq \mathbf{r} \Leftrightarrow j_m \leq r_m, \quad \forall 1 \leq m \leq M, \tag{5.11}$$

$\forall \, \mathbf{j} = (j_1, j_2, \dots, j_M),\ \mathbf{r} = (r_1, r_2, \dots, r_M) \in \mathbb{N}^M$ for notational convenience. We further set

$$P_{\mathbf{j}}^{\mathbf{r}}(z) := P_{j_1}^{r_1}(z_1) \, P_{j_2}^{r_2}(z_2) \cdots P_{j_M}^{r_M}(z_M) \quad \text{for } \mathbf{j} \leq \mathbf{r}.$$

$P_{j_m}^{r_m}$ is a polynomial in z_m, so $P_{\mathbf{j}}^{\mathbf{r}}$ is a polynomial in $z = (z_1, z_2, \dots, z_M)$ and

$$\mathcal{P}_{\mathbf{r}} = \operatorname{span}\{P_{\mathbf{j}}^{\mathbf{r}} \mid \mathbf{j} \leq \mathbf{r}\}.$$

Then $(P_{\mathbf{j}}^{\mathbf{r}})_{\mathbf{j} \leq \mathbf{r}}$ is the basis of $\mathcal{P}_{\mathbf{r}}$ we use to decouple the semi discrete problem.
We find (cf. [33], proposition 4.17):

Theorem 5.1
For a given $\mathbf{r} \in \mathbb{N}_0^M$*, let* χ_i^{KL} *be the solution of (5.8) (polynomial approximation w.r.t. second variable z). For every multi index* $\mathbf{j} \leq \mathbf{r}$ *we denote by* $\tilde{\chi}_{i,\mathbf{j}}^{KL}$ *the solution of the deterministic problem*

$$\begin{aligned} -\mathit{div}\left(K_M^{KL,\mathbf{j}}(y) \nabla \tilde{\chi}_{i,\mathbf{j}}^{KL}(y)\right) &= f_{i,\mathbf{j}}^{KL}(y), \quad y \in Y, \\ \tilde{\chi}_{i,\mathbf{j}}^{KL} &= 0, \quad y \in \partial Y. \end{aligned} \tag{5.12}$$

with

$$K_M^{KL,\mathbf{j}}(y) \quad := \quad E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m} \phi_m(y) \mu_{j_m}^{r_m} \tag{5.13}$$

$$\bar{K}_M^{KL,\mathbf{j}}(y) \quad := \quad E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m} \phi_m(y) \frac{\int_I P_{j_m}^{r_m}(z_m) z_m \, d\rho_m(z_m)}{\int_I P_{j_m}^{r_m}(z_m) \, d\rho_m(z_m)} \tag{5.14}$$

$$\bar{P}_{\mathbf{j}}^{\mathbf{r}} \quad := \quad \int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z) \, d\rho(z) = \prod_{m=1}^{M} \int_I P_{j_m}^{r_m}(z_m) \, d\rho_m(z_m) \tag{5.15}$$

$$f_{i,\mathbf{j}}^{KL}(y) \quad := \quad \bar{P}_{\mathbf{j}}^{\mathbf{r}} \ \mathit{div}\left(e_i \cdot \bar{K}_M^{KL,\mathbf{j}}(y)\right) \tag{5.16}$$

Then

$$\chi_i^{KL}(y, z) = \sum_{\mathbf{j} \leq \mathbf{r}} \tilde{\chi}_{i,\mathbf{j}}^{KL}(y) P_{\mathbf{j}}^{\mathbf{r}}(z) + y \cdot e_i. \tag{5.17}$$

So we have to solve a deterministic cell problem in each direction for a given $\mathbf{r} \in \mathbb{N}_0^M$ for every multi index $\mathbf{j} \leq \mathbf{r}$.

Next we determine the expected value of the upscaled coefficient

$$\left(K^{*KL}(z)\right)_{lk} = \int_Y \nabla \chi_l^{KL} \cdot \tilde{K}_M^{KL}(y, z) \nabla \chi_k^{KL} \, dy :$$

$$
\begin{aligned}
& E[\big(K^{*KL}(z)\big)_{lk}] \\
\approx\quad & \int_{I^M} e_l K^{*KL}(z) e_k \, d\rho(z) \\
=\quad & \int_{I^M}\int_Y \nabla\chi_l^{KL} \cdot \tilde{K}_M^{KL}(y,z)\nabla\chi_k^{KL} \, dy\, d\rho(z) \\
\overset{eq.(5.17)}{=}\quad & \int_{I^M}\int_Y \nabla\left(\sum_{\mathbf{j}\le\mathbf{r}} \tilde{\chi}_{l,\mathbf{j}}^{KL}(y)P_{\mathbf{j}}^{\mathbf{r}}(z) + y\cdot e_l\right) \\
& \qquad\qquad \cdot\tilde{K}_M^{KL}(y,z)\nabla\left(\sum_{\mathbf{j}'\le\mathbf{r}} \tilde{\chi}_{k,\mathbf{j}'}^{KL}(y)P_{\mathbf{j}'}^{\mathbf{r}}(z) + y\cdot e_k\right) dy\, d\rho(z) \\
=\quad & \int_{I^M}\int_Y \left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\tilde{\chi}_{l,\mathbf{j}}^{KL}(y)P_{\mathbf{j}}^{\mathbf{r}}(z) + e_l\right) \\
& \qquad\qquad \cdot\tilde{K}_M^{KL}(y,z)\left(\sum_{\mathbf{j}'\le\mathbf{r}} \nabla\tilde{\chi}_{k,\mathbf{j}'}^{KL}(y)P_{\mathbf{j}'}^{\mathbf{r}}(z) + e_k\right) dy\, d\rho(z) \\
=\quad & \int_{I^M}\int_Y \left[\left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\tilde{\chi}_{l,\mathbf{j}}^{KL}(y)P_{\mathbf{j}}^{\mathbf{r}}(z)\right)\cdot\tilde{K}_M^{KL}(y,z)\left(\sum_{\mathbf{j}'\le\mathbf{r}} \nabla\tilde{\chi}_{k,\mathbf{j}'}^{KL}(y)P_{\mathbf{j}'}^{\mathbf{r}}(z)\right)\right. \\
& \qquad\qquad + e_l\cdot\tilde{K}_M^{KL}(y,z)e_k + \left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\tilde{\chi}_{l,\mathbf{j}}^{KL}(y)P_{\mathbf{j}}^{\mathbf{r}}(z)\right)\cdot\tilde{K}_M^{KL}(y,z)e_k \\
& \qquad\qquad \left. + \, e_l\cdot\tilde{K}_M^{KL}(y,z)\left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\tilde{\chi}_{k,\mathbf{j}}^{KL}(y)P_{\mathbf{j}}^{\mathbf{r}}(z)\right)\right] dy\, d\rho(z) \\
=:\quad & I_1 + I_2 + I_3 + I_4.
\end{aligned}
$$

In the following we consider each integral separately. We start with the second one.

$$
\begin{aligned}
I_2 \quad &= \int_{I^M}\int_Y e_l\cdot\tilde{K}_M^{KL}(y,z)e_k\, dy\, d\rho(z) \\
&= \int_Y e_l\cdot\left(E_K(y)\int_{I^M} 1\, d\rho(z) + \sum_{m=1}^{M}\sqrt{\lambda_m}\phi_m(y)\int_{I^M} z_m\, d\rho(z)\right) e_k\, dy \\
&= \int_Y e_l\cdot\left(E_K(y)\prod_{m=1}^{M}\underbrace{\int_I 1\, d\rho_m(z_m)}_{\substack{=1,\\ \rho_m \text{ density on I}}} + \sum_{m=1}^{M}\sqrt{\lambda_m}\phi_m(y)\underbrace{\int_I z_m\, d\rho_m(z_m)}_{\substack{=0,\\ X_m \text{centered at } 0}}\right) e_k\, dy \\
&= \int_Y e_l\cdot E_K(y)e_k\, dy
\end{aligned}
$$

$$= \quad \delta_{lk} \int_Y E_K(y)\,dy.$$

For the first term I_1 we use the ortho-normality of $P^{r_m}_{j_m}$, i.e.,

$$\int_I P^{r_m}_{j_m}(z_m) P^{r_m}_{i_m}(z_m)\,d\rho_m(z_m) = \delta_{i_m j_m} \tag{5.18}$$

and the equation of the corresponding eigenproblem with $P^{r_m}_{j'_m}$ as test function, i.e.,

$$\int_I P^{r_m}_{j_m}(z_m) P^{r_m}_{j'_m}(z_m) z_m\,d\rho_m(z_m) = \mu^{r_m}_{j_m} \int_I P^{r_m}_{j_m}(z_m) P^{r_m}_{j'_m}(z_m)\,d\rho_m(z_m) = \mu^{r_m}_{j_m} \delta_{j_{m'} j'_{m'}}. \tag{5.19}$$

Furthermore we first consider

$$\begin{aligned}
I_5 \quad &:= \quad \int_{I^M} P^{\mathbf{r}}_{\mathbf{j}}(z) \tilde{K}^{KL}_M(y,z) P^{\mathbf{r}}_{\mathbf{j}'}(z)\,d\rho(z) \\
&= \quad \int_{I^M} \prod_{m'=1}^{M} P^{r_{m'}}_{j_{m'}}(z_{m'}) \left(E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) z_m \right) \prod_{m'=1}^{M} P^{r_{m'}}_{j'_{m'}}(z_{m'})\,d\rho(z) \\
&= \quad E_K(y) \prod_{m'=1}^{M} \underbrace{\int_I P^{r_{m'}}_{j_{m'}}(z_{m'}) P^{r_{m'}}_{j'_{m'}}(z_{m'})\,d\rho_{m'}(z_{m'})}_{\overset{eq.(5.18)}{=} \delta_{j_m j'_m}} \\
&\quad\quad + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) \int_{I^M} z_m \prod_{m'=1}^{M} P^{r_{m'}}_{j_{m'}}(z_{m'})^{r_{m'}}_{j'_{m'}} P^{r_{m'}}_{j'_{m'}}(z_{m'})\,d\rho(z) \\
&= \quad E_K(y)\delta_{\mathbf{j}\mathbf{j}'} + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) \int_I P^{r_m}_{j_m}(z_m) P^{r_m}_{j'_m}(z_m) z_m\,d\rho_m(z_m) \\
&\quad\quad \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \underbrace{\int_I P^{r_{m'}}_{j_{m'}}(z_{m'}) P^{r_{m'}}_{j'_{m'}}(z_{m'})\,d\rho_{m'}(z_{m'})}_{\overset{eq.(5.18)}{=} \delta_{j_{m'} j'_{m'}}} \\
&= \quad E_K(y)\delta_{\mathbf{j}\mathbf{j}'} + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) \underbrace{\int_I P^{r_m}_{j_m}(z_m) P^{r_m}_{j'_m}(z_m) z_m\,d\rho_m(z_m)}_{\overset{eq.(5.19)}{=} \mu^{r_m}_{j_m} \delta_{j_m j'_m}} \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \delta_{j_{m'} j'_{m'}} \\
&= \quad \left(E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) \mu^{r_m}_{j_m} \right) \delta_{\mathbf{j}\mathbf{j}'} \\
&\overset{eq.(5.13)}{=} \quad K^{KL,\mathbf{j}}_M(y)\delta_{\mathbf{j}\mathbf{j}'}.
\end{aligned}$$

Therefore for I_1 we find

$$I_1 \quad = \quad \int_{I^M} \int_Y \left(\sum_{\mathbf{j} \leq \mathbf{r}} \nabla \tilde{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z) \right) \cdot \tilde{K}^{KL}_M(y,z) \left(\sum_{\mathbf{j}' \leq \mathbf{r}} \nabla \tilde{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) \right) dy\,d\rho(z)$$

$$\begin{aligned}
&= \int_Y \left(\sum_{\mathbf{j}\le\mathbf{r}} \int_{I^M} \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z) \cdot \tilde{K}^{KL}_M(y,z) \left(\sum_{\mathbf{j}'\le\mathbf{r}} \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) \right) d\rho(z) \right) dy \\
&= \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \sum_{\mathbf{j}'\le\mathbf{r}} \underbrace{\int_{I^M} P^{\mathbf{r}}_{\mathbf{j}}(z) \cdot \tilde{K}^{KL}_M(y,z) P^{\mathbf{r}}_{\mathbf{j}'}(z)\, d\rho(z)}_{=I_5} \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}'}(y)\, dy \\
&= \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_M(y) \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\, dy.
\end{aligned}$$

The last two integrals are nearly the same; only the indices differ. That is why we consider I_3 only.

$$\begin{aligned}
I_3 &= \int_{I^M} \int_Y \left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z) \right) \cdot \tilde{K}^{KL}_M(y,z) e_k\, dy\, d\rho(z) \\
&= \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \int_{I^M} P^{\mathbf{r}}_{\mathbf{j}}(z) \tilde{K}^{KL}_M(y,z)\, d\rho(z)\, e_k\, dy \\
&\overset{eq.(5.7)}{=} \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \int_{I^M} P^{\mathbf{r}}_{\mathbf{j}}(z) \left(E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m} \phi_m(y) z_m \right) d\rho(z)\, e_k\, dy \\
&\overset{eq.(5.15)}{=} \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \bar{P}^{\mathbf{r}}_{\mathbf{j}} \left(E_K(y) + \sum_{m=1}^{M} \sqrt{\lambda_m} \phi_m(y) \frac{\int_I P^{r_m}_{j_m}(z_m) z_m\, d\rho_m(z_m)}{\int_I P^{r_m}_{j_m}(z_m)\, d\rho_m(z_m)} \right) e_k\, dy \\
&\overset{eq.(5.14)}{=} \sum_{\mathbf{j}\le\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \bar{K}^{KL,\mathbf{j}}_M(y)\, e_k\, dy.
\end{aligned}$$

Therefore for the expected value of the upscaled coefficient we obtain

$$\begin{aligned}
&\int_{I^M} e_l \cdot K^{*KL}(z) e_k\, d\rho(z) \\
&= \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_M(y) \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\, dy + \delta_{lk} \int_Y E_K(y)\, dy \\
&\quad + \sum_{\mathbf{j}\le\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \bar{K}^{KL,\mathbf{j}}_M(y)\, e_k\, dy + \sum_{\mathbf{j}\le\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y) \cdot \bar{K}^{KL,\mathbf{j}}_M(y)\, e_l\, dy \\
&= \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_M(y) \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\, dy + \delta_{lk} \int_Y E_K(y)\, dy \\
&\quad + \sum_{\mathbf{j}\le\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \bar{K}^{KL,\mathbf{j}}_M(y) \left(\nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot e_k + \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y) \cdot e_l \right) dy.
\end{aligned}$$

The first term

$$\tilde{K}^{KL}_{\mathrm{det}} := \sum_{\mathbf{j}\le\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_M(y) \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\, dy$$

is the sum of the upscaled coefficients of the deterministic problems.
With the notation $\int_Y E_K(y)\, dy = \bar{E}_K$ we end with

$$\begin{aligned}
&\int_{I^M} e_l K^{*KL}(z) e_k \, d\rho(z) \\
= \quad & \tilde{K}^{KL}_{\mathrm{det}} + \delta_{lk}\, \bar{E}_K + \sum_{\mathbf{j}\le\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \bar{K}^{KL,\mathbf{j}}_M(y) \left(\nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y)\cdot e_k + \nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\cdot e_l\right) dy.
\end{aligned} \tag{5.20}$$

If we translate the solution $\tilde{\chi}^{KL}_{i,\mathbf{j}}(y)$ to nonzero boundary condition, i.e., if we consider

$$\hat{\chi}^{KL}_{i,\mathbf{j}}(y) := \tilde{\chi}^{KL}_{i,\mathbf{j}}(y) + y\cdot e_i, \tag{5.21}$$

we get

$$\begin{aligned}
& \int_{I^M} e_l\cdot K^{*KL}(z) e_k \, d\rho(z) \\
= \quad & \int_{I^M}\int_Y \nabla\chi_l \cdot \tilde{K}^{KL}_M(y,z)\nabla\chi_k \, dy\, d\rho(z) \\
\overset{eq.(5.17)+(5.21)}{=} \quad & \int_{I^M}\int_Y \nabla\left(\sum_{\mathbf{j}\le\mathbf{r}} \left(\hat{\chi}^{KL}_{l,\mathbf{j}}(y) - y\cdot e_l\right) P^{\mathbf{r}}_{\mathbf{j}}(z) + y\cdot e_l\right) \\
& \qquad \cdot\tilde{K}^{KL}_M(y,z)\nabla\left(\sum_{\mathbf{j}'\le\mathbf{r}} \left(\hat{\chi}^{KL}_{k,\mathbf{j}'}(y) - y\cdot e_k\right) P^{\mathbf{r}}_{\mathbf{j}'}(z) + y\cdot e_k\right) dy\, d\rho(z) \\
= \quad & \int_{I^M}\int_Y \left(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z) - \sum_{\mathbf{j}\le\mathbf{r}} e_l P^{\mathbf{r}}_{\mathbf{j}}(z) + e_l\right) \\
& \qquad \cdot\tilde{K}^{KL}_M(y,z)\left(\sum_{\mathbf{j}'\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) - \sum_{\mathbf{j}'\le\mathbf{r}} e_k P^{\mathbf{r}}_{\mathbf{j}'}(z) + e_k\right) dy\, d\rho(z) \\
= \quad & \int_{I^M}\int_Y \Bigg(\sum_{\mathbf{j}\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) \\
& \qquad - \sum_{\mathbf{j}\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} e_k P^{\mathbf{r}}_{\mathbf{j}'}(z) \\
& \qquad + \sum_{\mathbf{j}\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{l,\mathbf{j}}(y) P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) e_k \\
& \qquad - \sum_{\mathbf{j}\le\mathbf{r}} e_l P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) \\
& \qquad + \sum_{\mathbf{j}\le\mathbf{r}} e_l P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} e_k P^{\mathbf{r}}_{\mathbf{j}'}(z) \\
& \qquad - \sum_{\mathbf{j}\le\mathbf{r}} e_l P^{\mathbf{r}}_{\mathbf{j}}(z)\cdot\tilde{K}^{KL}_M(y,z) e_k \\
& \qquad + e_l\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} \nabla\hat{\chi}^{KL}_{k,\mathbf{j}'}(y) P^{\mathbf{r}}_{\mathbf{j}'}(z) \\
& \qquad - e_l\cdot\tilde{K}^{KL}_M(y,z) \sum_{\mathbf{j}'\le\mathbf{r}} e_k P^{\mathbf{r}}_{\mathbf{j}'}(z) + e_l\cdot\tilde{K}^{KL}_M(y,z) e_k \Bigg) dy\, d\rho(z).
\end{aligned}$$

With our previous notation ($\hat{I}_i$ means dependency on $\hat{\chi}_{l,\mathbf{j}}$ instead of $\tilde{\chi}_{l,\mathbf{j}}$) we get

$$\begin{aligned}
& \int_{I^M} e_l \cdot K^{*KL}(z) e_k \, d\rho(z) \\
= \quad & \hat{I}_1 - \int_{I^M} \int_Y \sum_{\mathbf{j} \leq \mathbf{r}} \nabla \hat{\chi}_{l,\mathbf{j}}^{KL}(y) P_{\mathbf{j}}^{\mathbf{r}}(z) \cdot \tilde{K}_M^{KL}(y,z) \sum_{\mathbf{j}' \leq \mathbf{r}} e_k P_{\mathbf{j}'}^{\mathbf{r}}(z) \, dy \, d\rho(z) \\
& + \hat{I}_3 - \int_{I^M} \int_Y \sum_{\mathbf{j} \leq \mathbf{r}} e_l P_{\mathbf{j}}^{\mathbf{r}}(z) \cdot \tilde{K}_M^{KL}(y,z) \sum_{\mathbf{j}' \leq \mathbf{r}} \nabla \hat{\chi}_{k,\mathbf{j}'}^{KL}(y) P_{\mathbf{j}'}^{\mathbf{r}}(z) \, dy \, d\rho(z) \\
& + \int_{I^M} \int_Y \sum_{\mathbf{j} \leq \mathbf{r}} e_l P_{\mathbf{j}}^{\mathbf{r}}(z) \cdot \tilde{K}_M^{KL}(y,z) \sum_{\mathbf{j}' \leq \mathbf{r}} e_k P_{\mathbf{j}'}^{\mathbf{r}}(z) \, dy \, d\rho(z) \\
& - \int_{I^M} \int_Y \sum_{\mathbf{j} \leq \mathbf{r}} e_l P_{\mathbf{j}}^{\mathbf{r}}(z) \cdot \tilde{K}_M^{KL}(y,z) e_k \, dy \, d\rho(z) + \hat{I}_4 \\
& - \int_{I^M} \int_Y e_l \cdot \tilde{K}_M^{KL}(y,z) \sum_{\mathbf{j}' \leq \mathbf{r}} e_k P_{\mathbf{j}'}^{\mathbf{r}}(z) \, dy \, d\rho(z) + \hat{I}_2 \\
= \quad & \hat{I}_1 + \hat{I}_2 + \hat{I}_3 + \hat{I}_4 \\
& - \sum_{\mathbf{j} \leq \mathbf{r}} \sum_{\mathbf{j}' \leq \mathbf{r}} \int_Y \nabla \hat{\chi}_{l,\mathbf{j}}^{KL}(y) \cdot \underbrace{\left(\int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z) \tilde{K}_M^{KL}(y,z) P_{\mathbf{j}'}^{\mathbf{r}}(z) \, d\rho(z) \right)}_{= I_5} e_k \, dy \\
& - \sum_{\mathbf{j} \leq \mathbf{r}} \sum_{\mathbf{j}' \leq \mathbf{r}} \int_Y e_l \cdot \underbrace{\left(\int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z) \tilde{K}_M^{KL}(y,z) P_{\mathbf{j}'}^{\mathbf{r}}(z) \, d\rho(z) \right)}_{= I_5} \nabla \hat{\chi}_{k,\mathbf{j}}^{KL}(y) \;\; dy \\
& + \sum_{\mathbf{j} \leq \mathbf{r}} \sum_{\mathbf{j}' \leq \mathbf{r}} \int_Y e_l \cdot \underbrace{\left(\int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z) \tilde{K}_M^{KL}(y,z) P_{\mathbf{j}'}^{\mathbf{r}}(z) \, d\rho(z) \right)}_{= I_5} e_k \, dy \\
& - 2 \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y e_l \cdot \underbrace{\left(\int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z) \tilde{K}_M^{KL}(y,z) P_{\mathbf{j}'}^{\mathbf{r}}(z) \, d\rho(z) \right)}_{= \bar{P}_{\mathbf{j}}^{\mathbf{r}} \bar{K}_M^{KL,\mathbf{j}}(y)} e_k \, dy \\
= \quad & \hat{I}_1 + \hat{I}_2 + \hat{I}_3 + \hat{I}_4 \\
& - \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y \nabla \hat{\chi}_{l,\mathbf{j}}^{KL}(y) \cdot K_M^{KL,\mathbf{j}}(y) e_k \, dy - \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y e_l \cdot K_M^{KL,\mathbf{j}}(y) \nabla \hat{\chi}_{k,\mathbf{j}}^{KL}(y) \, dy \\
& + \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y e_l \cdot K_M^{KL,\mathbf{j}}(y) e_k \, dy - 2 \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y e_l \cdot \bar{P}_{\mathbf{j}}^{\mathbf{r}} \bar{K}_M^{KL,\mathbf{j}}(y) e_k \, dy
\end{aligned}$$

If we insert the previous calculated integrals and use the notation

$$\hat{K}_{\mathrm{det}}^{KL} := \sum_{\mathbf{j} \leq \mathbf{r}} \int_Y \nabla \hat{\chi}_{l,\mathbf{j}}^{KL}(y) \cdot K_M^{KL,\mathbf{j}}(y) \nabla \hat{\chi}_{k,\mathbf{j}}^{KL}(y) \, dy,$$

we obtain

$$
\begin{aligned}
&\int_{I^M} e_l \cdot K^{*KL}(z) e_k \, d\rho(z) \\
= \quad & \hat{K}^{KL}_{\text{det}} + \delta_{lk}\, \bar{E}_K + \sum_{\mathbf{j}\leq\mathbf{r}} \bar{P}^{\mathbf{r}}_{\mathbf{j}} \int_Y \bar{K}^{KL,\mathbf{j}}_{M}(y) \left(\nabla \hat{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot e_k + \nabla \hat{\chi}^{KL}_{k,\mathbf{j}}(y) \cdot e_l\right) \, dy \\
& - \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y \nabla \hat{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_{M}(y) e_k \, dy - \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y e_l \cdot K^{KL,\mathbf{j}}_{M}(y) \nabla \hat{\chi}^{KL}_{k,\mathbf{j}}(y) \, dy \\
& + \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y e_l \cdot K^{KL,\mathbf{j}}_{M}(y) e_k \, dy - 2 \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y e_l \cdot \bar{P}^{\mathbf{r}}_{\mathbf{j}} \bar{K}^{KL,\mathbf{j}}_{M}(y) e_k \, dy \\
= \quad & \hat{K}^{KL}_{\text{det}} + \delta_{lk}\, \bar{E}_K \\
& + \sum_{\mathbf{j}\leq\mathbf{r}} \Bigg[\int_Y \nabla \hat{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot \left(\bar{P}^{\mathbf{r}}_{\mathbf{j}} \bar{K}^{KL,\mathbf{j}}_{M}(y) - K^{KL,\mathbf{j}}_{M}(y)\right) e_k \, dy \\
& \quad + \int_Y e_l \cdot \left(\bar{P}^{\mathbf{r}}_{\mathbf{j}} \bar{K}^{KL,\mathbf{j}}_{M}(y) - K^{KL,\mathbf{j}}_{M}(y)\right) \nabla \hat{\chi}^{KL}_{k,\mathbf{j}}(y) \, dy \\
& \quad + \int_Y e_l \cdot \left(K^{KL,\mathbf{j}}_{M}(y) - 2\bar{P}^{\mathbf{r}}_{\mathbf{j}} \bar{K}^{KL,\mathbf{j}}_{M}(y)\right) e_k \, dy \Bigg].
\end{aligned}
$$

Alternatively, a different ansatz is to consider for a given $\mathbf{r} \in \mathbb{N}_0^M$ for every multi index $\mathbf{j} \leq \mathbf{r}$ the deterministic cell problem

$$
\begin{aligned}
-\text{div}\left(K^{KL,\mathbf{j}}_{M}(y)\left(\nabla \tilde{\chi}'^{KL}_{i,\mathbf{j}}(y) + e_i\right)\right) &= 0,\ y \in Y, \\
\tilde{\chi}'^{KL}_{i,\mathbf{j}} &= 0,\ y \in \partial Y
\end{aligned}
\tag{5.22}
$$

For each multi index we calculate the upscaled coefficient

$$
\left(K^{*\mathbf{j}}_{KL}\right)_{lk} = \int_Y \nabla\left(\tilde{\chi}'^{KL}_{l,\mathbf{j}} + y \cdot e_l\right) K^{KL,\mathbf{j}}_{M}(y) \nabla\left(\tilde{\chi}'^{KL}_{k,\mathbf{j}} + y \cdot e_k\right) \, dy
$$

and for the stochastic equation we use the arithmetic mean

$$
K^*_{KL} = \frac{\sum_{\mathbf{j}\leq\mathbf{r}} K^{*\mathbf{j}}_{KL}}{|\mathbf{r}|} \tag{5.23}
$$

where $|\mathbf{r}|$ is the number of multi indices.

5.3. Hierarchical matrix approximation of the covariance matrix

For the discrete Karhunen-Loève expansion we have to solve the eigenproblem (5.5), i.e.,

$$
\mathbf{K}\Phi_m = \lambda^h_m \mathbf{M}\Phi_m,
$$

with

$$
\begin{aligned}
(\mathbf{K})_{ij} &= \int_{Y\times Y} \text{cov}(y, y') \varphi_j(y') \varphi_i(y) \, dy' dy, \\
(\mathbf{M})_{ij} &= \int_Y \varphi_j(y) \varphi_i(y) \, dy.
\end{aligned}
$$

The covariance matrix $\mathbf{K}$ is usually dense and requires $\mathcal{O}(n^2)$ units of memory for storage. In this section we show how to approximate a general covariance matrix with hierarchical matrices ([37]) to reduce these costs. A similar approach can be found in ([44]). In this $\mathcal{H}$-matrix technique we divide the matrix into sub blocks and determine low-rank approximations if an appropriate admissibility condition is fulfilled. We compute the low-rank approximations in linear complexity with the ACA algorithm [37].

5.3.1. Cluster theory

For the definition of hierarchical matrices we have to define trees.

Definition 5.2 (Tree[37])
Let $\mathcal{N} \neq \emptyset$ be a finite set, let $b \in \mathcal{N}$ and $S : \mathcal{N} \mapsto \mathfrak{P}(\mathcal{N})$ be a mapping from $\mathcal{N}$ into subsets of $\mathcal{N}$. For $t \in \mathcal{N}$, a sequence $t_0, \ldots, t_m \in \mathcal{N}$ with $t_0 = r$, $t_m = t$ and $t_{i+1} \in S(t_i)$ for all $i \in \{0, \ldots m-1\}$ is called sequence of ancestors of t.
$T := (\mathcal{N}, r, S)$ is called a tree if there is exactly one sequence of ancestors for each $t \in \mathcal{N}$.
If T is a tree, the elements of $\mathcal{N}$ are called nodes, the element r is called the root node or root and denoted by root(T), and the set sons$(T, t) := S(t)$ is called set of sons.

A tree has the following properties

Lemma 5.3
Let $T = (\mathcal{N}, r, S)$ be a tree.

1. *Let $t \in \mathcal{N}$, and let $t_0, \ldots t_m \in N$ be its sequence of ancestors. For all $i, j \in \{0, \ldots m\}$ with $i \neq j$, we have $t_i \neq t_j$.*
2. *There is no $t \in \mathcal{N}$ with $r \in$ sons(t).*
3. *For each $t \in \mathcal{N} \setminus \{r\}$, there is a unique $t^+ \in \mathcal{N}$ with $t \in$ sons(t^+). This node is called father of t and denoted by father$(t) := t^+$.*

Proof: [37]
A vertex $t \in \mathcal{N}$ is a leaf if sons$(t) = \emptyset$ holds and we define the set of leaves

$$\mathcal{L}(T) := \{t \in \mathcal{N} \,:\, \mathrm{sons}(t) = \emptyset\}. \tag{5.24}$$

Definition 5.4 (Tree level,[37])
Let $T = (\mathcal{N}, r, S)$ be a tree. Let $t \in \mathcal{N}$, and let $t_0, \ldots t_m \in \mathcal{N}$ be its sequence of ancestors. The number $m \in \mathbb{N}_0$ is called the level of t and denoted by level(T, t). We define

$$T^{(l)} := \{t \in \mathcal{N} \,:\, \mathit{level}(T, t) = l\} \quad \text{for all } l \in \mathbb{N}_0.$$

If and only if $t = \mathrm{root}(T)$, we have level$(t) = 0$. The maximal level is called the depth of T and denoted by

$$\mathrm{depth}(T) := \max\{\mathrm{level}(t) : t \in \mathcal{N}\}.$$

Definition 5.5 (Labeled tree, [37])
Let $\mathcal{N}$, $L \neq \emptyset$ be finite sets, let $r \in \mathcal{N}$, let $S : \mathcal{N} \mapsto \mathfrak{P}(\mathcal{N})$ and $m : \mathcal{N} \mapsto L$ be mappings. $T := (\mathcal{N}, r, S, m, L)$ is a labeled tree if $(\mathcal{N}, r, S)$ is a tree. L is called the label set of T and for each $t \in \mathcal{N}$, $m(t) \in L$ is called the label of t and denoted by $\hat{t}$.

The following definition supplies us with candidates that can be checked for admissibility. If they are not admissible, we split them and repeat the procedure.

Definition 5.6 (Cluster tree, [37])
A labeled tree $T := (\mathcal{N}, r, S, m, \mathcal{I})$ is a cluster tree for an index set $\mathcal{I}$ if the following conditions hold:

1. $\widehat{root(T)} = \mathcal{I}$.
2. *Let $t \in \mathcal{N}$. If $sons(T,t) \neq \emptyset$, we have*
$$\hat{t} = \bigcup_{s \in sons(t)} \hat{s}.$$
3. *Let $t \in \mathcal{N}$. For all s_1, $s_2 \in sons(T,t)$ with $s_1 \neq s_2$, we have $\hat{s}_1 \cap \hat{s}_2 = \emptyset$.*

The vertices $t \in \mathcal{N}$ of a cluster tree are called clusters. A cluster tree for $\mathcal{I}$ is usually denoted by $T_{\mathcal{I}}$. We will use the abbreviation $t \in T_{\mathcal{I}}$ for $t \in \mathcal{N}$.

Properties of a cluster tree are stated in the next lemma.

Lemma 5.7
Let $T_{\mathcal{I}}$ be a cluster tree.

1. *For all $t, s \in T_{\mathcal{I}}$ with $t \neq s$ and $level(t) = level(s)$, we have $\hat{t} \cap \hat{s} = \emptyset$.*
2. *For all $t, s \in T_{\mathcal{I}}$ with $level(t) \leq level(s)$ and $\hat{t} \cap \hat{s} \neq \emptyset$, we have $s \in sons^*(t)$.*
3. *For all $i \in \mathcal{I}$, there is a leaf $t \in \mathcal{L}(T_{\mathcal{I}})$ with $i \in \hat{t}$.*
4. $\mathcal{I} = \bigcup_{t \in \mathcal{L}(T_{\mathcal{I}})} \hat{t}$.

Thereby $\text{sons}^*(t)$ denotes the set of descendants of a node $t \in \mathcal{N}$

$$\text{sons}^*(t) := \begin{cases} \{t\} & \text{if } \text{sons}(t) = \emptyset, \\ \{t\} \cup \bigcup_{s \in \text{sons}(t)} \text{sons}^*(s) & \text{otherwise.} \end{cases} \tag{5.25}$$

Proof: [37].
The next step is to assign suitable methods to construct such a cluster tree. In the following we describe two different algorithms.

5.3.2. Geometric bisection

For each index $i \in \mathcal{I}$ we choose, for simplicity, one point x_i of the support of the corresponding basis function. We start with the full index set $\mathcal{I}$, which is the root of the cluster tree by definition. To split the index set we define

$$a_l := \min\{(x_i)_l : i \in \hat{t}\} \text{ and } b_l := \max\{(x_i)_l : i \in \hat{t}\}$$

for all $l \in \{1, \dots d\}$. Therefore all points are contained in the axis-parallel box $[a_1, b_1] \times \cdots \times [a_d, b_d]$. Now we split the box perpendicular to the direction of maximal extent.

5.3.3. Regular subdivision

As before, we construct the cluster tree by defining how a cluster t is split. We assume that a box $B_t = [a_1, b_1] \times \cdots \times [a_d, b_d]$ with $x_i \in B_t$ for all $i \in \hat{t}$ and a splitting direction $j_t \in \{1, \ldots, d\}$ are given. We construct new boxes B_{t_0} and B_{t_1} by setting $c_j := (a_j + b_j)/2$ and

$$B_{t_0} = [a_1, b_1] \times \cdots \times [a_j, c_j] \times \cdots \times [a_d, b_d] \text{ and } B_{t_1} = [a_1, b_1] \times \cdots \times [c_j, b_j] \times \cdots \times [a_d, b_d].$$

The index sets $\hat{t}_0$ and $\hat{t}_1$ are defined by

$$\hat{t}_0 := \{i \in \hat{t} : x_i \in B_{t_0}\} \text{ and } \hat{t}_1 := \hat{t} \backslash \hat{t}_0.$$

We set $j_{t_0} = j_{t_1} = (j_t \mod d) + 1$. Since $x_i \in B_{t_0}$ for $i \in \hat{t}_0$ and $x_i \in B_{t_1}$ for $i \in \hat{t}_1$ hold by construction, we can repeat the procedure for t_0 and t_1.

5.3.4. Block cluster tree

With two cluster trees we can derive a hierarchy of block partitions of $\mathcal{I} \times \mathcal{J}$ corresponding to the matrix, the block cluster tree.

Definition 5.8 (Block cluster tree, [37])
Let $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ be cluster trees for index sets $\mathcal{I}$ and $\mathcal{J}$. A finite tree T is a block cluster tree for $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ if the following conditions hold:

1. *$root(T) = (root(T_{\mathcal{I}}), root(T_{\mathcal{J}}))$.*

2. *Each node $b \in T$ has the form $b = (t, s)$ for clusters $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{J}}$.*

3. *For each node $b = (t, s) \in T$ with $sons(b) \neq \emptyset$, we have*

$$sons(b) = \begin{cases} \{(t, s') : s' \in sons(s)\} & \text{if } sons(t) = \emptyset \text{ and} \\ & \quad sons(s) \neq \emptyset, \\ \{(t', s) : t' \in sons(t)\} & \text{if } sons(t) \neq \emptyset \text{ and} \\ & \quad sons(s) = \emptyset, \\ \{(t', s') : t' \in sons(t),\ s' \in sons(s)\} & \text{otherwise.} \end{cases} \tag{5.26}$$

4. *The label of a node $b = (t, s) \in T$ is given by $\hat{b} = \hat{t} \times \hat{s} \subseteq \mathcal{I} \times \mathcal{J}$.*

The vertices of T are called block clusters. A block cluster tree for $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ is usually denoted by $T_{\mathcal{I} \times \mathcal{J}}$.

Obviously $\widehat{\mathrm{root}(T_{\mathcal{I} \times \mathcal{J}})} = \mathcal{I} \times \mathcal{J}$ holds.

Definition 5.9 (Level-consistency, [37])
A block cluster tree $T_{\mathcal{I} \times \mathcal{J}}$ for $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ will be called level-consistent, if

$$level(b) = level(t) = level(s)$$

holds for all $b = (t, s) \in T_{\mathcal{I} \times \mathcal{J}}$.

It follows that only the last case in (5.26) can hold, if a block cluster tree is level-consistent.

5.3.5. Admissibility

We generalize the supports of the basis functions φ_i to clusters $t \in T_{\mathcal{I}}$

$$Q_t := \bigcup_{i\in\hat{t}} \mathrm{supp}(\varphi_i),$$

i.e., Q_t is the minimal subset of $\mathbb{R}^d$ that contains the supports of all basis functions φ_i with $i \in \hat{t}$. One possible admissibility condition is

$$\min\{\mathrm{diam}(Q_t), \mathrm{diam}(Q_s)\} \le \eta\, \mathrm{dist}(Q_t, Q_s), \tag{5.27}$$

where diam$(\cdot)$ is the Euclidian diameter of a set and dist$(\cdot)$ is the Euclidian distance of two sets.

Definition 5.10 (Admissible block cluster tree, [37])
A block cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ for $\mathcal{I}$ and $\mathcal{J}$ is called admissible with respect to an admissibility condition if

$$(t,s) \textit{ is admissible} \quad \textit{or} \quad \textit{sons}(t) = \emptyset \quad \textit{or} \quad \textit{sons}(s) = \emptyset$$

holds for all leaves $(t,s) \in \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})$.

We construct an admissible block cluster tree recursively. For two given clusters $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{J}}$ we check the admissibility. If they are admissible, we are done. Otherwise we repeat the procedure with all combinations of sons of t and s. However, in general it can be too expensive to check an admissibility condition like (5.27). A traditional way is to determine the Chebyshev circles for the domain, but we consider a simpler approach: Axis-parallel boxes.
For each cluster $t \in T_{\mathcal{I}}$ we define an axis-parallel bounding box $\tilde{Q}_t \subseteq \mathbb{R}^d$such that $Q_t \subset \tilde{Q}_t$ holds. Now we consider the admissibility condition

$$\min\{\mathrm{diam}(\tilde{Q}_t), \mathrm{diam}(\tilde{Q}_s)\} \le \eta\, \mathrm{dist}(\tilde{Q}_t, \tilde{Q}_s).$$

If this condition is fulfilled, also (5.27) holds.
The above admissibility condition is the standard admissibility condition. Another important admissibility condition is the so-called weak admissibility condition:

$$\tilde{Q}_t \neq \tilde{Q}_s \tag{5.28}$$

with $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{J}}$.

5.3.6. Low-rank approximation

The admissibility condition indicates blocks of the matrix $\mathbf{K}$ which allow rank k approximation and blocks where we have to calculate the exact entries of the matrix

$$\mathbf{K'}_{ij} = \begin{cases} \tilde{\mathbf{R}}_{ij} & \text{if } i \in \hat{t},\ j \in \hat{s} \text{ and } (b,s) \in T_{\mathcal{I}\times\mathcal{J}} \text{ admissible} \\ \mathbf{K}_{ij} & \text{otherwise,} \end{cases} \tag{5.29}$$

with the low-rank approximation $\tilde{\mathbf{R}} = \mathbf{A}\mathbf{B}^T \in \mathbb{R}^{\hat{t}\times\hat{s}}$ and $\mathbf{A} \in \mathbb{R}^{\hat{t}\times k}$ and $\mathbf{B} \in \mathbb{R}^{\hat{s}\times k}$, $k \in \mathbb{N}$. Note that any matrix of rank k can be represented in this form. One possibility to compute the low-rank approximation is the improved adaptive cross approximation algorithm, which one can find in Algorithm 1.

Algorithm 1 Improved adaptive cross approximation ACA+

We denote the already used pivot elements by $\mathcal{P}_{rows}$ and $\mathcal{P}_{cols}$. Denote the matrix to be approximated by $\mathbf{R} \in \mathbb{R}^{\hat{t}\times\hat{s}}$. A reference column of $\mathbf{R}$ is defined by

$$(a^{ref})_i := \mathbf{R}_{i,j_{ref}}, \quad i \in \{1,\ldots,\hat{t}\}, \quad j_{ref} \text{ arbitrary.}$$

The reference row is determined by

$$\begin{aligned} i_{ref} &:= \operatorname{argmin}_{i\in\{1,\ldots,\hat{t}\}} |(a^{ref})_i|, \\ (b^{ref})_j &:= \mathbf{R}_{i_{ref},j}, \quad j \in \{1,\ldots,\hat{s}\}. \end{aligned}$$

```
1. Determine the index $i^*$ of the largest entry in modulus in $a^{ref}$,
   $i^* := \operatorname{argmax}_{i\in\{1,\ldots,\hat{t}\}\setminus\mathcal{P}_{rows}} |a_i^{ref}|$.
2. Determine the index $j^*$ of the largest entry in modulus in $b^{ref}$,
   $j^* := \operatorname{argmax}_{j\in\{1,\ldots,\hat{s}\}\setminus\mathcal{P}_{cols}} |b_j^{ref}|$.
3. If $|a_{i^*}^{ref}| > |b_{j^*}^{ref}|$
   a) compute the vector $b^k \in \mathbb{R}^{\hat{s}}$ with entries $b_j^k := \mathbf{R}_{i^*,j}$;
   b) redefine the column pivot index $j^* = \operatorname{argmax}_{j\in\{1,\ldots,\hat{s}\}\setminus\mathcal{P}_{cols}} |b_j^k|$;
   c) compute the vector $a^k \in \mathbb{R}^{\hat{t}}$ with entries $a_i^k := \mathbf{R}_{i,j^*}/R_{i^*,j^*}$;
4. otherwise
   a) compute the vector $a^k \in \mathbb{R}^{\hat{t}}$ with entries $a_i^k := \mathbf{R}_{i,j^*}$;
   b) redefine the row pivot index $i^* = \operatorname{argmax}_{i\in\{1,\ldots,\hat{t}\}\setminus\mathcal{P}_{rows}} |a_i^k|$;
   c) compute the vector $b^k \in \mathbb{R}^{\hat{s}}$ with entries $b_j^k := \mathbf{R}_{i^*,j}/\mathbf{R}_{i^*,j^*}$.
5. Check the stopping criterion: If $\|a^k\|_2\|b^k\|_2/\|a^1\|_2\|b^1\|_2 \le \epsilon_{ACA}$ or $k = k_{max}$ stop.
6. $\mathcal{P}_{rows} = \mathcal{P}_{rows} \cup \{i^*\}$ and $\mathcal{P}_{cols} = \mathcal{P}_{cols} \cup \{j^*\}$.
7. Update reference column
   a) if $j^* \neq j^{ref}$: $a^{ref} := a^{ref} - a^k \cdot b_{j_{ref}}^k$
   b) otherwise we have to choose a new reference column corresponding to a reference
      index $j_{ref}$ that has not been a pivot index and that is consistent to $i_{ref}$:
      $j_{ref} = \begin{cases} j \in \{1,\ldots\hat{s}\} \setminus \mathcal{P}_{cols}, & \text{if } i^* = i^{ref} \\ \operatorname{argmin}_{j\in\{1,\ldots\hat{s}\}\setminus\mathcal{P}_{cols}} |b_j^{ref}|, & \text{else} \end{cases}$
      $(a^{ref})_i := \mathbf{R}_{i,j_{ref}}, \quad i \in \{1,\ldots,\hat{t}\}$
8. Update reference row
   a) if $i^* \neq i^{ref}$: $b^{ref} := b^{ref} - a_{i_{ref}}^k \cdot b^k$
   b) otherwise we have to choose a new reference row corresponding to a reference
      index $i_{ref}$ that has not been a pivot index and that is consistent to $j_{ref}$:
      $i_{ref} = \operatorname{argmin}_{i\in\{1,\ldots\hat{t}\}\setminus\mathcal{P}_{rows}} |a_i^{ref}|$, $(b^{ref})_j := \mathbf{R}_{i_{ref},j}, \quad j \in \{1,\ldots,\hat{s}\}$
9. Update the matrix: $\mathbf{R} = \mathbf{R} - a^k(b^k)^T$.
10. Update the rank: $k = k + 1$ and go back to 1.
```

6. Statement of the problems

In this section we describe the implemented problems: the microscale problem, the cell problem, the homogenization problem, the reconstruction problem and the problem with a stochastic coefficient. For the stochastic problem we consider two different distributions, the first one is a Gaussian distribution, therefore the covariance looks like

$$\mathrm{cov}_G(y, y') = \sigma^2 \exp\left(\frac{-|y-y'|^2}{\tau^2} \right).$$

Here, σ denotes the standard deviation and τ is proportional to the correlation length. The second one is a lognormal distribution. A lognormal distribution is a distribution of a random variable whose logarithm is normally or Gaussian distributed. If X is a random variable with a normal distribution, then $Y = \exp(X)$ has a lognormal distribution. If σ, μ and $\mathrm{cov}_G(y, y')$ denote the standard deviation, the expected value and the covariance of a Gaussian distributed random variable, respectively, then it holds for the lognormal distribution

$$\begin{array}{lll} \text{expected value:} & \mu_{log} & = \exp(\mu + \frac{\sigma^2}{2}) \\ \text{variance:} & \Sigma^2 & = \mu_{log}^2 \left(\exp(\sigma^2) - 1 \right) \\ \text{covariance:} & \mathrm{cov}_{log}(y, y') & = \mu_{log}^2 \left(\exp(\mathrm{cov}_G(y, y')) - 1 \right). \end{array}$$

In our example we consider the domain $D := (0,1)^2$ with squared obstacles placed at the center of each unit cell Y as illustrated in Figure 2.

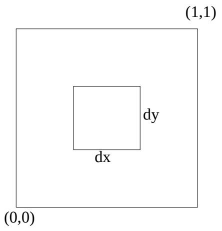


Figure 2: Unit cell Y with an obstacle of volume $dx \cdot dy$.

Microscale problem If we choose a periodic piecewise constant diffusion coefficient, which fulfills in the periodicity cell

$$K = \begin{cases} \alpha, & \text{inside the obstacle} \\ 1, & \text{else} \end{cases}$$

and Dirichlet boundary conditions, the microscale problem reads

$$\begin{aligned} -\mathrm{div}\,(K \nabla u_\epsilon) &= f, \quad x \in D \\ u_\epsilon &= g, \quad x \in \partial D. \end{aligned} \tag{6.1}$$

Additionally we consider a problem with mixed boundary conditions

$$\begin{aligned} -\mathrm{div}\,(K \nabla u_\epsilon) &= f, \quad x \in D \\ (K \nabla u_\epsilon) \cdot n &= \varphi, \quad x \in \partial D^N \\ u_\epsilon &= g, \quad x \in \partial D^D. \end{aligned} \tag{6.2}$$

The direct numerical simulation is difficult due to the fine scale heterogeneity. Therefore we consider the homogenized problem, additionally.

Cell problem In each direction we solve the cell problem:

$$\begin{aligned}-\mathrm{div}\,(K\nabla\chi_i) &= 0, \quad y\in Y\\ \chi_i &= y_i, \quad y\in\partial Y\end{aligned}$$

Homogenization problem With the cell solution χ_i we determine the upscaled coefficient $K^* = (K^*_{ij})_{1\le i,j\le 2}$ as follows

$$K^*_{ij} = \int_Y \nabla\chi_i\cdot K\nabla\chi_j\,dy.$$

So we obtain the following problem:

$$\begin{aligned}-\mathrm{div}\,(K^*\nabla u^*) &= f, \quad x\in D\\ u^* &= g, \quad x\in\partial D.\end{aligned}$$

K^* is a symmetric and positive definite matrix, which we calculate numerically.

Reconstruction To compare the coarse solution u^* and the fine scale solution u_ϵ, we reconstruct a fine scale approximation as follows (cf. [41])

$$-\mathrm{div}\,(K\nabla\hat{u}) = 0 \text{ in V}$$

with the boundary condition

$$\hat{u} = u^{edge} \text{ on } \partial\text{V}.$$

On $\partial D\cap\partial$V we use the global boundary conditions. Here V denotes a dual coarse grid block where the nodes of V are four neighboring cell centers of the original coarse grid and u^{edge} the solution of the four one-dimensional problems:

$$\begin{aligned}\frac{\partial}{\partial x_1}\left(K\frac{\partial u^{edge}}{\partial x_1}\right) &= 0, \text{ in } \Gamma_2,\ u^{edge}(x) = u^*(x) \text{ in } \Gamma_1\cap\Gamma_2\\ \frac{\partial}{\partial x_2}\left(K\frac{\partial u^{edge}}{\partial x_2}\right) &= 0, \text{ in } \Gamma_1,\ u^{edge}(x) = u^*(x) \text{ in } \Gamma_1\cap\Gamma_2\end{aligned}$$

with $\Gamma_1\cup\Gamma_2 = \partial V$ and Γ_i is the part of the boundary which is orthogonal to the unit vector e_i in the i^{th} direction.
We solve these one-dimensional problems analytically.

Stochastic problem For the stochastic equation

$$\begin{aligned}-\mathrm{div}\left(K(\frac{x}{\epsilon},\omega)\nabla u_\epsilon(x,\omega)\right) &= f(x) \quad x\in D\\ u_\epsilon &= g \qquad \text{at } \partial D\end{aligned}$$

P -a.e. $\omega\in\Omega$ we assume, in the case of normal distributed random variables, that the expected value of the stochastic coefficient is equal to the deterministic coefficient, i.e.,

$$E_K = \begin{cases}\alpha, & \text{inside the obstacle}\\ 1, & \text{else}\end{cases}$$

and as mentioned above we consider the Gaussian covariance function

$$\mathrm{cov}_G(y,y') = \sigma^2 \exp\left(\frac{-|y-y'|^2}{\tau^2}\right)$$

and therefore for the lognormal distribution we get ($\mu = 0$)

$$\mathrm{cov}_{log}(y,y') = \exp(\sigma^2)\left(\exp\left(\sigma^2 \exp\left(\frac{-|y-y'|^2}{\tau^2}\right)\right) - 1\right).$$

In the example with the lognormal distribution we consider two different cases. In one case we assume as in the Gaussian distribution, that the expected value is equal to the deterministic coefficient and in the other we use the exact mean $\mu_{log} = \exp(\frac{\sigma^2}{2})$.
Numerically we solve the deterministic equation as follows. First we assemble the matrix and the right-hand side with the cell-centered finite volume method, which we describe in the following section and thereafter we solve the resulting system of equations with the cg method. We describe the generation of the random variables which is needed in the Monte Carlo simulation in Section 8.

7. Cell-centered finite volume method

In this section we derive the cell-centered finite volume method either with a scalar coefficient K or if K is a symmetric and positive definite matrix.

7.1. Scalar coefficient

To derive the cell-centered finite volume method, if the coefficient is scalar, we integrate equation (6.1) over a cell C$_i$. We allow Dirichlet and Neumann boundary conditions.
Applying the divergence theorem to the left-hand side, we get:

$$\begin{aligned}
- & \int_{C_i} \mathrm{div}\,(K\nabla u)\; dx \\
= & \int_{\partial C_i} -(K\nabla u)\cdot \nu \, ds \\
= & \sum_{j\in N_i} \int_{\gamma_{ij}} -(K\nabla u)\cdot \nu_{ij}\, ds + \sum_{j\in B_i} \int_{\gamma_{ij}} -(K\nabla u)\cdot \nu_{ij}\, ds \\
\approx & \sum_{j\in N_i} -v_{ij} K(\bar{x}_{ij}) \frac{u(\bar{x}_j) - u(\bar{x}_i)}{h_j} + \sum_{j\in B_i^D} -v_{ij} K(\bar{x}_{ij}) \frac{g(\bar{x}_{ij}) - u(\bar{x}_i)}{h_{ij}} \\
& + \sum_{j\in B_i^N} -v_{ij}\varphi(\bar{x}_{ij})
\end{aligned}$$

and to the right-hand side:

$$\int_{C_i} f\, dx \approx V_i f(\bar{x}_i).$$

We obtain the system

$$AU = F$$

with $U_i = u(\bar{x}_i)$,

$$F_i = V_i f(\bar{x}_i) + \sum_{j \in B_i^{\mathrm{D}}} \frac{v_{ij}}{h_{ij}} K(\bar{x}_{ij})\, g(\bar{x}_{ij}) + \sum_{j \in B_i^{\mathrm{N}}} v_{ij} \varphi(\bar{x}_{ij}),$$

$$A_{ii} = \sum_{j \in N_i} \frac{v_{ij}}{h_j} K(\bar{x}_{ij}) + \sum_{j \in B_i^{\mathrm{D}}} \frac{v_{ij}}{h_{ij}} K(\bar{x}_{ij})$$

and

$$A_{ij} = \begin{cases} -\frac{v_{ij}}{h_j} K(\bar{x}_{ij}), & \text{if } j \in N_i \\ 0, & \text{else} \end{cases}.$$

If the right-hand side has the following form

$$f = \operatorname{div}(K(x)\, e_k)$$

we approximate $\int_{C_i} f\, dx$ as follows

$$\int_{C_i} f\, dx \approx \sum_{j \in N_i} K(\bar{x}_{ij}) \frac{(\bar{x}_j)_k - (\bar{x}_i)_k}{h_j} + \sum_{j \in B_i} K(\bar{x}_i) \frac{(\bar{x}_{ij})_k - (\bar{x}_i)_k}{h_{ij}}$$

The following notations are used:

C_i	cell of the grid with number i
γ_{ij}	face common to C_i and C_j
V_i	volume of cell i
v_{ij}	volume γ_{ij} (or boundary intersection)
$\bar{x}_i$	cell center of cell i
$\bar{x}_{ij}$	center of the intersection
ν	outer normal
ν_{ij}	normal pointing from C_i to C_j
h_j	$= \sum_{k=1}^{2} \lvert (\bar{x}_i)_k - (\bar{x}_j)_k \rvert$
h_{ij}	$= \sum_{k=1}^{2} \lvert (\bar{x}_i)_k - (\bar{x}_{ij})_k \rvert$
N_i	set of the global numbers of the neighbors of cell i
B_i^D	index set of boundary intersections with Dirichlet condition
B_i^N	index set of boundary intersections with Neumann condition
B_i	$= B_i^D + B_i^N$
$K(\bar{x}_{ij})$	$= \frac{2}{\frac{1}{K(\bar{x}_i)} + \frac{1}{K(\bar{x}_j)}}$, harmonic mean.

7.2. Matrix coefficient

Let

$$K = \begin{pmatrix} K_1 & K_2 \\ K_2 & K_1 \end{pmatrix}$$

be constant and positive definite. For the sake of simplicity we consider in this section only Dirichlet boundary conditions. Then we derive analogously, if C_i has less than two Dirichlet boundary edges:

$$A_{ii} = \sum_{j \in N_i} \frac{v_{ij}}{h_j} K_1 + \sum_{j \in B_i^{\mathrm{D}}} \frac{v_{ij}}{h_{ij}} K_1.$$

If C_i is either the bottom left or the upper right corner cell with two Dirichlet boundary segments, we get:

$$A_{ii} = \sum_{j \in N_i} \frac{v_{ij}}{h_j} K_1 + \sum_{j \in B_i^{\mathrm{D}}} \frac{v_{ij}}{h_{ij}} K_1 - \frac{K_2}{2}$$

and if it is either the bottom right or the upper left corner cell with two Dirichlet boundary segments, we get:

$$A_{ii} = \sum_{j \in N_i} \frac{v_{ij}}{h_j} K_1 + \sum_{j \in B_i^{\mathrm{D}}} \frac{v_{ij}}{h_{ij}} K_1 + \frac{K_2}{2}.$$

Let n_x be the number of cells in x-direction and $x_i^{ul}, x_i^{ur}, x_i^{bl}$ and x_i^{br} the upper left node, the upper right node, the bottom left node and the bottom right node of cell C_i, respectively, then we get, if C_i is an interior cell:

$$\begin{aligned}
A_{ii+1} &= -\frac{v_{ii+1}}{h_{i+1}} K_1, \\
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}} K_1, \\
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}} K_1, \\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}} K_1, \\
\\
A_{ii+n_x+1} &= -\frac{K_2}{2}, \\
A_{ii+n_x-1} &= \frac{K_2}{2}, \\
A_{ii-n_x+1} &= \frac{K_2}{2}, \\
A_{ii-n_x-1} &= -\frac{K_2}{2}, \\
\\
F_i &= V_i f\left(\bar{x}_i\right).
\end{aligned}$$

If C_i is located at the left boundary with Dirichlet boundary conditions, we get:

$$\begin{aligned}
A_{ii+1} &= -\frac{v_{ii+1}}{h_{i+1}} K_1, \\
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}} K_1 - \frac{K_2}{2}, \\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}} K_1 + \frac{K_2}{2}, \\
\\
A_{ii+n_x+1} &= -\frac{K_2}{2}, \\
A_{ii-n_x+1} &= \frac{K_2}{2}, \\
\\
F_i &= V_i f\left(\bar{x}_i\right) + \frac{v_{ii-1}}{h_{ii-1}} K_1 g(\bar{x}_{ii-1}) - 2K_2 g(x_i^{ul}) + 2K_2 g(x_i^{bl}).
\end{aligned}$$

If C_i is located at the right boundary with Dirichlet boundary conditions, we get:

$$\begin{aligned}
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}}K_1,\\
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}}K_1+\frac{K_2}{2},\\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}}K_1-\frac{K_2}{2},\\
A_{ii+n_x-1} &= \frac{K_2}{2},\\
A_{ii-n_x-1} &= -\frac{K_2}{2},\\
F_i &= V_i f\left(\bar{x}_i\right)+\frac{v_{ii+1}}{h_{ii+1}}K_1 g(\bar{x}_{ii+1})+2K_2 g(x_i^{ur})-2K_2 g(x_i^{br}).
\end{aligned}$$

If C_i is located at the upper boundary with Dirichlet boundary conditions, we get:

$$\begin{aligned}
A_{ii+1} &= -\frac{v_{ii+1}}{h_{i+1}}K_1+\frac{K_2}{2},\\
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}}K_1-\frac{K_2}{2},\\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}}K_1,\\
A_{ii-n_x+1} &= \frac{K_2}{2},\\
A_{ii-n_x-1} &= -\frac{K_2}{2},\\
F_i &= V_i f\left(\bar{x}_i\right)+\frac{v_{ii+n_x}}{h_{ii+n_x}}K_1 g(\bar{x}_{ii+n_x})+2K_2 g(x_i^{ur})-2K_2 g(x_i^{ul}).
\end{aligned}$$

If C_i is located at the lower boundary with Dirichlet boundary conditions, we get:

$$\begin{aligned}
A_{ii+1} &= -\frac{v_{ii+1}}{h_{i+1}}K_1-\frac{K_2}{2},\\
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}}K_1+\frac{K_2}{2},\\
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}}K_1\\
A_{ii+n_x+1} &= -\frac{K_2}{2},\\
A_{ii+n_x-1} &= \frac{K_2}{2},\\
F_i &= V_i f\left(\bar{x}_i\right)+\frac{v_{ii-n_x}}{h_{ii-n_x}}K_1 g(\bar{x}_{ii-n_x})-2K_2 g(x_i^{br})+2K_2 g(x_i^{bl}).
\end{aligned}$$

If C_i is the bottom left corner and both edges have Dirichlet boundary conditions, then we get:

$$A_{ii+1} = -\frac{v_{ii+1}}{h_{i+1}}K_1-\frac{K_2}{2},$$

$$
\begin{aligned}
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}}K_1 - \frac{K_2}{2}, \\
A_{ii+n_x+1} &= -\frac{K_2}{2}, \\
F_i &= V_i f\left(\bar{x}_i\right) + \frac{v_{ii-1}}{h_{ii-1}}K_1 g(\bar{x}_{ii-1}) + \frac{v_{ii-n_x}}{h_{ii-n_x}}K_1 g(\bar{x}_{ii-n_x}) \\
&\quad +2K_2 g(x_i^{bl}) - 2K_2 g(x_i^{br}) - 2K_2 g(x_i^{ul}).
\end{aligned}
$$

If C_i is the bottom right corner and both edges have Dirichlet boundary conditions, then we get:

$$
\begin{aligned}
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}}K_1 + \frac{K_2}{2}, \\
A_{ii+n_x} &= -\frac{v_{ii+n_x}}{h_{i+n_x}}K_1 + \frac{K_2}{2}, \\
A_{ii+n_x-1} &= \frac{K_2}{2}, \\
F_i &= V_i f\left(\bar{x}_i\right) + \frac{v_{ii+1}}{h_{ii+1}}K_1 g(\bar{x}_{ii+1}) + \frac{v_{ii-n_x}}{h_{ii-n_x}}K_1 g(\bar{x}_{ii-n_x}) \\
&\quad +2K_2 g(x_i^{ur}) - 2K_2 g(x_i^{br}) + 2K_2 g(x_i^{bl}).
\end{aligned}
$$

If C_i is the upper right corner and both edges have Dirichlet boundary conditions, then we get:

$$
\begin{aligned}
A_{ii-1} &= -\frac{v_{ii-1}}{h_{i-1}}K_1 - \frac{K_2}{2}, \\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}}K_1 - \frac{K_2}{2}, \\
A_{ii-n_x-1} &= -\frac{K_2}{2}, \\
F_i &= V_i f\left(\bar{x}_i\right) + \frac{v_{ii+1}}{h_{ii+1}}K_1 g(\bar{x}_{ii+1}) + \frac{v_{ii+n_x}}{h_{ii+n_x}}K_1 g(\bar{x}_{ii+n_x}) \\
&\quad +2K_2 g(x_i^{ur}) - 2K_2 g(x_i^{br}) - 2K_2 g(x_i^{ul}).
\end{aligned}
$$

If C_i is the upper left corner and both edges have Dirichlet boundary conditions, then we get:

$$
\begin{aligned}
A_{ii+1} &= -\frac{v_{ii+1}}{h_{i+1}}K_1 + \frac{K_2}{2}, \\
A_{ii-n_x} &= -\frac{v_{ii-n_x}}{h_{i-n_x}}K_1 + \frac{K_2}{2}, \\
A_{ii-n_x+1} &= \frac{K_2}{2}, \\
F_i &= V_i f\left(\bar{x}_i\right) + \frac{v_{ii-1}}{h_{ii-1}}K_1 g(\bar{x}_{ii-1}) + \frac{v_{ii+n_x}}{h_{ii+n_x}}K_1 g(\bar{x}_{ii+n_x}) \\
&\quad -2K_2 g(x_i^{ul}) + 2K_2 g(x_i^{bl}) + 2K_2 g(x_i^{ur}).
\end{aligned}
$$

8. Generation of random variables

In the following section we describe how we generate the random variables. The main issue is the generation of the normal distribution. As mentioned before, if X is a random variable with a Gaussian distribution, then $Y = \exp(X)$ has a lognormal distribution. We start with two in $(0, 1]$ uniformly distributed independent random variables U_1 and U_2. With the Box-Muller transformation we determine an independent random variable with standard normal distribution via

$$Z_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)\,.$$

Assume we have a vector of standard normal random variables $Z = (Z_1, Z_2, \ldots, Z_N)$. Let Q denote the matrix of eigenvectors of the covariance matrix C_K of the given covariance function $\mathrm{cov}_G(y, y')$ and Λ the corresponding diagonal matrix of eigenvalues, then $X = AZ$ with $A = Q\sqrt{\Lambda}$, is a vector of Gaussian distributed random variables with expected value 0 and covariance $\mathrm{cov}_G(y, y')$.

9. Numerical results

In this section we present numerical results for the methods described in Section 6. We start with deterministic examples with only one scale where the exact solution is known to verify the implemented finite volume method (cf. Sec. 9.1). In Section 9.2 we compute the effective coefficients for deterministic problems and we compare the reconstructed coarse solution with the fine-scale reference solution. In Section 9.3 we compare the two introduced approaches to approximate the effective tensor in a stochastic setting. We consider different distributions and in Section 9.4 we discuss the use of hierarchical matrices to compute the Karhunen-Loève expansion.

All implementations are based on the software package DUNE ([12, 11, 24, 14, 25]), the eigenvalue problem we solve with ARPACK ([1]) and for the hierarchical matrix computation we use HLib ([15]).

9.1. Finite volume test

To verify the finite volume method we have implemented two test problems, for which the analytical solutions are known. The first one is symmetric in both directions. We consider

$$\begin{aligned} -\Delta u &= 2\pi^2 \sin(\pi x_1)\sin(\pi x_2) \;\text{ in } D = (0,1)^2 \\ u &= 0 \text{ at } \partial D \end{aligned}$$

with the analytical solution

$$u = \sin(\pi x_1)\sin(\pi x_2)\,.$$

The second problem is asymmetric, it reads

$$\begin{aligned} -\Delta u &= 10\pi^2 \sin(\pi x_1)\sin(3\pi x_2) \;\text{ in } D = (0,1)^2 \\ u &= 0 \text{ at } \partial D. \end{aligned}$$

In the case of the asymmetric test the analytical solution is

$$u = \sin(\pi x_1)\sin(3\pi x_2)\,.$$

In Figure 3 we illustrate the convergence behavior in the L^2-norm for both test problems when we refine the grid. In a logarithmic scale we see linear convergence.

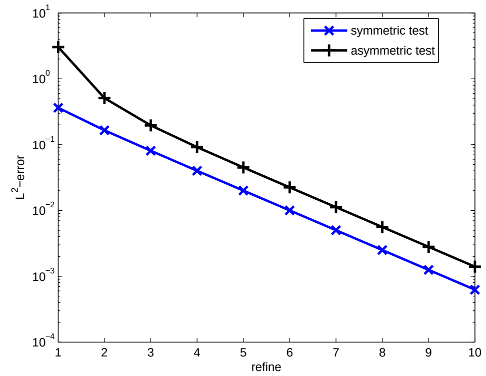


Figure 3: L^2-error between approximation and exact solution (compare Section 9.1).

9.2. Deterministic equations

Here we consider the deterministic microscale problems (6.1) and (6.2). In the case of both, Neumann and Dirichlet boundary conditions, we use $\partial D^D = \{x = (x_1, x_2) \in \partial D$ with $x_1 = 0$ or $x_1 = 1\}$ and $\partial D^N = \partial D \setminus \partial D^D$.

In our simulations we use

$$\begin{aligned} f(x) &= x_1, \\ g(x) &= x_1 \\ \varphi(x) &= 0 \\ K(y) &= \begin{cases} \alpha, & \text{if } y \in O = [0.45, 0.55]^2 \\ 1, & \text{if } y \in [0,1]^2 \backslash O, \end{cases} \end{aligned}$$

i.e., the obstacle O in the unit cell is a square of length 0.1. For α we use either $\alpha = 0.1$ or $\alpha = 20$.

For both αs we calculated the upscaled coefficient on a grid with 1048576 cells. For $\alpha = 0.1$ we have

$$K^*_{0.1} = \begin{pmatrix} 0.982998 & -9.24133e-10 \\ -9.24133e-10 & 0.982998 \end{pmatrix}$$

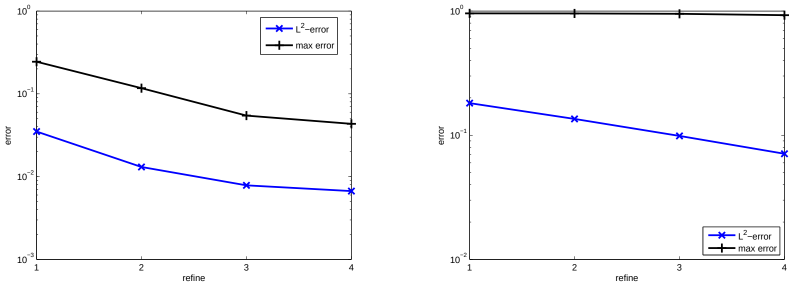
and for $\alpha = 20$

$$K^*_{20} = \begin{pmatrix} 1.01996 & 6.92546e-10 \\ 6.92546e-10 & 1.01996 \end{pmatrix}.$$

For these coefficients we calculate the coarse solution with 16 × 16 coarse blocks and after reconstruction we compute the L^2-error and the maximum error of the approximation and the fine scale solution (262144 cell) as illustrated in Figure 4.

9.3. Stochastic equations

In this section we approximate the expected value of the upscaled coefficient via Monte Carlo simulation and Karhunen-Loève expansion as described above and compare them with the effective coefficient of a deterministic problem with E_K as microscale coefficient with the same grid size (1024 cells). In the case of the Monte Carlo simulation we use two different stopping criteria. The first one calculates the error of the diagonal entries of two sequenced means, e.g., (MC1):



(a) Dirichlet B.C, $\alpha = 0.1$ (b) mixed B.C., $\alpha = 20$

Figure 4: Error between the reconstructed coarse solution and a fine scale reference solution (262144 cells, 16×16 coarse blocks) with the boundary conditions described above.

$|k_N - k_{N-1}| \leq \epsilon_{tol} := 0.001$ with $k_N = \left((K_N^{*MC})_{11}, (K_N^{*MC})_{22}\right)^T$. The other one is (MC2): $N > \frac{\mathrm{Var}(K_N^{*MC})}{\epsilon_{tol}^2}$. Thereby, N denotes the number of realizations. However, there is no guaranty that the first condition (MC1) leads to a reasonable coefficient. Two sequenced values could be only close to each other, but far away from the mean value. In the other case the number of solved problems depends on the truncation order of the Karhunen-Loève expansion. The truncation condition is $\lambda \leq \epsilon_{tol}$. Here, with KL1 we denote the upscaled coefficient via the arithmetic mean (cf. (5.23)). KL2 denotes the coefficient from equation (5.20). α is either set to 1 or to 0.1. The deterministic coefficient and the expected value are equal to 2 in an additional example. In the following we consider Gaussian distributed random variables and after that random variables with a lognormal distribution.

9.3.1. Gaussian distributed random variables

In the case of Gaussian distributed random variables we consider the following type of covariance function

$$\mathrm{cov}_G(y, y') = \sigma^2 \exp\left(\frac{-|y - y'|^2}{\tau^2}\right). \tag{9.1}$$

To achieve the Karhunen-Loève expansion we have to determine the eigenpairs of the covariance operator first. The 20 largest eigenvalues of the covariance operator are shown in Figure 5. The eigenvalues decay very fast to zero, in the Gaussian case we have an exponential decay order. This decay of the eigenvalues is a necessary condition for the convergence of the Karhunen-Loève expansion.

In our calculations we use different τs ($\tau = 1$ or $\tau = 0.5$) and different standard deviations σs as well ($\sigma = 0.0001, 0.001, 0.01, 0.1$).

In Table 28 we present the calculated upscaled coefficient for the above mentioned stopping criteria for $E_K(y) = 1$ and $\tau = 1$. In the Tables 29, 30 and 31 we consider the cases $E_K(y) = 2$ and $\tau = 1$, $E_K(y) = 1$ and $\tau = 0.5$ and $E_K(y) = 2$ and $\tau = 0.5$, respectively.

In Table 27 we present the results for $\alpha = 0.1$ and $\tau = 1$. If $\alpha = 0.1$ the corresponding effective coefficient with $E_K = \begin{cases} 0.1, & \text{inside the obstacle} \\ 1, & \text{else} \end{cases}$ and 1024 grid cells is

$$K^* = \begin{pmatrix} 0.973177 & 1.03076 \cdot 10^{-16} \\ 1.03076 \cdot 10^{-16} & 0.973177 \end{pmatrix}.$$

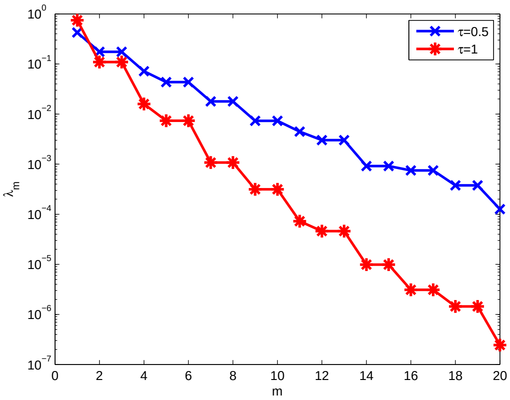
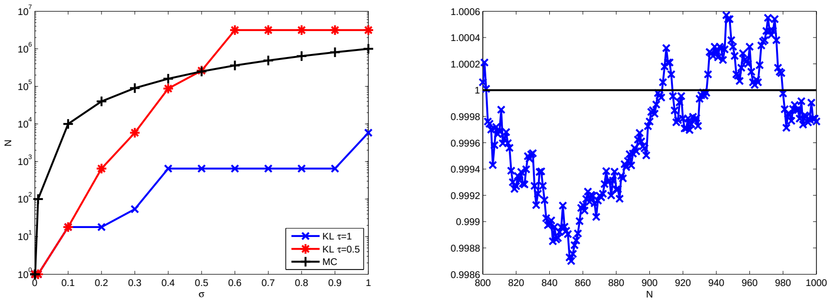


Figure 5: Decay rate of the 20 largest eigenvalues of Gaussian distributions with $\sigma = 1$.

We observe that all stopping criteria give a good approximation of the mean. As expected the accuracy decreases and the number of local problems to solve in each direction increases while we increase the standard deviation. The heuristic stopping criterion ends with the smallest number of such problems but also the error is the largest. The number of cell problems we have to solve for different standard deviations σ in the Monte Carlo (MC2) ansatz and in the Karhunen-Loève one, is illustrated in Figure 6(a). In the case of $\tau = 1$ the number of the needed cell problems for the Karhunen-Loève approach is less than for the Monte Carlo ansatz. So it may be reasonable to choose the Karhunen-Loève approach, but of course one has to take the costs of generating random numbers and of solving the large eigenproblem into account in ones decision. For $\sigma = 0.4$ the Monte Carlo ansatz is already more appropriate than the Karhunen-Loève ansatz, if $\tau = 0.5$.

In Figure 6(b) we show the behavior of the mean entry of the upscaled coefficient, if we use Monte Carlo simulation. One can see, that the value is close to the mean value before we have solved as many cell problems as the stopping criterion (MC2) demands. Therefore it makes sense to look for another stopping criterion, our suggestion is the criterion (MC1).



(a) Needed realizations for KL and MC.

(b) Behavior of the diagonal entry of the MC upscaled coefficient with $\sigma = 0.1$, $\tau = 1$.

Figure 6: Number of realizations and the diagonal entry in the MC case for $E_K(y) = 1$.

9.3.2. Lognormal distributed random variables

We consider a lognormal distributed random variable Y. As mentioned before the expected value is $\mu_{log} = \exp(\mu + \frac{\sigma^2}{2})$. We assume that the normal mean is zero, i.e., $\mu = 0$. We calculate two different upscaled coefficients. In the first case we use μ_{log} instead of the deterministic coefficient, i.e.,

$$\begin{aligned} \tilde{K}_M^{KL}(y,z) &= \mu_{log} + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) z_m \\ K_n^{MC}(y) &= Y_n, \end{aligned}$$

and in the second one

$$\begin{aligned} \tilde{K}_M^{KL}(y,z) &= \alpha + \sum_{m=1}^{M} \sqrt{\lambda_m}\phi_m(y) z_m \\ K_n^{MC}(y) &= Y_n - \mu_{log} + \alpha \end{aligned}$$

For the lognormal variance Σ^2 we use the same values as for the normal one in the previous example. To determine the corresponding normal random variables, we need the standard deviation of the underlying normal distribution of the lognormal distribution and the expected value for the above mentioned coefficients. We get

Σ	σ	μ_{log}
0.0001	0.0001	1
0.001	0.000999999	1
0.01	0.00999925	1.00005
0.1	0.0992635	1.00494.

In Table 32 and 33 we summarized the results of the first case for $\tau = 1$ and $\tau = 0.5$, respectively. In Tables 34-36 one finds the corresponding results for $\alpha = 1$ and $\alpha = 2$ for the different τs. As in the Gaussian example we observe decreasing accuracy and an increasing number of problems to solve by increasing the standard deviation. Again we have a good approximation of the mean and the Karhunen-Loève approaches perform better.

9.4. H-matrix results

In this section we show the gain in time of using a hierarchical matrix to calculate the eigenpairs of the Karhunen-Loève expansion. The underlying grid has 4096 cells. The results which we show here are for two different leafsizes. With leafsize we denote the smallest set size of indices which is allowed to be in a cluster tree, i.e., the nodes which are not further bisect have size leafsize. The cluster tree in Figure 7 has leafsize one.

The required time to assemble the matrix $\mathbf{K}'$ with leafsize 1 (left) and leafsize 4 (right) for different ϵ_{ACA}, which denotes the stopping condition for the ACA+ algorithm is illustrated in Figure 8. Assembling the full-rank matrix needs 225 seconds, that means assembling the hierarchical matrix with the weak admissibility condition is at least 5 times faster and with the standard condition we still gain a factor of 2.

In the Tables 37, 38, 39 and 40 we show the time to calculate a given number m of eigenpairs for the different admissibility conditions and leafsizes 1 and 4. For the weak admissibility condition (Table 37, 38) we can see that every choice of leafsize, ϵ_{ACA} and the number of calculated eigenpairs accelerate the computational time. In general, the increase of leafsize results in a decrease of time.

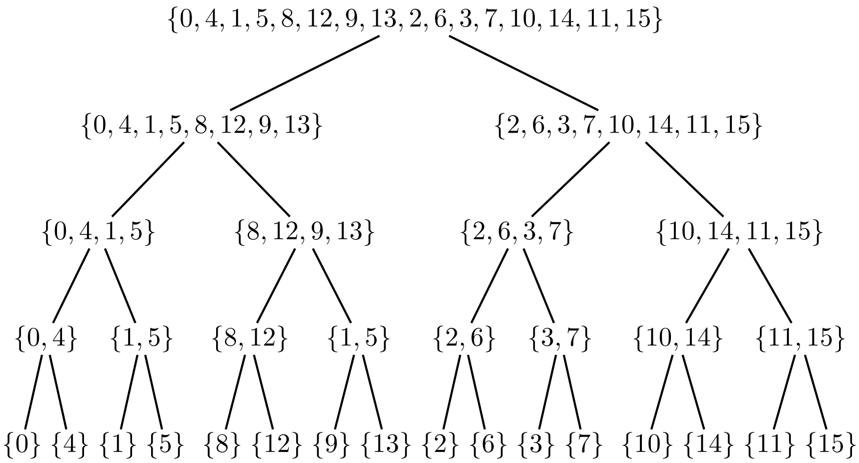
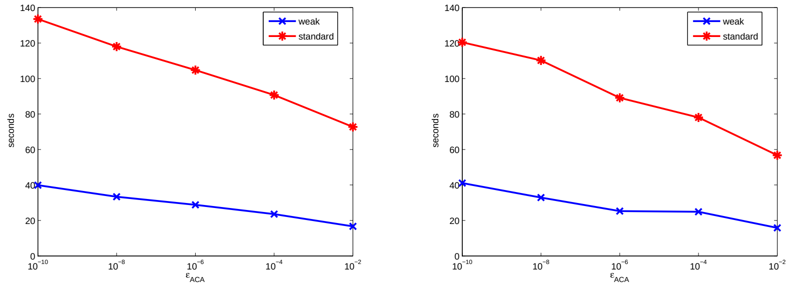


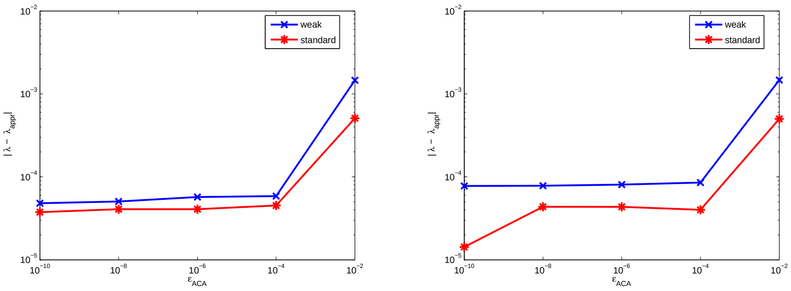
Figure 7: Cluster tree.



(a) Leafsize 1.

(b) Leafsize 4.

Figure 8: Required time to assemble the matrix.



(a) Leafsize 1.

(b) Leafsize 4.

Figure 9: Error of the eigenvalues due to the low-rank approximation.

The absolute error of the eigenvalues (c.f. Figure 9) does not depend significantly on ϵ_{ACA}, only for $\epsilon_{\text{ACA}} = 10^{-2}$ the order is different. Therefore is: leafsize= 4, $\epsilon_{\text{ACA}} = 10^{-4}$ the most reasonable one. In the case of the standard admissibility condition the error of the eigenvalues is smaller, but the time to calculate them is larger. For leafsize 1 the required time is even larger than the time of the full-rank calculation. If one decides to take the standard condition leafsize 1 is not reasonable at all, for leafsize 4 one can choose a parameter set for which the required time is smaller than the full-rank calculation, e.g., $\epsilon_{\text{ACA}} = 10^{-10}$. Since the error does not differ very much, if we chance the admissibility condition, the weak one is more reasonable because of the significant gain in time.

In Figure 10 we consider errors of the matrix due to the different admissibility conditions. As expected the error of the standard condition is smaller than the corresponding one with the weak condition.

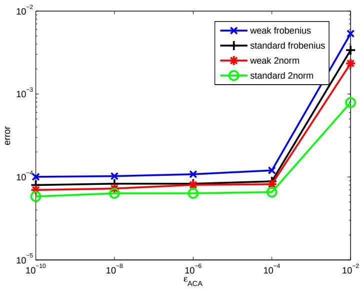
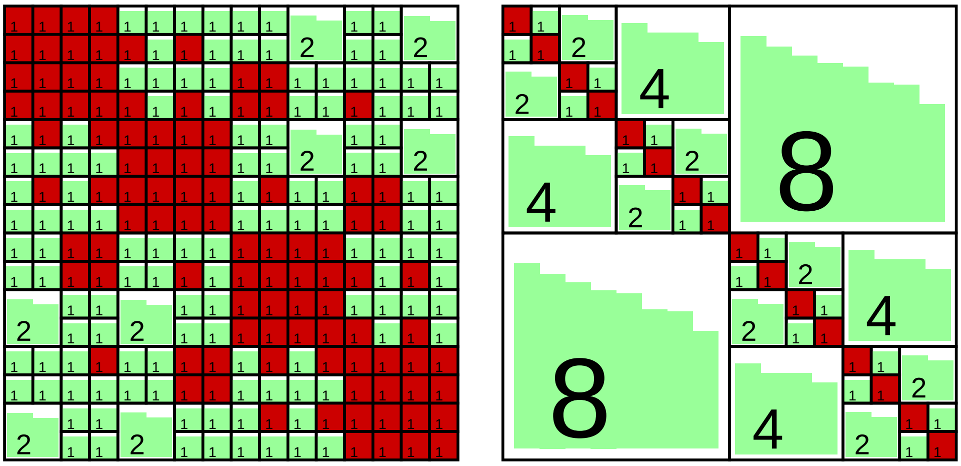


Figure 10: Matrix error due to the low rank approximation with different admissibility conditions.

The resulting block matrix structures for the two admissibility conditions are illustrated in Figure 11.



(a) Standard admissibility condition.

(b) Weak admissibility condition.

Figure 11: Block matrix structure for two different admissibility conditions with the underlying index set: $\{0, 4, 1, 5, 8, 12, 9, 13, 2, 6, 3, 7, 10, 14, 11, 15\}$ and the stopping criterion: $\epsilon_{ACA} = 10^{-4}$.

Part II.

Using multi-level Monte Carlo for numerical homogenization

In this part we consider multi-level Monte Carlo (MLMC) methods in the multiscale context ([31]). Heinrich ([38]) introduced multi-level Monte Carlo in 2001 for finite- and infinite-dimensional integration. The multi-level Monte Carlo framework is used for stochastic ODEs by Giles ([35, 34]). Recently it has been applied to PDEs with stochastic coefficients by Schwab et al.([10]) and Cliffe et al. ([22]).

The main idea of multi-level Monte Carlo is to consider the quantity of interest at different levels. At each level a different number of samples is used to compute the expected value of this quantity. In particular, the fewest samples are used at the finest level where the computation for each realization is expensive, while more samples are used at the coarsest level which is inexpensive to compute.

By selecting the number of realizations at each level carefully one can decrease the computational costs. More precisely, to compute effective properties we solve local problems on representative volumes (RVE) and average some appropriate quantity over these volumes. Because of small scales these computations can be expensive. That is why we use different representative volume sizes as levels in multi-level Monte Carlo methods. We take many samples for smaller RVEs where it is inexpensive to solve.

The convergence of multi-level Monte Carlo depends on the accuracy of the computations at each level. Here, two different accuracies come into the play. One arises by the approximation of the effective properties in each representative volume and another is from solving coarse-scale equations. The accuracy of solving the coarse-scale problems depends on the mesh size H of the coarse mesh. It is of the order H^2. The estimation of the accuracy of the effective properties is more difficult. Under some assumptions on the coefficients, it is known that the accuracy of the effective property approximation behaves ([57, 27, 16, 42, 23, 9, 36]) as $(\epsilon/\eta)^{\beta}$ for some $\beta > 0$, where η is the size of RVE and ϵ is the small scale.

We observe that multi-level Monte Carlo can be applied when the effective properties are stochastic. This is true for many applications. However, for these cases homogenizations theories are not well studied. Homogenization theories are developed mainly for ergodic coefficients. We discuss extensions of these results and apply them in our multi-level Monte Carlo. In our analysis, we discuss various choices of numbers of realizations for the local problems and the coarse-scale problems. We also discuss fine-scale meshes.

To approximate coarse-grid solutions, one has to compute effective properties at many levels. If this is carried out carefully there is an overlap between these levels, i.e., if effective properties at the finest level are computed, then these properties are computed for every coarser level. In our construction we compute the effective properties at the finest level (denoted by H_L) using largest RVE (denoted by η_L). This is the most expensive computation of the homogenized solution. At the next level H_{L-1}, some effective properties using η_L-size RVEs are already computed. We complement these effective property computations by adding new η_{L-1}-size RVEs ($\eta_L > \eta_{L-1}$) at fewer locations. As a result, we obtain coarse-grid solutions computed at the coarse grid H_i with RVE size η_j. By summing over various levels, we cannot eliminate all the terms as in standard MLMC and we propose weighted MLMC where appropriately chosen weights are used at different levels.

The following part is structured as followed. First we introduce multi-level Monte Carlo in a general setting and the notations used for the different levels. Next we consider the computation of effective properties where we use MLMC to compute the expectation or two-point correlation of homogenized coefficient. In Section 12 we discuss weighted multi-level Monte Carlo, which we apply in Section 13 to compute the expectation of the coarse-scale solution. We present numerical results for one- and two-dimensional examples. In the two-dimensional case, we only consider three different levels. In all computations, we show that one can achieve a speed-up with MLMC methods.

10. Preliminaries

10.1. Multi-level Monte Carlo method

We give a brief introduction of the multi-level Monte Carlo approach in a general setting. With G we denote a random function, $G = G(x, \omega)$. For example we will consider functions of the effective coefficient or the coarse-grid solution. We are interested in the efficient computation of the expectation of this quantity, denoted by $E[G]$.

A standard approach is the Monte Carlo method, where the expected value $E[G]$ is approximated by the arithmetic mean of a number (M) of realizations of G (denoted by G^i), i.e., $E_M := \frac{1}{M}\sum_{i=1}^{M} G^i$.

The idea of multi-level Monte Carlo (MLMC) is to consider the quantity of interest G_l on different levels l. In our case levels are various representative volume sizes or mesh sizes. We assume it is most computationally expensive to compute many realizations at the level of interest L. With $L-1$,..., 1 we introduce smaller levels, and assume that the lower the level, the cheaper the computation of G_l, and the less accurate G_l is with respect to G_L. We assume $G_0 = 0$.

We write the quantity of interest at level L as telescopic sum of the smaller levels

$$G_L = \sum_{l=1}^{L} (G_l - G_{l-1}).$$

As mentioned above we vary either the RVE size or the coarse grid resolution. For the standard MC approach we compute M realizations of the random variable G_L at the level of interest L. In contrast we work with M_l realizations of G_l at each level with $M_1 \geq M_2 \geq \cdots \geq M_L$. For the expectation we write

$$E[G_L] = \sum_{l=1}^{L} E\left[G_l - G_{l-1}\right].$$

At each level we approximate the expectation of the differences with the arithmetic mean

$$E[G_l - G_{l-1}] \approx E_{M_l}(G_l - G_{l-1}) = \frac{1}{M_l}\sum_{i=1}^{M_l}(G_l^i - G_{l-1}^i)$$

where G_l^i is the ith realization of G computed at level l (note that we have M_l realizations of G_{l-1} since $M_{l-1} \geq M_l$). In the MLMC approach the expected value $E[G_L]$ is approximated by

$$E^L(G_L) := \sum_{l=1}^{L} E_{M_l}\left(G_l - G_{l-1}\right). \tag{10.1}$$

The realizations of G_{l-1} used with G_l to evaluate $E_{M_l}(G_l - G_{l-1})$ do not have to be independent of the realizations G_{l-1} used for $E_{M_{l-1}}(G_{l-1} - G_{l-2})$ (cf. Section 10.2). In our analysis we consider root mean square errors

$$\begin{aligned} e_{MLMC}(G_L) &= \sqrt{E\left[\vert\vert\vert E[G_L] - E^L(G_L)\vert\vert\vert^2\right]} && (10.2)\\ e_{MC}(G_L) &= \sqrt{E\left[\vert\vert\vert E[G_L] - E_{\hat{M}}(G_L)\vert\vert\vert^2\right]} && (10.3) \end{aligned}$$

with an appropriate norm $\vert\vert\vert \cdot \vert\vert\vert$ depending on the quantity of interest, e.g., the absolute value for any entry of the homogenized coefficient. For the error estimation we will use (see [17])

$$E\left[\vert\vert\vert E[G] - E_M(G)\vert\vert\vert^2\right] \leq \frac{1}{M} E\left[\vert\vert\vert G - E[G]\vert\vert\vert\right]^2 \tag{10.4}$$

which is valid for any random variable G and norm associated with a scalar product.

10.2. Remark on same or independent samples

For multi-level Monte Carlo we use M_1 samples of G_1 and M_l of $G_l - G_{l-1}$ with $M_l < M_{l-1}$, $2 \leq l \leq L$. The question arises if the M_l realizations of G_{l-1} in the terms $G_l - G_{l-1}$ and $G_{l-1} - G_{l-2}$ have to be independent. In the following we answer the question for only two different levels if the considered norm is the absolute value. Later (cf. paragraph "independent samples" in Section 14.1.2) we underline this result for more than two levels numerically. In the following we assume, we compute M_1 realizations of the quantity G_1 at level 1 and M_2 of G_2. In this way we ensure to have the same computational cost in both approaches. If we use the same realizations of G_1 in both levels we can write the MLMC approximation as

$$\begin{aligned} E^L_{same}(G_L) &:= \frac{1}{M_2}\sum_{i=1}^{M_2}\left(G_2^i - G_1^i\right) + \frac{1}{M_1}\sum_{i=1}^{M_1} G_1^i && (10.5)\\ &= \sum_{i=1}^{M_2}\left(\left(\frac{1}{M_1} - \frac{1}{M_2}\right)G_1^i + \frac{1}{M_2}G_2^i\right) + \sum_{i=M_2+1}^{M_1}\frac{1}{M_1}G_1^i \end{aligned}$$

and in the independent case

$$E^L_{ind}(G_L) := \frac{1}{M_2}\sum_{i=1}^{M_2}\left(G_2^i - G_1^i\right) + \frac{1}{\widetilde{M}_1}\sum_{i=M_2+1}^{M_1} G_1^i \tag{10.6}$$

with $\widetilde{M}_1 = M_1 - M_2$ and $\widetilde{M}_1 > \widetilde{M}_2$ (therefor $M_1 > M_2$). Note that for the MLMC approximation in the case of independent samples we use $\widetilde{M}_l = M_l - M_{l+1}$ instead of M_l to approximate $E[G_l - G_{l-1}]$. Such that this approximation of the expectation is less accurate, but the independence of the sample at the different levels might increase the accuracy of the whole approximation of $E[G_L]$. If we add and subtract $E[G_1]$, we get

$$\begin{aligned} E^L_{same}(G_L) &= \sum_{i=1}^{M_2}\left(\left(\frac{1}{M_1} - \frac{1}{M_2}\right)\overline{G}_1^i + \frac{1}{M_2}G_2^i\right) + \sum_{i=M_2+1}^{M_1}\frac{1}{M_1}\overline{G}_1^i,\\ E^L_{ind}(G_L) &= \frac{1}{M_2}\sum_{i=1}^{M_2}\left(G_2^i - \overline{G}_1^i\right) + \frac{1}{\widetilde{M}_1}\sum_{i=M_2+1}^{M_1}\overline{G}_1^i \end{aligned}$$

with $\overline{G}_l^i = G_l^i - E[G_l]$. As mentioned above we are interested in the root mean square error. If we use the same samples we get

$$\begin{aligned}
& \left(e_{MLMC}^{same}(G_2)\right)^2 \\
= \; & E\left[\left(E_{same}^L(G_2) - E[G_2]\right)^2\right] \\
= \; & E\left[\left(\sum_{i=1}^{M_2}\left(\left(\frac{1}{M_1}-\frac{1}{M_2}\right)\overline{G}_1^i + \frac{1}{M_2}G_2^i\right) + \sum_{i=M_2+1}^{M_1}\frac{1}{M_1}\overline{G}_1^i - E[G_2]\right)^2\right] \\
= \; & E\left[\left(\sum_{i=1}^{M_2}\left(\left(\frac{1}{M_1}-\frac{1}{M_2}\right)\overline{G}_1^i + \frac{1}{M_2}\overline{G}_2^i\right) + \sum_{i=M_2+1}^{M_1}\frac{1}{M_1}\overline{G}_1^i\right)^2\right] \\
= \; & \sum_{i=1}^{M_2} E\left[\left(\left(\frac{1}{M_1}-\frac{1}{M_2}\right)\overline{G}_1^i + \frac{1}{M_2}G_2^i\right)^2\right] + \frac{1}{M_1^2}\sum_{i=M_2+1}^{M_1} E\left[\left(\overline{G}_1^i\right)^2\right] \\
= \; & M_2\left[\frac{1}{M_2^2}\mathrm{Var}(G_2) + \left(\frac{1}{M_1}-\frac{1}{M_2}\right)^2\mathrm{Var}(G_1) + \frac{2}{M_2}\left(\frac{1}{M_1}-\frac{1}{M_2}\right)\mathrm{Cov}(G_1,G_2)\right] + \frac{M_1-M_2}{M_1^2}\mathrm{Var}(G_1) \\
= \; & \frac{1}{M_2}\mathrm{Var}(G_2) + M_2\left(\frac{1}{M_1}-\frac{1}{M_2}\right)^2\mathrm{Var}(G_1) + 2\left(\frac{1}{M_1}-\frac{1}{M_2}\right)\mathrm{Cov}(G_1,G_2) \\
= \; & \frac{1}{M_2}\mathrm{Var}(G_2) + \frac{M_1-M_2}{M_1M_2}\mathrm{Var}(G_1) + 2\frac{M_2-M_1}{M_1M_2}\mathrm{Cov}(G_1,G_2)
\end{aligned}$$

and for the independent case

$$\begin{aligned}
& \left(e_{MLMC}^{ind}(G_2)\right)^2 \\
= \; & E\left[\left(E_{ind}^L(G_2) - E[G_2]\right)^2\right] \\
= \; & E\left[\left(\frac{1}{M_2}\sum_{i=1}^{M_2}\left(G_2^i - \overline{G}_1^i\right) + \frac{1}{\widetilde{M}_1}\sum_{i=M_2+1}^{M_1}\overline{G}_1^i - E[G_2]\right)^2\right] \\
= \; & E\left[\left(\frac{1}{M_2}\sum_{i=1}^{M_2}\left(\overline{G}_2^i - \overline{G}_1^i\right) + \frac{1}{\widetilde{M}_1}\sum_{i=M_2+1}^{M_1}\overline{G}_1^i\right)^2\right] \\
= \; & \frac{1}{M_2^2}\sum_{i=1}^{M_2} E\left[\left(\overline{G}_2^i - \overline{G}_1^i\right)^2\right] + \frac{1}{\widetilde{M}_1^2}\sum_{i=M_2+1}^{M_1} E\left[\left(\overline{G}_1^i\right)^2\right] \\
= \; & \frac{1}{M_2^2}\sum_{i=1}^{M_2}\left(E\left[\left(\overline{G}_2^i\right)^2\right] - 2E\left[\overline{G}_2^i\overline{G}_1^i\right] + E\left[\left(\overline{G}_1^i\right)^2\right]\right) + \frac{1}{\widetilde{M}_1}\mathrm{Var}(G_1) \\
= \; & \frac{1}{M_2}\left[\mathrm{Var}(G_1) + \mathrm{Var}(G_2) - 2\mathrm{Cov}(G_1,G_2)\right] + \frac{1}{\widetilde{M}_1}\mathrm{Var}(G_1) \\
= \; & \left(\frac{1}{\widetilde{M}_1} - \frac{1}{M_2}\right)\mathrm{Var}(G_1) + \frac{1}{M_2}\mathrm{Var}(G_2) - \frac{2}{M_2}\mathrm{Cov}(G_1,G_2) \\
= \; & \frac{M_1}{\widetilde{M}_1M_2}\mathrm{Var}(G_1) + \frac{1}{M_2}\mathrm{Var}(G_2) - \frac{2}{M_2}\mathrm{Cov}(G_1,G_2) \\
= \; & \frac{M_1}{(M_1-M_2)M_2}\mathrm{Var}(G_1) + \frac{1}{M_2}\mathrm{Var}(G_2) - \frac{2}{M_2}\mathrm{Cov}(G_1,G_2).
\end{aligned}$$

In both cases we used the independence of G_l^i and G_l^j for $i \neq j$.

Remark 10.1
If we consider the L^2-norm instead of the absolute value, we do not have $Var(G_l)$ but $\| Var(G_l)\|_{L^1}$.

To check which error is smaller we look at the difference.

$$
\begin{aligned}
& \left(e^{ind}_{MLMC}(G_2)\right)^2 - (e^{same}_{MLMC}(G_2))^2 \\
= \; & \frac{M_1}{(M_1-M_2)M_2}\mathrm{Var}(G_1) + \frac{1}{M_2}\mathrm{Var}(G_2) - \frac{2}{M_2}\mathrm{Cov}(G_1,G_2) \\
& - \left(\frac{1}{M_2}\mathrm{Var}(G_2) + \frac{M_1-M_2}{M_1M_2}\mathrm{Var}(G_1) + 2\frac{M_2-M_1}{M_1M_2}\mathrm{Cov}(G_1,G_2)\right) \\
= \; & \left[\frac{M_1}{(M_1-M_2)M_2} - \frac{M_1-M_2}{M_1M_2}\right]\mathrm{Var}(G_1) - 2\left[\frac{1}{M_2} - \frac{M_2-M_1}{M_1M_2}\right]\mathrm{Cov}(G_1,G_2) \\
= \; & \frac{2M_1-M_2}{M_1(M_1-M_2)}\mathrm{Var}(G_1) - \frac{2}{M_2}\mathrm{Cov}(G_1,G_2) \\
\geq \; & \frac{2M_1-M_2}{M_1(M_1-M_2)}\mathrm{Var}(G_1) - \frac{2}{M_2}\sqrt{\mathrm{Var}(G_1)\mathrm{Var}(G_2)} \\
\geq \; & \frac{2M_1-M_2}{M_1(M_1-M_2)}\mathrm{Var}(G_1). \\
\geq \; & 0.
\end{aligned}
$$

Here we use $\mathrm{Var}(G_1) \geq \mathrm{Var}(G_2)$. The above analysis shows for two different levels that we achieve a better accuracy with the same amount of computational costs, if we reuse the samples. This coincides with our numerical results in the paragraph 'independent samples' in Section 14.1.2.

10.3. Definition of meshes and representative volume sizes

In our application we vary representative volume sizes, the sizes of coarse meshes, and the fine-scale discretization of these representative volume sizes (see Figure 12 for illustration). In the MLMC framework choosing a level l means choosing a particular RVE or mesh size. We denote the hierarchy of the representative volumes by

$$\eta_1 < \eta_2 < ... < \eta_L$$

We assume that the number of realizations used at the level l for the RVE size η_l is m_l. We take

$$m_1 > m_2 > ... > m_L.$$

As different coarse meshes sizes we take

$$H_1 > H_2 > ... > H_L$$

and the corresponding number of realizations

$$M_1 > M_2 > ... > M_L.$$

One can also vary the fine-scale meshes for solving local representative volume problems. We denote these mesh sizes by

$$h_1 > h_2 > ... > h_L.$$

The level L always corresponds to the most expensive and accurate choice (e.g., largest RVE size, finest mesh), and thus to the smallest number of realizations. We note that one does not have to take the same number of levels L for coarse-grid sizes and RVEs.

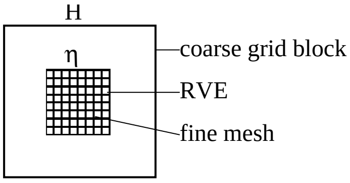


Figure 12: Illustration of a coarse grid block with RVE.

11. Multi-level Monte Carlo for the upscaled coefficients and their properties

In this section we apply MLMC to compute the expectation of the effective coefficient. Therefore, we solve local problems on RVEs with various sizes. As a first step, we will fix the fine-scale grid and later we consider both, various RVE sizes and fine meshes. If the homogenized coefficients are stochastic the multi-level Monte Carlo method is more efficient than a standard Monte Carlo approach. Otherwise, both methods are equally accurate. As mentioned earlier the homogenized coefficient can be stochastic, if no ergodicity is assumed. In this setting we will consider various cases. The first case we will consider is when the heterogeneous field in (3.1) has the product form

$$K\left(x, \frac{x}{\epsilon}, \omega, \omega'\right) = A(x,\omega)B\left(\frac{x}{\epsilon}, \omega'\right)$$

where A and B are two random scalar valued functions. ω corresponds to the randomness of the macroscopic scale and ω' to the one of the microscopic scale. With $K^*(x,\omega,\omega')$ we denote the homogenized coefficient matrix which depends on the macroscopic variables (x,ω) and on ω' if no ergodicity on B is assumed. Then we assume

$$E_{\omega'}\left[\left|\!\left|\!\left| K^*(x,\omega,\omega') - \mathcal{H}_\eta\left(K\left(x, \frac{x}{\epsilon}, \omega, \omega'\right)\right)\right|\!\right|\!\right|^2\right] \le C\left(\frac{\epsilon}{\eta}\right)^\beta$$

for some deterministic constant C independent of ω, x, ϵ and η and any matrix norm $|\!|\!|\cdot|\!|\!|$. Furthermore, the rate β is assumed to be independent of ω, x, ϵ and η. Another, more general case we will consider is when one cannot split the randomness of macroscopic and microscopic scale, explicitely. Then the coefficient writes $K\left(x, \frac{x}{\epsilon}, \omega\right)$. We assume that K is scalar-valued, that we can do homogenization in every macroscopic point, and that the following assumption holds

$$E\left[\left|\!\left|\!\left| K^*(x,\omega) - \mathcal{H}_\eta\left(K\left(x, \frac{x}{\epsilon}, \omega\right)\right)\right|\!\right|\!\right|^2\right] \le C\left(\frac{\epsilon}{\eta}\right)^\beta$$

with some constant C and rate β independent of x, ϵ and η. This is similar to the known results for ergodic homogeneous coefficient recalled in Section 14.1.1.

11.1. Various RVE sizes and fixed fine mesh

We assume a highly resolved discretization of the cell problem, i.e., no fine-scale discretization error. Later on we introduce how to compute the two-point correlation function. At each level we set

$$\sqrt{E\left[|\!|\!| K^* - K_l^* |\!|\!|^2\right]} = \delta_l$$

for some matrix norm $|||\cdot|||$ and to simplify the notation we set $K_l^* := K_{\eta_l}^*$. We assume that

$$\delta_l \leq C \left(\frac{\epsilon}{\eta_l}\right)^{\frac{\beta}{2}}, \tag{11.1}$$

with $\beta > 0$ and $C > 0$ indendent of l, ϵ and η. For some special cases, one can estimate β rigorously, but in general we suggest a precomputation strategy to estimate β. This will be discussed later. Note that Central Limit Type results correspond to $\beta = d$ (see e.g., [13] for such estimates in a weakly stochastic case). For clarity we summarize the basic steps of MLMC for the upscaled coefficients below.

1. Generate m_1 random variables $\omega_1 \cdots \omega_{m_1}$.

2. For each level l, $1 \leq l \leq L$, and each realization ω_j, $1 \leq j \leq m_l$,

    - Solve the RVE problems

    $$\mathrm{div}(K_\epsilon(x,\omega_j)\nabla\chi_i^j) = 0, \text{ in } Y_{\eta_l}, \quad \chi_i^j(x,\omega_j) = x_i \text{ on } \partial Y_{\eta_l},$$

    for any $i = 1, ..., d$.

    - Compute the homogenized coefficients $K_l^*(x,\omega_j)$ with

    $$\forall 1 \leq i \leq d, \quad K_l^*(x,\omega_j)e_i = \frac{1}{{\eta_l}^d}\int_{Y_{\eta_l}} K_\epsilon(x,\omega_j)\nabla\chi_i^j.$$

3. For each level l, $1 \leq l \leq L$, compute

    $$E_{m_l}(K_l^* - K_{l-1}^*) = \frac{1}{m_l}\sum_{j=1}^{m_l} K_l^*(x,\omega_j) - K_{l-1}^*(x,\omega_j),$$

    where we set $K_0^* = 0$. Note that we keep implicit the dependence of $E_{m_l}(K_l^* - K_{l-1}^*)$ with respect to x and the randomness.

4. Compute the multi-level approximation $E^L(K_L^*)$ of the expected value $E[K_L^*]$ following (10.1):

    $$E^L(K_L^*) = \sum_{l=1}^{L} E_{m_l}(K_l^* - K_{l-1}^*)(x).$$

We start to estimate the root mean square error of the approximation of $E([K_L^*]_{ij})$, for any entry ij ($1 \leq i,j \leq d$) of the matrix K_L^*. To simplify the notation, we write the calculations below as if K_l^* is a scalar quantity, i.e., the calculations are to be understood as calculations on the entry $[K_l^*]_{ij}$. This estimate can be extended to any smooth, scalar-valued function f of K_L^*, this will be discussed in Remark 11.2 below. For the multi-level Monte Carlo approach, we get

$$\begin{aligned}
e_{MLMC}(K_L^*) &= \sqrt{E\left[(E[K_L^*] - E^L(K_L^*))^2\right]} \\
&= \sqrt{E\left[\left(E\left[\sum_{l=1}^{L}\left(K_l^* - K_{l-1}^*\right)\right] - \sum_{l=1}^{L} E_{m_l}\left(K_l^* - K_{l-1}^*\right)\right)^2\right]} \\
&= \sqrt{E\left[\left(\sum_{l=1}^{L}\left(E - E_{m_l}\right)\left(K_l^* - K_{l-1}^*\right)\right)^2\right]} \\
&\le \sum_{l=1}^{L}\sqrt{E\left[\left(\left(E - E_{m_l}\right)\left(K_l^* - K_{l-1}^*\right)\right)^2\right]} \\
&\le \sum_{l=1}^{L}\frac{1}{\sqrt{m_l}}\sqrt{E\left[(K_l^* - K_{l-1}^* - E[K_l^* - K_{l-1}^*])^2\right]}
\end{aligned}$$

where we have used (10.4). Writing that $K_l^* - K_{l-1}^* = (K_l^* - K^*) + (K^* - K_{l-1}^*)$ we deduce

$$\begin{aligned}
e_{MLMC}(K_L^*) &= \sqrt{E\left[(E[K_L^*] - E^L(K_L^*))^2\right]} \\
&\le \sum_{l=1}^{L}\frac{1}{\sqrt{m_l}}\sqrt{E\left[(K_l^* - K^* - E[K_l^* - K^*])^2\right]} \\
&+ \sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}\sqrt{E\left[(K^* - K_{l-1}^* - E[K^* - K_{l-1}^*])^2\right]} \\
&+ \frac{1}{\sqrt{m_1}}\sqrt{E\left[(K^* - E[K^*])^2\right]} \\
&\le \sum_{l=1}^{L}\frac{1}{\sqrt{m_l}}\sqrt{E\left[(K_l^* - K^*)^2\right]} \\
&+ \sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}\sqrt{E\left[(K^* - K_{l-1}^*)^2\right]} \\
&+ \frac{1}{\sqrt{m_1}}\sqrt{E\left[(K^* - E[K^*])^2\right]}. \\
&= \sum_{l=1}^{L}\frac{1}{\sqrt{m_l}}\delta_l + \sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}\delta_{l-1} + \frac{1}{\sqrt{m_1}}\sqrt{E\left[(K^* - E[K^*])^2\right]}. \\
&= \sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}(\delta_l + \delta_{l-1}) + \frac{1}{\sqrt{m_1}}\left(\delta_1 + \sqrt{E\left[(K^* - E[K^*])^2\right]}\right).
\end{aligned}$$

Using (11.1), we get

$$\begin{aligned}
e_{MLMC}(K_L^*) \le C\sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}&\left(\left(\frac{\epsilon}{\eta_l}\right)^{\beta/2} + \left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta/2}\right) \\
&+ \frac{1}{\sqrt{m_1}}\left(C\left(\frac{\epsilon}{\eta_1}\right)^{\beta/2} + \sqrt{E\left[(K^* - E[K^*])^2\right]}\right).
\end{aligned}$$

If the error is fixed, the optimal choice for the number m_l of realizations at level l (i.e., with RVE size η_l) is reached when these error parts are equilibrated. We choose

$$m_l = \begin{cases} \left(\frac{\eta_L}{\epsilon}\right)^{\beta} \left(\left(\frac{\epsilon}{\eta_1}\right)^{\frac{\beta}{2}} + \frac{\sqrt{E[(K^*-E[K^*])^2]}}{C} \right)^2 \alpha_1^{-2}, & l = 1 \\ \left(\frac{\eta_L}{\eta_l}\right)^{\beta} \left(1 + \left(\frac{\eta_l}{\eta_{l-1}}\right)^{\frac{\beta}{2}}\right)^2 \alpha_l^{-2}, & l \geq 2 \end{cases} \tag{11.2}$$

with $\alpha_l > 0$, $1 \leq l \leq L$. Then we have

$$\begin{aligned} e_{MLMC}(K_L^*) &\leq C \sum_{l=2}^{L} \frac{1}{\sqrt{m_l}} \left(\left(\frac{\epsilon}{\eta_l}\right)^{\beta/2} + \left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta/2} \right) \\ &\quad + \frac{1}{\sqrt{m_1}} \left(C \left(\frac{\epsilon}{\eta_1}\right)^{\beta/2} + \sqrt{E\left[(K^* - E[K^*])^2\right]} \right) \\ &= C \sum_{l=2}^{L} \alpha_l \frac{\left(\frac{\epsilon}{\eta_l}\right)^{\beta/2} + \left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta/2}}{\left(\frac{\eta_L}{\eta_l}\right)^{\frac{\beta}{2}} \left(1 + \left(\frac{\eta_l}{\eta_{l-1}}\right)^{\frac{\beta}{2}}\right)} \\ &\quad + \alpha_1 \frac{C \left(\frac{\epsilon}{\eta_1}\right)^{\beta/2} + \sqrt{E\left[(K^* - E[K^*])^2\right]}}{\left(\frac{\eta_L}{\epsilon}\right)^{\frac{\beta}{2}} \left(\left(\frac{\epsilon}{\eta_1}\right)^{\frac{\beta}{2}} + \frac{\sqrt{E[(K^*-E[K^*])^2]}}{C} \right)} \\ &= C \sum_{l=2}^{L} \alpha_l \left(\frac{\epsilon}{\eta_L}\right)^{\frac{\beta}{2}} \frac{\left(\frac{1}{\eta_l}\right)^{\beta/2} + \left(\frac{1}{\eta_{l-1}}\right)^{\beta/2}}{\left(\frac{1}{\eta_l}\right)^{\frac{\beta}{2}} \left(1 + \left(\frac{\eta_l}{\eta_{l-1}}\right)^{\frac{\beta}{2}}\right)} + C\alpha_1 \left(\frac{\epsilon}{\eta_L}\right)^{\frac{\beta}{2}} \\ &= C \left(\frac{\epsilon}{\eta_L}\right)^{\frac{\beta}{2}} \sum_{l=1}^{L} \alpha_l \\ &= \left(\frac{\epsilon}{\eta_L}\right)^{\frac{\beta}{2}} C(\alpha), \end{aligned}$$

where $C(\alpha) = C \sum_{l=1}^{L} \alpha_l$.

For comparison we consider the error of standard Monte Carlo, if we calculate the approximated upscaled coefficient only for the largest RVE (of size η_L). Using (10.4), we have

$$\sqrt{E\left((E[K_L^*] - E_{\widehat{m}}(K_L^*))^2\right]} \leq \frac{1}{\sqrt{\widehat{m}}} \sqrt{E\left[(K_L^* - E[K_L^*])^2\right]}.$$

As mentioned above, we assume K^* to be a random quantity, with some positive variance. Therefor it is natural to assume that the variance is roughly independent of L. Thus the Monte Carlo error is of the order $\tilde{C}/\sqrt{\widehat{m}}$. To have a Monte Carlo error of the same order as the MLMC error, we take $\widehat{m} = O\left(\left(\frac{\eta_L}{\epsilon}\right)^{\beta}\right)$ samples.

If we choose these numbers of realizations for MLMC and MC, both methods reach the same accuracy and we can compare their costs. Let N_l denote the cost to solve the RVE problem (4.9) on the domain $Y_{\eta_l}^x$ of size η_l. The number of degrees of freedom is of the order $(\eta_l/\epsilon)^d$. Assuming

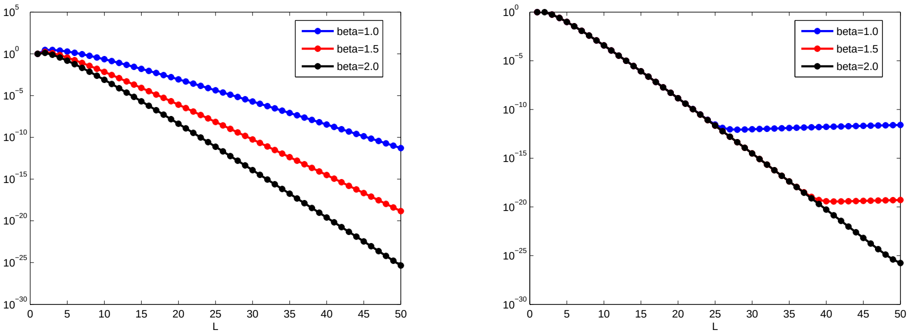
$N_l = (\eta_l/\epsilon)^d$, we have the following cost for MLMC

$$
\begin{aligned}
W^{MLMC}_{\text{RVE}} &= \sum_{l=1}^{L} m_l N_l \\
&\simeq \sum_{l=2}^{L} \left(\frac{\eta_L}{\eta_l}\right)^{\beta} \left(1 + \left(\frac{\eta_l}{\eta_{l-1}}\right)^{\frac{\beta}{2}}\right)^2 \alpha_l^{-2} \left(\frac{\eta_l}{\epsilon}\right)^d \\
&\quad + \left(\frac{\eta_L}{\epsilon}\right)^{\beta} \left(\left(\frac{\epsilon}{\eta_1}\right)^{\frac{\beta}{2}} + \frac{\sqrt{E\left[(K^* - E[K^*])^2\right]}}{C}\right)^2 \alpha_1^{-2} \left(\frac{\eta_1}{\epsilon}\right)^d .
\end{aligned}
$$

In the case of MC, the cost reads

$$
\begin{aligned}
W^{MC}_{\text{RVE}} &= \widehat{m} N_L \simeq \left(\frac{\eta_L}{\epsilon}\right)^{\beta} \left(\frac{\eta_L}{\epsilon}\right)^d \\
&= \left(\frac{\eta_L}{\epsilon}\right)^{\beta+d} .
\end{aligned}
$$

In Figure 13 we illustrate the ratio $\frac{W^{MLMC}_{\text{RVE}}}{W^{MC}_{\text{RVE}}}$ for different numbers of levels L and rates β. As we can see, for a given number L of levels, the cost ratio decreases if the rate β increases, at equal accuracy. Otherwise stated, the faster the convergence of the homogenized matrix with respect to the RVE size (cf. (4.16)), the more interesting the MLMC approach is.



(a) $\epsilon = \frac{\eta_1}{10}$

(b) $\epsilon = \frac{2^{-50}}{10}$

Figure 13: RVE work ratio: $\frac{W^{MLMC}}{W^{MC}}$ for different levels L and βs, $d = 2$ $\eta_l = 2^{-L+l}$ and $\alpha_l = \frac{1}{L}$, $m_1 = \left(\frac{\eta_L}{\epsilon}\right)^{\beta}\left(\left(\frac{\epsilon}{\eta_1}\right)^{\frac{\beta}{2}} + 1\right)^2 \alpha_1^{-2}$, $m_l = \left(\frac{\eta_L}{\eta_l}\right)^{\beta}\left(\left(\frac{\eta_l}{\eta_{l-1}}\right)^{\frac{\beta}{2}} + 1\right)^2 \alpha_l^{-2}$, $l \geq 2$.

Remark 11.1

In the above calculations, we assumed that the work of solving a local problem scales as N_l where N_l is the number of degrees of freedoms. This is true if one uses iterative solvers and the conditioned number of the preconditioned system is independent of the small scale ϵ. One can also compare the work between MLMC and MC approaches when the work of solving a local problems scales as $C(\epsilon)N^{1+\gamma}$ for some $\gamma > 0$.

Remark 11.2
Note that estimating the difference between the expectation $E[K_L^*]$ and the MLMC approximation $E^L(K_L^*)$ can be replaced by estimating the difference of $E[f(K_L^*)]$ and $E^L(f(K_L^*))$ for any smooth, scalar-valued function $f(K_L^*)$, so that the estimate of the difference between two consecutive levels can be related to the estimate of K_l^* between two consecutive levels. Then we can write $\left|f(K_l^*) - f(K_{l-1}^*)\right| \leq C_f \sum_{i,j=1}^d \left|[K_l^*]_{ij} - [K_{l-1}^*]_{ij}\right|$ for some constant C_f and proceed as above.

Remark 11.3
To compute the optimal number of realizations m_l the convergence rate β is needed, which is not known, in general. In Section 14.1.1 we propose some means to estimate β numerically.

Above we have shown how to estimate $E[K_L^*]$. Another important function is the two-point correlation function

$$Cor_{K^*}(x,y) = E\left([K^*(x,\omega)]_{ij}[K^*(y,\omega)]_{qp}\right)$$

between the components ij and qp of the homogenized matrix at points x and y (note that we work with non-centered values of K^*). To keep it simple we consider only two fixed locations x and y. Consider m_l independent samples of the homogenized coefficient $K_l^{*,k}$ at level l, $(1 \leq k \leq m_l)$. We define

$$Cor_{m_l}(K_l^*) := \frac{1}{m_l}\sum_{k=1}^{m_l}[K_l^{*,k}(x)]_{ij}\,[\,K_l^{*,k}(y)]_{qp}$$

as an empirical estimator of $E\left([K^*(x,\omega)]_{ij}[K^*(y,\omega)]_{qp}\right)$. As MLMC approximation for the two-point correlation function $Cor_{K^*}(x,y)$ we get

$$Cor^L(K_L^*) := \sum_{l=1}^{L}\left(Cor_{m_l}(K_l^*) - Cor_{m_l}(K_{l-1}^*)\right).$$

11.2. Coarse and fine meshes

Above we have assumed that we could solve the RVE problems (4.9) exactly, but in practice, these problems are solved numerically within some accuracy. In this section we consider the extension, where we vary the RVE size η_i and the fine mesh h_j. We assume that the error in the approximation of K^* satisfies

$$\delta_{ij} \leq C\sqrt{\left(\frac{\epsilon}{\eta_i}\right)^{\beta} + \left(\frac{h_j}{\epsilon}\right)^{\gamma}} \tag{11.3}$$

for some constant C independent of η_i, h_j and ϵ. We take m_{ij} samples at the level (i,j). It is possible to consider all pairs (i,j); however, the cost of the computations can be large. For each RVE size η_l we will choose a corresponding fine-grid size h_l. We denote this as level l as before K_{l,h_l}^*. We get as above

$$\begin{aligned}e_{MLMC}(K_{L,h_L}^*) &= \sqrt{E\left[(E[K_{L,h_L}^*] - E^L(K_{L,h_L}^*))^2\right]}\\ &\leq \sum_{l=2}^{L}\frac{1}{\sqrt{m_l}}(\delta_{ll} + \delta_{l-1l-1}) + \frac{1}{\sqrt{m_1}}\left(\delta_{11} + \sqrt{E\left[(K^* - E[K^*])^2\right]}\right).\end{aligned}$$

Using (11.3), we get

$$
\begin{aligned}
e_{MLMC}(K_L^*) \le C \sum_{l=2}^{L} \frac{1}{\sqrt{m_l}} & \left(\sqrt{\left(\frac{\epsilon}{\eta_l}\right)^{\beta} + \left(\frac{h_l}{\epsilon}\right)^{\gamma}} + \sqrt{\left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta} + \left(\frac{h_{l-1}}{\epsilon}\right)^{\gamma}} \right) \\
& + \frac{1}{\sqrt{m_1}} \left(C \sqrt{\left(\frac{\epsilon}{\eta_1}\right)^{\beta} + \left(\frac{h_1}{\epsilon}\right)^{\gamma}} + \sqrt{E\left[(K^* - E[K^*])^2\right]} \right).
\end{aligned}
$$

If the error is fixed, the optimal choice for the number m_l of realizations at level l (i.e., with RVE size η_l and mesh size h_l) is reached when these error parts are equilibrated. We choose

$$
m_l = \begin{cases}
\dfrac{\left(\sqrt{\left(\frac{\epsilon}{\eta_1}\right)^{\beta} + \left(\frac{h_1}{\epsilon}\right)^{\gamma}} + \frac{1}{C}\sqrt{E[(K^* - E[K^*])^2]}\right)^2}{\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma}} \alpha_1^{-2}, & l = 1 \\
\dfrac{\left(\sqrt{\left(\frac{\epsilon}{\eta_l}\right)^{\beta} + \left(\frac{h_l}{\epsilon}\right)^{\gamma}} + \sqrt{\left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta} + \left(\frac{h_{l-1}}{\epsilon}\right)^{\gamma}}\right)^2}{\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma}} \alpha_l^{-2}, & l \ge 2.
\end{cases}
$$

Then we have

$$
e_{MLMC}(K_L^*) \le \left[\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma} \right] C(\alpha)
$$

where $C(\alpha) = C \sum_{l=1}^{L} \alpha_l$. For comparison we calculate the corresponding error of standard MC with RVE size η_L and fine grid h_L

$$
\sqrt{E\left[(E[K^*_{L,h_L}] - E_{\widehat{m}}(K^*_{L,h_L}))^2\right]} \lesssim \frac{1}{\sqrt{\widehat{m}}} \sqrt{E\left[(K^*_{L,h_L} - E[K^*_{L,h_L}])^2\right]}.
$$

Again we have a Monte Carlo error of the order of $\tilde{C}/\sqrt{\widehat{m}}$. To equilibrate the error terms in this case we take $\widehat{m} = O\left(\left(\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma} \right)^{-1} \right)$ samples.

If we choose these numbers of realizations for MLMC and MC, both methods reach the same accuracy and we can compare their costs. Let N_l denote the cost to solve the RVE problem (4.9) on the domain $Y^x_{\eta_l}$ of size η_l with mesh size h_l. The number of degrees of freedom is of the order $(\eta_l/h_l)^d$. Assuming $N_l = (\eta_l/h_l)^d$, we have the following cost for MLMC

$$
\begin{aligned}
W^{MLMC}_{\text{RVE}} &= \sum_{l=1}^{L} m_l N_l \\
&\simeq \sum_{l=2}^{L} \frac{\left(\sqrt{\left(\frac{\epsilon}{\eta_l}\right)^{\beta} + \left(\frac{h_l}{\epsilon}\right)^{\gamma}} + \sqrt{\left(\frac{\epsilon}{\eta_{l-1}}\right)^{\beta} + \left(\frac{h_{l-1}}{\epsilon}\right)^{\gamma}}\right)^2}{\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma}} \alpha_l^{-2} \left(\frac{\eta_l}{h_l}\right)^2 \\
&\quad + \frac{\left(\sqrt{\left(\frac{\epsilon}{\eta_1}\right)^{\beta} + \left(\frac{h_1}{\epsilon}\right)^{\gamma}} + \frac{1}{C}\sqrt{E\left[(K^* - E[K^*])^2\right]}\right)^2}{\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma}} \alpha_1^{-2} \left(\frac{\eta_1}{h_1}\right)^2.
\end{aligned}
$$

For MC, the amount of work is

$$W_{\text{RVE}}^{MC} \quad = \quad \widehat{m}N_L \simeq \left(\left(\frac{\epsilon}{\eta_L}\right)^{\beta} + \left(\frac{h_L}{\epsilon}\right)^{\gamma}\right)^{-1} \left(\frac{\eta_L}{h_L}\right)^{2}.$$

12. Weighted multi-level Monte Carlo

Previously, we approximated $E[G_L]$ with the multi-level Monte Carlo method, where G_L is the quantity of interest at the highest level of accuracy. To approximate $E[G_L]$ we used lower levels of accuracy, G_l. In (10.1) we introduced the unbiased estimator $E^L(G_L)$, i.e., when an infinitely large number of samples is considered, this estimator converges to $E[G_L]$.

However, in some cases (cf. Section 13.3), this turns out to be restrictive, and it is easier to approximate $\sum_{l=1}^{L} w_l E[G_l]$ (where the weights satisfy $\sum_{l=1}^{L} w_l = 1$) than $E[G_L]$. This is the weighted multi-level Monte Carlo approach. For the special case $w_L = 1$ and $w_l = 0$ for $1 \leq l < L$ we get the previously introduced MLMC method.

Analogous to (4.16) we assume in each level

$$\|G - G_l\| \leq \delta_l$$

with $\delta_1 > \delta_2 > \cdots > \delta_L$ and

$$\|G\| = \left(E[\|G(\omega,\cdot)\|_V^2]\right)^{\frac{1}{2}} \tag{12.1}$$

where V is a Hilbert space and $G(\omega,\cdot) \in V$. As before we consider a telescopic sum

$$\sum_{l=1}^{L} w_l G_l = \sum_{l=1}^{L} \alpha_l (G_l - G_{l-1})$$

and we get

$$\alpha_l = \sum_{i=l}^{L} w_i.$$

If we choose $w_1 \leq w_2 \leq \cdots \leq w_L$ with $\sum_{l=1}^{L} w_l = 1$ we have

$$1 = \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_L = w_L.$$

By considering the weighted sum $\sum_{l=1}^{L} w_l G_l$ instead of the quantity at level L, G_L, we introduce a new systematic error, we get

$$\|G - \sum_{l=1}^{L} w_l G_l\| = \|\sum_{l=1}^{L} w_l (G - G_l)\| \leq \sum_{l=1}^{L} w_l \|G - G_l\| \leq \sum_{l=1}^{L} w_l \delta_l. \tag{12.2}$$

Note that in the MLMC approach the systematic error is of the size $\|G - G_L\| \leq \delta_L$. Now we approximate the expected value of the weighted sum

$$\sum_{l=1}^{L} w_l E[G_l] = \sum_{l=1}^{L} \alpha_l E[G_l - G_{l-1}] \approx \sum_{l=1}^{L} \alpha_l E_{M_l}(G_l - G_{l-1})$$

with $M_1 \geq M_2 \geq \cdots \geq M_L$. We get the following error

$$\begin{aligned}
& \|E\left[\sum_{l=1}^{L} w_l G_l\right] - \sum_{l=1}^{L} \alpha_l E_{M_l}(G_l - G_{l-1})\| \\
= \quad & \|\sum_{l=1}^{L} \alpha_l (E - E_{M_l})(G_l - G_{l-1})\| \\
\lesssim \quad & \sum_{l=1}^{L} \frac{\alpha_l}{\sqrt{M_l}} \|G - G_l\| + \sum_{l=2}^{L} \frac{\alpha_l}{\sqrt{M_l}} \|G - G_{l-1}\| + \frac{\alpha_1}{\sqrt{M_1}} \|G - E[G]\| \\
\leq \quad & \sum_{l=2}^{L} \frac{\alpha_l}{\sqrt{M_l}} (\delta_l + \delta_{l-1}) + \frac{\alpha_1}{\sqrt{M_1}} \left(\delta_1 + \|G - E[G]\|\right).
\end{aligned}$$

If the error is fixed, the optimal choice for the number M_l of realizations at level l is reached when these error parts are equilibrated. We choose

$$M_l = \begin{cases} \left(\frac{\alpha_1}{\delta_L}\right)^2 \left(\delta_1 + \|G - E[G]\|\right)^2, & l = 1 \\ \left(\frac{\alpha_l}{\delta_L}\right)^2 (\delta_l + \delta_{l-1})^2, & 2 \leq l \leq L. \end{cases}$$

Then we get

$$\|E\left(\sum_{l=1}^{L} w_l G_l\right) - \sum_{l=1}^{L} \alpha_l E_{M_l}(G_l - G_{l-1})\| = O(\delta_L).$$

This error is of the same order as in the MLMC approach though one needs to exercise caution regarding the systematic error (see (12.2)).

13. Multi-level Monte Carlo for the homogenized solution

In this section we use multi-level Monte Carlo to estimate the expectation of the homogenized solution. In particular, we consider $V = H_0^1(D)$ in (12.1). With v_l we denote the projection of v onto a finite dimensional subspace with mesh width H_l, the finite element or finite volume space for example. Then we assume to control the discretization error

$$\|v - v_l\| \lesssim H_l,$$

13.1. Standard MLMC

As a first step we assume no error due to upscaling the coefficient, i.e the error in each level for each realization depends only on the discretization error. In this case we get standard MLMC, which one can find in [10]. We briefly mention their results here.
They find the error bound, cf.[10], Th.4.5,

$$\|E[u] - E^L(u_L)\| \lesssim H_L,$$

with M_l samples on mesh level l given by

$$M_l = l^{2+2\iota} \left(\frac{H_l}{H_L}\right)^2 = l^{2+2\iota} 2^{2(L-l)}, \quad l = 1, \cdots, L \tag{13.1}$$

for an arbitrarily small $\iota > 0$ and

$$W_{coarse}^{MLMC} \leq C_\iota \begin{cases} H_L^{-4} & \text{for } d = 1, \\ H_L^{-2}(\log H_L^{-2})^{3+\iota} & \text{for } d = 2, \\ H_L^{-2}(\log H_L^{-2})^{2+\iota} & \text{for } d = 3, \end{cases}$$

where the constant depends on ι but is independent of L.

13.2. Separable case

In the following we take the error in upscaling the coefficient into account. Note that we assume that the RVE problems are solved exactly as in Section 11. To start with we consider a micro-scale problem with separated scales

$$-\text{div}\left(A(x,\omega)B(\frac{x}{\epsilon},\omega')\nabla u_\epsilon\right) = f \text{ in } D$$

with some boundary conditions. The formally derived homogenized problem is

$$-\,\text{div}\,(A(x,\omega)B^*\nabla u^*) = f \text{ in } D. \tag{13.2}$$

To approximate the expected value of the upscaled solution u^* we consider the tuples (H_l, M_l, η_l, m_l) for $1 \leq l \leq L$, with the coarse mesh size H_l, RVE size η_l and M_l coarse-grid and m_l local problem solves.

To calculate the homogenized coefficient

$$\left(B_l^{*i}\right)_{n,m} = \frac{1}{|Y_{\eta_l}|}\int_{Y_{\eta_l}} \nabla\chi_n^i \cdot B_l^i \nabla\chi_m^i \, dy$$

we solve in each direction, $1 \leq j \leq d$, m_l corresponding deterministic RVE problems

$$\begin{aligned} -\text{div}\left(B_l^i(y)\nabla\chi_j^i\right) &= 0,\ y \in Y_{\eta_l}, \\ \chi_j^i &= e_j \cdot y \text{ on } \partial Y_{\eta_l}. \end{aligned} \tag{13.3}$$

As before, we denote the arithmetic mean of these coefficients with $E_{m_l}(B_l^*)$, i.e.,

$$E_{m_l}(B_l^*) = \frac{1}{m_l}\sum_{i=1}^{m_l} B_l^{*i}.$$

Then, we compute the solution of the deterministic homogenized problem

$$-\,\text{div}\left(A_l^k(x)E_{m_l}(B_l^*)\nabla u_l^k\right) = f,\ 1 \leq k \leq M_l, \tag{13.4}$$

for each mesh size H_l and realization. Here $A_l^k(x)$ is the kth realization of $A(x,\omega)$. Note we use a simplistic treatment for averaging over ω', since we assume that most of the randomness is in $A(x,\omega)$. In general one has to solve the homogenized problem for each realization of B_l^*. We calculate the empirical mean $E_{M_l}(u_l) = \frac{1}{M_l}\sum_{k=1}^{M_l} u_l^k$ of the solutions of (13.4). We use this mean $E_{M_l}(u_l)$ to approximate the expected value $E[u_l]$ of the solution u_l of the stochastic homogenized problem

$$-\,\text{div}\,(A(x,\omega)E_{m_l}(B_l^*)\nabla u_l) = f \text{ in } D \tag{13.5}$$

with mesh size H_l and the homogenized coefficients calculated with RVE size η_l. With $u_0^k := 0$, $1 \leq k \leq M_1$, we define

$$E^L(u_L) = \sum_{l=1}^{L} E_{M_l}(u_l - u_{l-1}).$$

We use $E^L(u_L)$ as approximation of $E[u^*]$. Then we get

$$\begin{aligned}
\|E[u_L] - E^L(u_L)\| &= \|E[\sum_{l=1}^{L} u_l - u_{l-1}] - \sum_{l=1}^{L} E_{M_l}(u_l - u_{l-1})\| \\
&\leq \sum_{l=1}^{L} \|(E - E_{M_l})(u_l - u_{l-1})\| \\
&= \sum_{l=1}^{L} \|(E - E_{M_l})(u_l - u_{l-1})\| \\
&\leq \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}} \|u_l - u_{l-1} - E[u_l - u_{l-1}]\| \\
&\leq \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}} \|u_l - u_{l-1}\| \\
&= \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}} \|u_l - u^* + u^* - u_{l-1}\| \\
&\leq \sum_{l=2}^{L} \frac{1}{\sqrt{M_l}} \left(\|u^* - u_l\| + \|u^* - u_{l-1}\|\right) + \frac{1}{\sqrt{M_1}} \left(\|u^* - u_1\| + \|u^*\|\right).
\end{aligned}$$

Here we used (10.4) and

$$\|G - E[G]\|^2 = E[\|G - E[G]\|_V^2] = E\|G\|_V^2 - \|E[G]\|_V^2 \leq E\|G\|_V^2 = \|G\|^2.$$

To estimate $\|E[u_L] - E^L(u_L)\|$ we need an error bound for $\|u^* - u_l\|$, $1 \leq l \leq L$. Therefore we denote with $u_{H_l}^*$ the solution of

$$-\operatorname{div}\left(A(x,\omega)B^* \nabla u_{H_l}^*\right) = f, \tag{13.6}$$

i.e., $u_{H_l}^*$ is the solution solved with mesh size H_l but without any upscaling error of the coefficient B. It follows

$$\|u^* - u_l\| \quad \leq \quad \|u^* - u_{H_l}^*\| + \|u_{H_l}^* - u_l\|.$$

Thereby $\|u^* - u_{H_l}^*\|$ denotes the discretization error and we have

$$\|u^* - u_{H_l}^*\| \lesssim H_l$$

For the remaining term it holds

$$\|u_{H_l}^* - u_l\| \lesssim \sqrt{E\left[|E_{m_l}(B_l^*) - B^*|^2\right]}.$$

Proof. We write

$$-\text{div}\left(A(x,\omega)B^*\nabla u^*_{H_l}\right) = -\text{div}\left(A(x,\omega)E_{m_l}(B^*_l)\nabla u_l\right).$$

Then,

$$\text{div}\left(A(x,\omega)B^*\nabla(u^*_{H_l} - u_l)\right) = \text{div}\left(A(x,\omega)(E_{m_l}(B^*_l) - B^*)\nabla u_l\right).$$

Multiplying both sides by $(u^*_{H_l} - u_l)$ and integrating by parts (note that $(u^*_{H_l} - u_l) = 0$ on the boundary), we get

$$\begin{aligned}
\int |\nabla(u^*_{H_l} - u_l))|^2 \quad &\lesssim \quad \int A(x,\omega)B^*|\nabla(u^*_{H_l} - u_l))|^2 \\
&= \quad \int A(x,\omega)(E_{m_l}(B^*_l) - B^*)\nabla u_l \cdot \nabla(u^*_{H_l} - u_l) \\
&\lesssim \quad |E_{m_l}(B^*_l) - B^*| \int \nabla u_l \cdot \nabla(u^*_{H_l} - u_l) \\
&\overset{C.S.}{\leq} \quad |E_{m_l}(B^*_l) - B^*|(\int |\nabla u_l|^2)^{1/2}(\int |\nabla(u^*_{H_l} - u_l)|^2)^{1/2}.
\end{aligned}$$

From here, we have

$$\|u^*_{H_l} - u_l\|_V \lesssim |E_{m_l}(B^*_l) - B^*|\|u^*\|_V$$

and therefore

$$\|u^*_{H_l} - u_l\|^2 = E\left[\|u^*_{H_l} - u_l\|^2_V\right] \lesssim E\left[|E_{m_l}(B^*_l) - B^*|^2\right] \|u^*\|^2.$$

□

To get an error estimate for the MLMC approach, we note that

$$\begin{aligned}
\sqrt{E\left[|E_{m_l}(B^*_l) - B^*|^2\right]} \quad &\leq \quad \sqrt{E\left[|E_{m_l}(B^*_l) - E[B^*_l]|^2\right]} + \sqrt{E\left[|E[B^*_l] - B^*|^2\right]} \\
&\lesssim \quad \frac{1}{\sqrt{m_l}}\sqrt{E\left[|(B^*_l) - E[B^*_l]|^2\right]} + \delta_l.
\end{aligned}$$

Using our assumption

$$\delta_l^2 \lesssim \left(\frac{\epsilon}{\eta_l}\right)^\beta, \quad \beta > 0,$$

we get

$$\begin{aligned}
\|E[u_L] - E^L(u_L)\| \quad &\lesssim \quad \sum_{l=1}^{L} \frac{1}{\sqrt{M_l}}\left(H_l + \left(\frac{\epsilon}{\eta_l}\right)^{\frac{\beta}{2}} + \frac{C_l}{\sqrt{m_l}} + H_{l-1} + \left(\frac{\epsilon}{\eta_{l-1}}\right)^{\frac{\beta}{2}} + \frac{C_{l-1}}{\sqrt{m_{l-1}}}\right) \\
&\quad + \frac{1}{\sqrt{M_1}}\left(\|u^*\| + H_l + \left(\frac{\epsilon}{\eta_1}\right)^{\frac{\beta}{2}} + \frac{C_1}{\sqrt{m_1}}\right).
\end{aligned}$$

For a standard MC approach, the error is given by

$$\|E[u_{MC}] - E_{\hat{M}}(u_{MC})\| \quad \lesssim \quad \frac{1}{\sqrt{\hat{M}}}.$$

13.3. General case. Application of weighted MLMC

In this section we consider the general coefficient $A(x, \omega, \frac{x}{\epsilon})$. Here one has to solve RVE problems in many macro-grid points. At every level we will solve a local problem in each coarse-grid block. The sizes of the RVE can be different at different locations. At every level l we denote the number of RVE problems with P_l ($\propto H_l^{-2}$). This number coincides with the number of coarse-grid blocks. We denote the set of coarse-grid points where we solve the local problems at level l with $\mathcal{P}_l$. We assume these sets are nested, i.e., $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \cdots \subset \mathcal{P}_L$. As before we solve for M_l samples coarse-grid problems with mesh size H_l. We calculate the coefficient by solving RVE problems for each realization and averaging the energy over the spatial domain. If we calculate $K^*_{\eta_l,H_l}$ with RVE size η_l, we get $K^*_{\eta_l,H_j}$ for $j < l$ at the same coarse-grid points, automatically. This is true, since the sets of coarse-grid points $\mathcal{P}_l$ are nested. So we only solve RVE problems for $M_l - M_{l+1}$ independent realizations of size η_l with coarse mesh size H_l and get M_l coefficients with different RVE sizes as shown in Table 3 for illustration. Let u_{η_j,H_i} be the solution with a coarse grid with mesh size H_i and RVE size η_j. Instead of $E[u^*]$ we interested in the approximation of $E[\tilde{u}]$ for some $\tilde{u}$. To benefit from the different numbers of effective coefficients for the different grid and RVE size we take

	H_L	H_{L-1}	$\cdots$	H_1	# coefficients to calculate with RVE size η_l
η_1				$K^*_{\eta_1,H_1}$	$M_1 - M_2$
$\vdots$				$\vdots$	$\vdots$
η_{L-1}		$K^*_{\eta_{L-1},H_{L-1}}$	$\cdots$	$K^*_{\eta_{L-1},H_1}$	$M_{L-1} - M_L$
η_L	$K^*_{\eta_L,H_L}$	$K^*_{\eta_L,H_{L-1}}$	$\cdots$	$K^*_{\eta_L,H_1}$	M_L
# coefficients on mesh size H_l	M_L	M_{L-1}	$\cdots$	M_1	

Table 3: Calculating the (blue) coefficients on the diagonal will automatically give the lower triangular values in the matrix.

$$\tilde{u} = \sum_{l=1}^{L} \frac{\alpha_l}{M_l} \sum_{j=l}^{L} (M_j - M_{j+1})(u_{\eta_j,H_l} - u_{\eta_j,H_{l-1}}) \tag{13.7}$$

for $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_L > 0$, where we assume $M_{L+1} = 0$ and $u_{\eta_j,H_0} = 0$ for $1 \leq j \leq L$.

As mentioned in Section 12 we introduce a different systematic error. To compute this we write

$$\tilde{u} = \sum_{l=1}^{L-1} \sum_{j=l+1}^{L} \left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}} \right) (M_j - M_{j+1}) u_{\eta_j,H_l} + \sum_{l=1}^{L} \alpha_l \frac{M_l - M_{l+1}}{M_l} u_{\eta_l,H_l}.$$

This is true since

$$\begin{aligned}
\tilde{u} &= \sum_{l=1}^{L} \frac{\alpha_l}{M_l} \sum_{j=l}^{L} (M_j - M_{j+1})(u_{\eta_j,H_l} - u_{\eta_j,H_{l-1}}) \\
&= \frac{\alpha_L}{M_L} M_L (u_{\eta_L,H_L} - u_{\eta_L,H_{L-1}}) \\
&\quad + \frac{\alpha_{L-1}}{M_{L-1}} \Big[(M_{L-1} - M_L)(u_{\eta_{L-1},H_{L-1}} - u_{\eta_{L-1},H_{L-2}})
\end{aligned}$$

$$
\begin{aligned}
&\qquad\qquad +M_L(u_{\eta_L,H_{L-1}} - u_{\eta_L,H_{L-2}})\Big] \\
&\quad +\frac{\alpha_{L-2}}{M_{L-2}}\Big[(M_{L-2}-M_{L-1})(u_{\eta_{L-2},H_{L-2}} - u_{\eta_{L-2},H_{L-3}}) \\
&\qquad\qquad +(M_{L-1}-M_L)(u_{\eta_{L-1},H_{L-2}} - u_{\eta_{L-1},H_{L-3}}) \\
&\qquad\qquad +M_L(u_{\eta_L,H_{L-2}} - u_{\eta_L,H_{L-3}})\Big] \\
&\quad +\cdots \\
= &\quad \frac{\alpha_L}{M_L}M_L u_{\eta_L,H_L} + \left(\frac{\alpha_{L-1}}{M_{L-1}} - \frac{\alpha_L}{M_L}\right) M_L u_{\eta_L,H_{L-1}} \\
&\quad +\frac{\alpha_{L-1}}{M_{L-1}}(M_{L-1}-M_L)u_{\eta_{L-1},H_{L-1}} \\
&\quad +\left(\frac{\alpha_{L-2}}{M_{L-2}} - \frac{\alpha_{L-1}}{M_{L-1}}\right)(M_{L-1}-M_L)u_{\eta_L,H_{L-1}} \\
&\quad +\left(\frac{\alpha_{L-2}}{M_{L-2}} - \frac{\alpha_{L-1}}{M_{L-1}}\right)M_L u_{\eta_L,H_L} \\
&\quad +\cdots \\
= &\quad \sum_{l=1}^{L-1}\sum_{j=l+1}^{L}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)(M_j - M_{j+1})u_{\eta_j,H_l} + \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l}u_{\eta_l,H_l}.
\end{aligned}
$$

If we use $\|u^* - u_{\eta_j,H_l}\| \le H_l + \delta_j$ the new systematic error reads

$$
\begin{aligned}
& \|Cu^* - \tilde{u}\| \\
\le & \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l}\|u^* - u_{\eta_l,H_l}\| + \sum_{l=1}^{L-1}\sum_{j=l+1}^{L}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)(M_j - M_{j+1})\|u^* - u_{\eta_j,H_l}\| \\
\le & \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l}(H_l + \delta_l) + \sum_{l=1}^{L-1}\sum_{j=l+1}^{L}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)(M_j - M_{j+1})(H_l + \delta_j) \\
\le & \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l}(H_l + \delta_l) + \sum_{l=1}^{L-1}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)(H_l + \delta_l)\sum_{j=l+1}^{L}(M_j - M_{j+1}) \\
= & \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l}(H_l + \delta_l) + \sum_{l=1}^{L-1}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)(H_l + \delta_l)M_{l+1} \\
= & \sum_{l=1}^{L-1}(H_l + \delta_l)\Big[\alpha_l - \alpha_l\frac{M_{l+1}}{M-l} + \alpha_l\frac{M_{l+1}}{M-l} - \alpha_{l+1}\Big] + (H_L + \delta_L)\alpha_L \\
= & \sum_{l=1}^{L}(H_l + \delta_l)\Big[\alpha_l - \alpha_{l+1}\Big]
\end{aligned}
$$

with

$$
C = \sum_{l=1}^{L}\alpha_l\frac{M_l - M_{l+1}}{M_l} + \sum_{l=1}^{L-1}\left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}}\right)\sum_{j=l+1}^{L}(M_j - M_{j+1})
$$

$$\begin{aligned}
&= \sum_{l=1}^{L} \alpha_l \frac{M_l - M_{l+1}}{M_l} + \sum_{l=1}^{L-1} \left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}} \right) M_{l+1} \\
&= \sum_{l=1}^{L} \left(\alpha_l - \alpha_l \frac{M_{l+1}}{M_l} \right) + \sum_{l=1}^{L-1} \left(-\alpha_{l+1} + \alpha_l \frac{M_{l+1}}{M_l} \right) \\
&= \sum_{l=1}^{L} \alpha_l - \sum_{l=1}^{L} \alpha_l \frac{M_{l+1}}{M_l} - \sum_{l=1}^{L-1} \alpha_{l+1} + \sum_{l=1}^{L-1} \alpha_l \frac{M_{l+1}}{M_l} \\
&= \alpha_1 .
\end{aligned}$$

If we choose

$$\alpha_l = \sum_{j=l}^{L} \widetilde{\alpha}_j \frac{H_L + \delta_L}{H_j + \delta_j}$$

we get

$$\|Cu^* - \tilde{u}\| \le (H_L + \delta_L) \sum_{l=1}^{L} \widetilde{\alpha}_l$$

and

$$C = \sum_{j=1}^{L} \widetilde{\alpha}_j \frac{H_L + \delta_L}{H_j + \delta_j}.$$

For a given mesh size $\hat{H}$ and RVE size $\hat{\eta}$ the systematic error for standard MC is

$$\|u^* - u_{\hat{\eta},\hat{H}}\| \le \hat{H} + \hat{\delta}.$$

To have the same systematic error we choose

$$\begin{aligned}
\hat{H} &= \sum_{l=1}^{L} (\alpha_l - \alpha_{l+1}) H_l \\
\hat{\delta} &= \sum_{l=1}^{L} (\alpha_l - \alpha_{l+1}) \delta_l .
\end{aligned}$$

For the expected value of $\tilde{u}$ we get

$$\begin{aligned}
E[\tilde{u}] &= \sum_{l=1}^{L-1} \sum_{j=l+1}^{L} \left(\frac{\alpha_l}{M_l} - \frac{\alpha_{l+1}}{M_{l+1}} \right) (M_j - M_{j+1}) E[u_{\eta_j, H_l}] + \sum_{l=1}^{L} \alpha_l \frac{M_l - M_{l+1}}{M_l} E[u_{\eta_l, H_l}] \\
&= \sum_{l=1}^{L} \frac{\alpha_l}{M_l} \sum_{j=l}^{L} (M_j - M_{j+1}) E[u_{\eta_j, H_l} - u_{\eta_j, H_{l-1}}].
\end{aligned}$$

As multi-level approximation we use

$$E^{L*}(\tilde{u}) = \sum_{l=1}^{L} \alpha_l E^*_{M_l}(u_l - u_{l-1}) \tag{13.8}$$

instead of the sum of the arithmetic means at different levels as before. Where $E^*_{M_l}(u_l - u_{l-1})$ is defined as

$$E^*_{M_l}(u_l - u_{l-1}) \quad := \quad \frac{1}{M_l} \sum_{j=l}^{L} (M_j - M_{j+1}) E_{M_j - M_{j+1}}(u_{\eta_j, H_l} - u_{\eta_j, H_{l-1}})$$

$$= \frac{1}{M_l}\sum_{j=l}^{L}\left(\sum_{i=1}^{M_j-M_{j+1}}\left(u_{\eta_j,H_l}-u_{\eta_j,H_{l-1}}\right)(\omega_i)\right).$$

For clarity we summarize the main steps of weighted MLMC for the coarse grid problem:

1. Generate M_1 random variables $\omega_1\cdots\omega_{M_1}$.
2. For each level l, $1\le l\le L$, and each realization ω_j, $M_{l+1}<j\le M_l$, $(M_{L+1}=0)$
   - Solve in each coarse grid block with mesh size H_l the RVE problems
   $$\begin{aligned}\mathrm{div}(A_\epsilon(x,\omega_j)\nabla\chi_i^j) &= 0, \text{ in } Y_{\eta_l},\ i=1,...,d\\ \chi_i^j &= x_i \text{ on } \partial Y_{\eta_l}.\end{aligned}$$
   - Compute in each coarse grid block with mesh size H_l the homogenized coefficients
   $$K^*_{\eta_l,H_l}(x,\omega_j)e_i=\frac{1}{{\eta_l}^d}\int_{Y_{\eta_l}}A_\epsilon(x,\omega_j)\nabla\chi_i^j.$$
   - Solve the coarse grid problem with grid size H_k, for $1\le k\le l$
   $$-\mathrm{div}\left(K^*_{\eta_l,H_k}(x,\omega_j)\nabla u_{\eta_l,H_k}(x,\omega_j)\right)=f(x) \text{ in } D$$
3. For each level l, $1\le l\le L$, compute
$$\begin{aligned}E^*_{M_l}(u_l-u_{l-1})(x) &= \frac{1}{M_l}\sum_{k=l}^{L}\left(\sum_{j=1}^{M_k-M_{k+1}}\left(u_{\eta_k,H_l}-u_{\eta_k,H_{l-1}}\right)(x,\omega_j)\right)\\ &= \frac{1}{M_l}\sum_{k=l}^{L}(M_k-M_{k+1})E_{M_k-M_{k+1}}(u_{\eta_k,H_l}-u_{\eta_k,H_{l-1}})(x).\end{aligned}$$
4. Compute the weighted multi-level approximation of the expected value
$$E^{L*}(\tilde{u})(x)=\sum_{l=1}^{L}\alpha_l E^*_{M_l}(u_l-u_{l-1})(x)$$
with $\alpha_1\ge\alpha_2...\ge\alpha_l$.

To estimate the weighted MLMC error, we have

$$\begin{aligned}&\|E[\tilde{u}]-\sum_{l=1}^{L}\alpha_l E^*_{M_l}(u_l-u_{l-1})\|\\ =\ &\|\sum_{l=1}^{L}\alpha_l\frac{1}{M_l}\sum_{j=l}^{L}(M_j-M_{j+1})(E-E_{M_j-M_{j+1}})(u_{\eta_j,H_l}-u_{\eta_j,H_{l-1}})\|\\ \le\ &\sum_{l=1}^{L}\alpha_l\frac{1}{M_l}\sum_{j=l}^{L}(M_j-M_{j+1})\|(E-E_{M_j-M_{j+1}})(u_{\eta_j,H_l}-u_{\eta_j,H_{l-1}})\|\end{aligned}$$

$$\begin{aligned}
&\lesssim \quad \sum_{l=1}^{L} \alpha_l \frac{1}{M_l} \sum_{j=l}^{L} (M_j - M_{j+1}) \frac{1}{\sqrt{M_j - M_{j+1}}} \| u_{\eta_j, H_l} - u_{\eta_j, H_{l-1}} - E[u_{\eta_j, H_l} - u_{\eta_j, H_{l-1}}] \| \\
&\lesssim \quad \sum_{l=1}^{L} \alpha_l \sum_{j=l}^{L} \frac{\sqrt{M_j - M_{j+1}}}{M_l} \| u_{\eta_j, H_l} - u^* + u^* - u_{\eta_j, H_{l-1}} \| \\
&\lesssim \quad \sum_{l=1}^{L} \alpha_l \sum_{j=l}^{L} \frac{\sqrt{M_j - M_{j+1}}}{M_l} \| u_{\eta_j, H_l} - u^* \| + \alpha_1 \sum_{j=1}^{L} \frac{\sqrt{M_j - M_{j+1}}}{M_1} \| u^* \| \\
&\lesssim \quad \sum_{l=1}^{L} \alpha_l \sum_{j=l}^{L} \frac{\sqrt{M_j - M_{j+1}}}{M_l} (H_l + \delta_j) + \alpha_1 \sum_{j=1}^{L} \frac{\sqrt{M_j - M_{j+1}}}{M_1} \\
&\lesssim \quad \sum_{l=1}^{L} \alpha_l \frac{1}{\sqrt{M_l}} (H_l + \delta_l) + \alpha_1 \frac{1}{\sqrt{M_1}}
\end{aligned}$$

To equate the error terms we choose

$$M_l = C \begin{cases} \left(\frac{\alpha_1}{\gamma_1 (\hat{H} + \hat{\delta})} \right)^2 & l = 1 \\ \left(\frac{\alpha_l (H_l + \delta_l)}{\gamma_l (\hat{H} + \hat{\delta})} \right)^2 & l \geq 2. \end{cases}$$

Then we have

$$\| E[\tilde{u}] - \sum_{l=1}^{L} \alpha_l E^*_{M_l} (u_l - u_{l-1}) \| = O(\hat{H} + \hat{\delta}).$$

In the MC case we choose $\hat{M} = C(\hat{H} + \hat{\delta})^{-2}$ to get an error of the same order. As before the cost of solving the coarse scale problems is

$$W^{MLMC}_{\text{coarse}} = \sum_{l=1}^{L} M_l H_l^{-2}$$

and for single level MC

$$W^{MC}_{\text{coarse}} = \hat{M} \hat{H}^{-2}.$$

The dominating part of the computational cost is the solution of RVE problems. We assume we solve at $P_l \leq H_l^{-2}$ points RVE problems with the nested sets of points of size $P_1 < P_2 \cdots < P_L$. For MC we assume we solve at $\hat{P}$ points. Then, we get

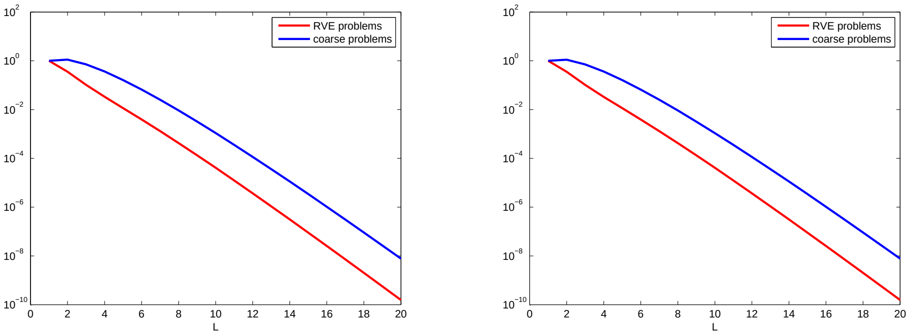
$$W^{MC}_{\text{RVE}} = \hat{M} \hat{P} \left(\frac{\hat{\eta}}{\epsilon} \right)^2 = C \frac{\hat{P} \hat{\eta}^2}{\epsilon^2 (\hat{H} + \hat{\delta})^2}.$$

Note that for the weighted MLMC approach we solve $M_l - M_{l+1}$ RVE problems for η_l. For MLMC, we achieve the following work

$$\begin{aligned}
W^{MLMC}_{\text{RVE}} \quad &= \quad \sum_{l=1}^{L} (M_l - M_{l+1}) \left(\frac{\eta_l}{\epsilon} \right)^2 P_l \\
&= \quad \sum_{l=1}^{L} M_l \left(\frac{\eta_l}{\epsilon} \right)^2 P_l - \sum_{l=1}^{L} M_{l+1} \left(\frac{\eta_l}{\epsilon} \right)^2 P_l
\end{aligned}$$

$$\begin{aligned}
&= \sum_{l=1}^{L} M_l \left(\frac{\eta_l}{\epsilon}\right)^2 P_l - \sum_{l=2}^{L} M_l \left(\frac{\eta_{l-1}}{\epsilon}\right)^2 P_{l-1} \\
&= M_1 \left(\frac{\eta_1}{\epsilon}\right)^2 P_1 + \sum_{l=2}^{L} M_l \left(\left(\frac{\eta_l}{\epsilon}\right)^2 P_l - \left(\frac{\eta_{l-1}}{\epsilon}\right)^2 P_{l-1}\right) \\
&= C\left(\left(\frac{\alpha_1}{\gamma_1(\hat{H}+\hat{\delta})}\right)^2 \left(\frac{\eta_1}{\epsilon}\right)^2 P_1 + \sum_{l=2}^{L}\left(\frac{\alpha_l(H_l+\delta_l)}{\gamma_l(\hat{H}+\hat{\delta})}\right)^2 \left(\left(\frac{\eta_l}{\epsilon}\right)^2 P_l - \left(\frac{\eta_{l-1}}{\epsilon}\right)^2 P_{l-1}\right)\right) \\
&= C\frac{\hat{P}\hat{\eta}^2}{\epsilon^2(\hat{H}+\hat{\delta})^2}\left(\left(\frac{\eta_1}{\hat{\eta}}\right)^2 \frac{\alpha_1^2 P_1}{\gamma_1^2 \hat{P}} + \sum_{l=2}^{L}\left(\frac{\alpha_l}{\gamma_l}\right)^2 (H_l+\delta_l)^2 \left(\left(\frac{\eta_l}{\hat{\eta}}\right)^2 \frac{P_l}{\hat{P}} - \left(\frac{\eta_{l-1}}{\hat{\eta}}\right)^2 \frac{P_{l-1}}{\hat{P}}\right)\right)
\end{aligned}$$

In Figure 14 we illustrate the ratios of the work of weighted MLMC and MC for solving the coarse problems and the RVE problems. As we can see, the cost ratio decreases if the number of levels L increases.



(a) $\epsilon = \frac{\eta_1}{10}$

(b) $\epsilon = \frac{2^{-50}}{10}$

Figure 14: Work ratios $\frac{W_{\text{RVE}}^{MLMC}}{W_{\text{RVE}}^{MC}}$ and $\frac{W_{\text{coarse}}^{MLMC}}{W_{\text{coarse}}^{MC}}$ for up to 20 levels, $\beta = 2$, $\eta_l = 2^{-L+l}$, $\gamma_l = \frac{1}{L}$ and $\alpha_l = \sum_{i=l}^{L} \frac{1}{L} \frac{H_L + \left(\frac{\epsilon}{\eta_L}\right)^\beta}{H_i + \left(\frac{\epsilon}{\eta_i}\right)^\beta} \left(\sum_{i=1}^{L} \frac{1}{L} \frac{H_L + \left(\frac{\epsilon}{\eta_L}\right)^\beta}{H_i + \left(\frac{\epsilon}{\eta_i}\right)^\beta}\right)^{-1}$.

14. Numerical results

In this section we present some representative numerical results. We consider the problem (in dimension d)

$$-\text{div}\left[K(x, \omega, \frac{x}{\epsilon}, \omega')\nabla u\right] = f \text{ in } D = (0,1)^d.$$

The boundary conditions and the function f will be given below. Note that the homogenized coefficient is independent of these choices. In the following we compare our MLMC results with standard MC results at the highest level. In contrast to our theoretical analysis where we equate the error and compare the computational costs, we equate the costs for calculating the coefficient and the solution separately and compare the errors in the numerics. We will consider one-dimensional and two-dimensional examples. We have implemented the one-dimensional methods in Matlab ([2])

and analytical solutions are known in most of the examples. As previously, we use the modular toolbox DUNE ([12, 11, 24, 14]) for the two-dimensional problems. Here we use the cell-centered finite volume method as described in Section 7. In Table 4 we show the different mesh and RVE sizes for three different levels. In the MLMC approach we approximate the homogenized coefficient with $L = 3$ different RVE sizes η_l (cf. Fig. 15) and when we calculate the coarse solution we use $L = 3$ different coarse mesh sizes H_l. Note that we used the same fine mesh h_l for each level. In the two-dimensional examples we generate the coefficient with the Karhunen-Loève expansion and the characteristic length scale is related to the correlation length $\epsilon = \frac{\tau}{\sqrt{2}} = \frac{0.04}{\sqrt{2}}$.

l	H_l	h_l	η_l	# cells in RVE of size η_l
1	$\frac{1}{16}$	$\frac{1}{128}$	0.125	256
2	$\frac{1}{32}$	$\frac{1}{128}$	0.25	1024
3	$\frac{1}{64}$	$\frac{1}{128}$	0.5	4096

Table 4: Different RVE and mesh sizes for the two-dimensional case.

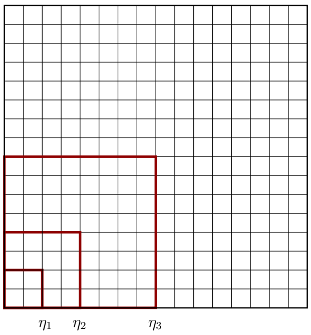


Figure 15: Nested domains for the RVE problem, in dimension $d = 2$.

In Section 14.1 we present numerical results for the homogenized coefficient. First we explain in Section 14.1.1 how to estimate the convergence rate β (cf. (4.16)) numerically and then we consider one-dimensional (cf. Section 14.1.2) and two-dimensional (cf. Section 14.1.3) results for approximating the upscaled coefficient with MLMC. We present numerical results for the coarse solution using weighted MLMC in Section 14.2.

14.1. Numerical results of the homogenized coefficient

As a first step we estimate the convergence rate β numerically. This is important in order to choose later in this section the number m_l of realizations at each level l in the one-dimensional (cf. Section 14.1.2) and two-dimensional (cf. Section 14.1.3) examples using MLMC appropriately.

14.1.1. Numerical study of the convergence rate

For our theoretical study we assume (4.16) holds, i.e.,

$$E\left[\left|\left|\left|K^*_{\eta_l} - K^*\right|\right|\right|^2\right] \leq C\left(\frac{\epsilon}{\eta_l}\right)^{\beta}$$

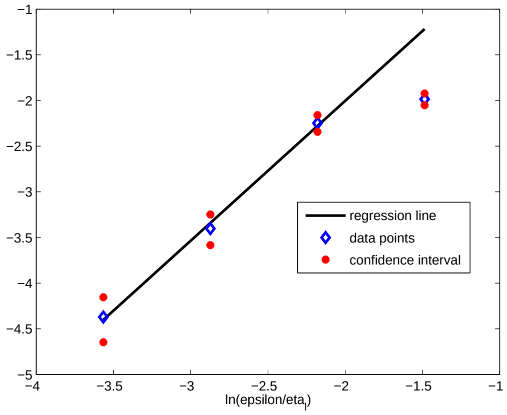


Figure 16: Computed data points with corresponding regression line with slope β and upper and lower points of the confidence interval. 4 different levels,$\mathfrak{m}_l = (2000, 1000, 300, 140)$, $\beta = 1.53$ and $\ln C = 1.059$.

for some constant C and rate β independent of ϵ and η. In this section we estimate the convergence rate β numerically. Therefor we consider as coefficient a scalar random field $K(\frac{x}{\epsilon}, \omega')$ defined for $x \in D \subset \mathbb{R}^2$ with expected value $E[K] = 10$ (independent of x and ϵ) and Gaussian covariance function $\mathrm{cov}(x, x') = \sigma^2 \exp(-\frac{|x-x'|^2}{\tau^2})$ with standard deviation $\sigma = \sqrt{2}$ and $\tau = \sqrt{2}\epsilon = 0.04$ (recall that $|x - x'|$ denotes the Euclidian distance in $\mathbb{R}^2$). We generate samples of the coefficient with the Karhunen-Loève expansion. For any $1 \leq l \leq L$, we calculate the effective coefficients $K_l^*(\omega_j')$ for the RVE $[0, \eta_l]^2$ (with $\eta_l = 0.5^{L-l}$) for various realizations ω_j', $1 \leq j \leq m_l$. Since we cannot access the theoretical reference value $K^* = \lim_{\eta\to\infty} E[K_\eta^*]$ in practice, we define the reference value as

$$K_{ref}^* := \frac{1}{L}\sum_{l=1}^{L}\frac{1}{m_l}\sum_{j=1}^{m_l} K_l^*(\omega_j')$$

where we use all the realizations on the RVEs $[0, \eta_l]$, $1 \leq l \leq L$. For this choice of reference it holds $E[K_{ref}^*] \neq E[K_L^*]$. Since it is expensive to compute the effective coefficient with RVE $[0, \eta_L]$ the statistical error would be large and we cannot use the unbiased estimator $\frac{1}{m_L}\sum_{j=1}^{m_L} K_L^*(\omega_j')$. In practice, we consider only the first entry $[K_l^*]_{11}$ and we use four levels and $\mathfrak{m}_l = (2000, 1000, 300, 140)$. At each level , $1 \leq l \leq L$, we expect

$$\frac{1}{m_l}\sum_{j=1}^{m_l} |K_l^*(\omega_j') - K_{ref}^*|^2 \approx E\left[|K_l^*(\omega_j') - K_{ref}^*|^2\right] \approx C\left(\frac{\epsilon}{\eta_l}\right)^\beta .$$

By taking the logarithm we get

$$\ln(\frac{1}{m_l}\sum_{j=1}^{m_l} |K_l^*(\omega_j') - K_{ref}^*|^2) \approx \beta \ln\left(\frac{\epsilon}{\eta_l}\right) + \ln C.$$

Results are shown in Figure 16, where we plot the computed data points with confidence intervals. It turns out that the data points for the smallest RVE $[0, \eta_1]^2$ show a different behavior than the other data sets. Since the results with the smallest RVE are the least accurate ones, we take only the three larger RVEs into account to compute the rate β. The linear regression line for these results is also plotted in Figure 16. We find a line with slope $\beta = 1.53$ and intercept $\ln C = 1.059$. Note that this rate β is smaller, but close to the theoretical value $\beta^{\text{theo}} = 2$ which would be obtained by a Central Limit theorem argument. This rate β is essential to choose the number of realizations m_l at each level according to (11.2) appropriately. In the following two-dimensional examples we use the RVEs $[0,\eta_1]^2$, $[0,\eta_2]^2$ and $[0,\eta_3]^2$ only. If we take only these RVEs into account we end with a rate β close to 1. Therefore we choose $\beta = 1$ in the two-dimensional examples instead of β^{theo}. In the one-dimensional cases we determine β for each coefficient.

14.1.2. One dimensional examples

First, we present numerical results for one-dimensional problems

$$\frac{d}{dx}\left(K(\frac{x}{\epsilon},\omega,\omega')\,\frac{d}{dx}u(x,\frac{x}{\epsilon},\omega,\omega')\right) = f \text{ in } D \subset \mathbb{R}$$

with some boundary conditions for a coefficient $K(\frac{x}{\epsilon},\omega,\omega')$ such that the local problems are analytically solvable. In one dimension the local problem reads

$$\begin{aligned}\frac{d}{dx}\left(K(\frac{x}{\epsilon},\omega,\omega')\,\frac{d}{dx}\chi(x,\omega,\omega')\right) &= 0, \quad x \in Y =]a,b[\\ \chi(x,\omega,\omega') &= x, \quad x \in \{a,b\}.\end{aligned}$$

For fixed ω and ω' it follows

$$K(\frac{x}{\epsilon},\omega,\omega')\,\frac{d}{dx}\chi(x,\omega,\omega') = C_1$$

and therefore we get

$$\chi(x,\omega,\omega') = C_1\int_a^x \frac{1}{K(\frac{y}{\epsilon},\omega,\omega')}\,dy + C_2.$$

With the help of the boundary conditions we determine the constants

$$\begin{aligned}C_2 &= a,\\ C_1 &= \frac{b-a}{\int_a^b \frac{1}{K(\frac{y}{\epsilon},\omega,\omega')}\,dy}.\end{aligned}$$

As in (4.13) we calculate the homogenized coefficient as

$$\begin{aligned}K^*_{a,b}(\omega,\omega') &= \frac{1}{b-a}\int_a^b K(\frac{y}{\epsilon},\omega,\omega')\frac{d}{dy}\chi(y,\omega,\omega')\,dy\\ &= \frac{1}{b-a}\int_a^b C_1\,dy\\ &= C_1.\end{aligned}$$

So the homogenized coefficient is the harmonic mean, i.e.,

$$K^*_{a,b}(\omega,\omega') = \frac{b-a}{\int_a^b \frac{1}{K(\frac{x}{\epsilon},\omega,\omega')}\,dx}.$$

In the MLMC approach we consider different RVEs $[a_l,b_l]$, $1 \le l \le L$, with size $\eta_l = b_l - a_l$. Analogous to our previous notation we denote the homogenized coefficient at level l, i.e., with RVE size η_l, with

$$K^*_l(\omega,\omega') := K^*_{a_l,b_l}(\omega,\omega').$$

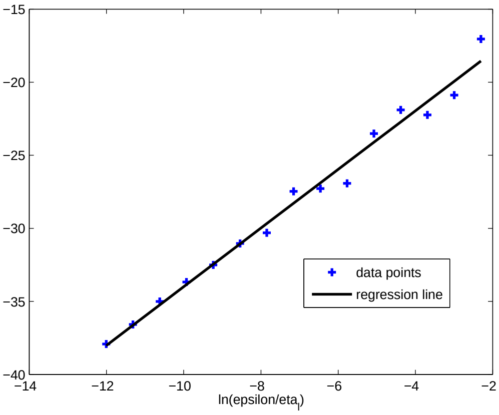


Figure 17: Computed data points with corresponding regression line with slope β. 15 levels, mean over 50000 samples, $\beta = 2.0070$ and $\ln C = -13.9254$.

Separable coefficient **Example 1** As a first step we use the separable coefficient $K(\frac{x}{\epsilon}, \omega, \omega')$ with the following uniformly bounded away from 0 inverse

$$K^{-1}(\frac{x}{\epsilon}, \omega, \omega') = \left(C + \sum_{i=1}^{N} \chi_i(\omega') \sin^2(\frac{2\pi x \varphi_i}{\epsilon}) \right) \exp(\omega),$$

where ω and χ_i are i.i.d. uniformly distributed in $[0,1]$, and φ_i are fixed random numbers in $[0.2, 2]$, and $C > 0$ is deterministic. This coefficient is separable in the sense that K^{-1} is a product of a function of ω and a function of ω'. As mentioned above, the homogenized coefficient is the harmonic mean in this case. Therefore the homogenized coefficient on the RVE $[a, b]$ is

$$\begin{aligned} & K^*_{a,b}(\omega, \omega') \\ = & \left(\frac{1}{b-a} \int_a^b K^{-1}(\frac{x}{\epsilon}, \omega, \omega')\, dx \right)^{-1} \\ = & \left(\frac{\exp(\omega)}{b-a} \left[C(b-a) + \sum_{i=1}^{N} \chi_i(\omega') \left(\frac{b-a}{2} - \frac{\sin\left(\frac{4\pi b \varphi_i}{\epsilon}\right) - \sin\left(\frac{4\pi a \varphi_i}{\epsilon}\right)}{\frac{8\pi\varphi_i}{\epsilon}} \right) \right] \right)^{-1}. \end{aligned}$$

In our simulation we use $C = 1$, $N = 20$ and $\epsilon = \frac{0.5^L}{10}$. With this choice of ϵ we ensure that the smallest RVE (of size 0.5^L) considered in the MLMC approach is much larger than the characteristic length scale of the homogenized coefficient. As described in Section 14.1.1 we determine the convergence rate β for this homogenized coefficient. We consider $L = 15$ different levels with RVEs $[a_l, b_l] = [0, 0.5^{L+1-l}]$ and as reference we use

$$K^*_{ref}(\omega, \omega') = K^*_{0,\infty}(\omega, \omega') = \left(C \exp(\omega) + \frac{\exp(\omega)}{2} \sum_{i=1}^{N} \chi_i(\omega') \right)^{-1}.$$

We get a convergence rate of approximately 2 (cf. Figure 17). That is why we choose in the following for the different numbers of realizations m_l per level for the MLMC approach the ratio

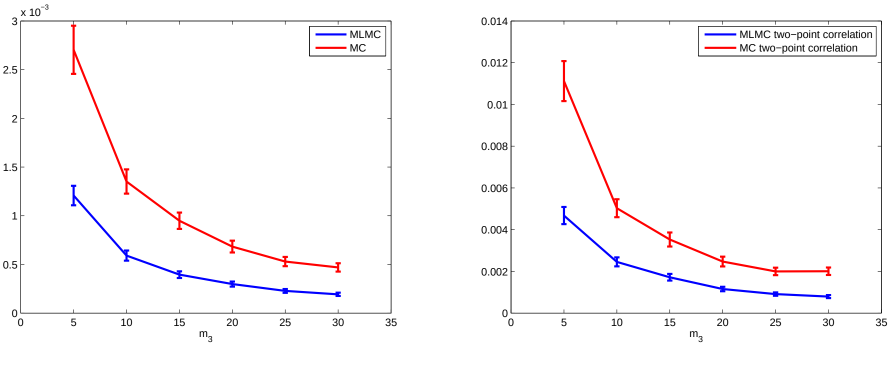
$\frac{m_l}{m_{l+1}} = \left(\frac{\eta_{l+1}}{\eta_l}\right)^\beta = 2^\beta = 4$. One realization is identified with the tuple $(\omega, \chi_1(\omega'), \cdots \chi_N(\omega'))$ and the expectations are taken with respect to both, ω and ω'. We are interested in the relative mean square error between the MLMC approximation $E^L(K_L^*)$ with L levels and the true expected value $E[K_L^*]$ at level L, i.e., $\left(e^{rel}_{MLMC}\right)^2(K_L^*) = \frac{(e_{MLMC})^2(K_L^*)}{\left(E[K_L^*]\right)^2}$ with $e_{MLMC}(K_L^*)$ defined as in (10.2). Therefore we consider the MC approach with 400000 realizations of the coefficient on the largest RVE $[a_L, b_L] = [0, 0.5]$ as reference. For MLMC we use RVEs $[a_l, b_l] = [0, 0.5^{L+1-l}]$ and $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$. For comparison, we calculate the relative error of standard MC $\left(e^{rel}_{MC}\right)^2(K_L^*) = \frac{(e_{MC})^2(K_L^*)}{\left(E[K_L^*]\right)^2}$ (with $e_{MC}(K_L^*)$ defined in (10.3)) with the largest RVE $[a_L, b_L] = [0, 0.5]$ with $\hat{m} = \frac{\sum_{l=1}^L m_l b_l}{b_L}$ samples, so that both approaches have the same costs. Since the errors depend on the set of chosen random numbers we repeat the computations $Nb = 1000$ times and calculate the corresponding confidence intervals

$$\left[\text{mean}((e^{rel})^2) - \frac{2\text{std}((e^{rel})^2)}{\sqrt{Nb}}, \text{mean}((e^{rel})^2) + \frac{2\text{std}((e^{rel})^2)}{\sqrt{Nb}}\right].$$

In Figure 18 we illustrate the relative mean square errors for three different levels for the expected value and the two-point correlation of the effective coefficient with the corresponding confidence intervals.

We can observe that the MLMC approach yields smaller errors for the same amount of work. In both cases, expected value and two-point correlation, the relative mean square error with the standard MC approach is approximately 2.3 times larger than the one for the MLMC approach, i.e.,

$$\frac{\frac{1}{Nb}\sum_{j=1}^{Nb}[e^{rel}_{MC}(K_L^*(\omega_j, \omega_j'))]^2}{\frac{1}{Nb}\sum_{j=1}^{Nb}[e^{rel}_{MLMC}(K_L^*(\omega_j, \omega_j'))]^2} \approx 2.3.$$



(a) Relative mean square errors of the expected value of the effective coefficients.

(b) Relative mean square errors of the two-point correlation of the effective coefficients.

Figure 18: Relative mean square errors with equated costs and $\mathfrak{m} = (16m_3, 4m_3, m_3)$. Example 1 separable coefficient.

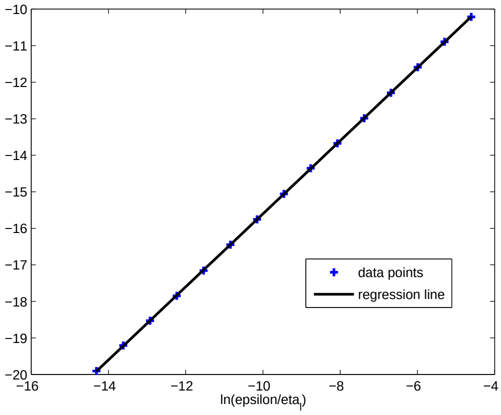


Figure 19: Computed data points with corresponding regression line with slope β. 15 levels, $\epsilon = 1$, mean over 50000 samples, $\beta = 1.0003$ and $\ln C = -5.5997$.

Separable stationary coefficient **Example 2** Here we consider an example where the effective coefficient does not depend on ω', in the limit of infinite large RVEs. We choose

$$K^{-1}(x,\omega,\omega') = \left(C + \sum_{i\in\mathbb{Z}} \chi_i(\omega') 1_{[i,i+1)}(x) \sin^2(2\pi x) \right) \exp(\omega),$$

where ω and χ_i are i.i.d. uniformly distributed in $[0,1]$, $C = 1$ and $1_{[i,i+1)}(x)$ denotes the indicator function which is 1 for $x \in [i, i+1)$ and zero elsewhere. Therefore the homogenized coefficient on the RVE $[a,b]$ (for simplicity we choose a, $b \in \mathbb{Z}$) is

$$\begin{aligned} & K^*_{a,b}(\omega,\omega') \\ = & \left(\frac{1}{b-a} \int_a^b K^{-1}(x,\omega,\omega')\,dx \right)^{-1} \\ = & \left(\frac{\exp(\omega)}{b-a} \left[C(b-a) + 0.5 \sum_{i=a}^{b-1} \chi_i(\omega') \right] \right)^{-1}. \end{aligned}$$

Since K is stationary in the variables (x, ω'), the standard stochastic homogenization theory holds, i.e., the exact effective coefficient is independent of ω'. In this case the exact effective coefficient is

$$K^*(\omega) = \left(E_{\omega'} \left[\int_0^1 K^{-1}(x,\omega,\omega')\,dx \right] \right)^{-1} = \left(\exp(\omega) \left(C + 0.5E[\chi] \right) \right)^{-1}.$$

Note that $\lim_{b-a\to\infty} K^*_{a,b}(\omega,\omega') = K^*(\omega)$ almost surely in ω'. As previously we estimate the convergence rate β of the homogenized coefficient to choose the rate $\frac{m_l}{m_{l+1}}$ of the number of realizations appropriately. Here we use $K^*(\omega)$ as reference, 15 different levels and we consider the RVEs $[a_l, b_l] = [0, 100 \cdot 2^{l-1}]$. In this case we get a convergence rate of approximately 1 (cf. Figure 19). According to this β we choose $\mathfrak{m} = (2^{L-l} m_L, \cdots, 2m_L, m_L)$ in the MLMC approach for the RVEs as mentioned above. We compare these results to a standard MC approach on the largest RVE $[a_L, b_L] = [0, 100 \cdot 2^{L-1}]$ with the same costs, i.e., for MC we choose $\hat{m} =$

$\frac{\sum_{l=1}^{L} m_l b_l}{b_L}$. Following Section 11.1 the theoretical reference is $E[K_L^*(\omega,\omega')]$, where $K_L^*(\omega,\omega')$ is the homogenized coefficient on the largest RVE. We work with a reference value which can be computed analytically and is very close to $E[K_L^*(\omega,\omega')]$ when the RVE at level L is large. In this paragraph as reference we use

$$K_{ref}^* := E[K^*(\omega)] = \left(\exp(0.5)\,(C + 0.5E[\chi])\right)^{-1} = \frac{1-\frac{1}{e}}{C+0.25}.$$

By design it holds $K_{ref}^* = \lim_{b-a\to\infty} E[K_L^*(\omega,\omega')]$.
In this example we consider $L = 3$, 5 or 7. In Figure 20 we compare the relative mean square errors $\left(e_{MLMC}^{rel}\right)^2$ and $\left(e_{MC}^{rel}\right)^2$ for the expected value and the two-point correlation of the effective coefficient to the corresponding confidence intervals for $Nb = 1000$ different sets of random numbers. Again we can observe that MLMC is more accurate for the same amount of work and the gain in accuracy increases if we increase the number of levels. More precisely we have, for both the expected value and the two-point correlation of the coefficient

$$\frac{\frac{1}{Nb}\sum_{j=1}^{Nb}[e_{MC}^{rel}(K_L^*(\omega_j))]^2}{\frac{1}{Nb}\sum_{j=1}^{Nb}[e_{MLMC}^{rel}(K_L^*(\omega_j))]^2} \approx \begin{cases} 1.4, & L=3 \\ 3.2, & L=5 \\ 9.0, & L=7 \end{cases}.$$

Non separable coefficient **Example 3** In this paragraph we use the coefficient

$$K^{-1}(\frac{x}{\epsilon},\omega,\omega') = C(1+\omega) + \exp\left(\omega\omega'\sin(\frac{x}{\epsilon})\right)\cos(\frac{x}{\epsilon})$$

with ω and ω' i.i.d. random variables uniformly distributed in $[0.5, 1]$, $\epsilon = \frac{0.5^L}{10}$ (to ensure that the smallest RVE is large compared to ϵ) and $C = 2e$ (which ensures that the coefficient is uniformly bounded away from 0). In comparison to the two previous examples ω and ω' are not separable. The effective coefficient on $[a, b]$ is

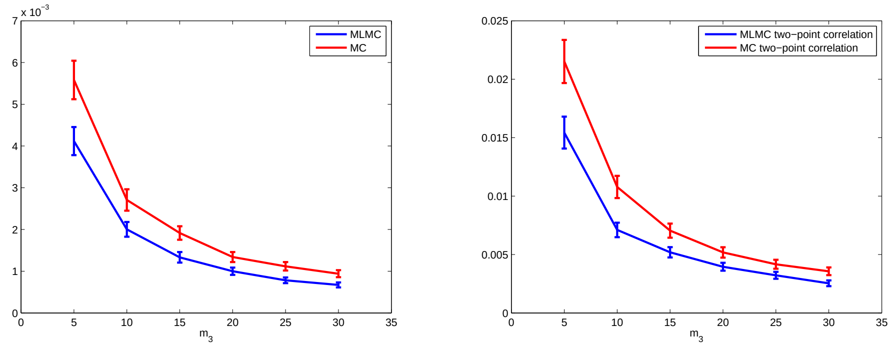
$$\begin{aligned} & K_{a,b}^*(\omega,\omega') \\ = & \left(\frac{1}{b-a}\int_a^b K^{-1}(\frac{x}{\epsilon},\omega,\omega')\,dx\right)^{-1} \\ = & \left(\frac{1}{b-a}\left[C(1+\omega)(b-a) + \frac{\epsilon}{\omega\omega'}\left(\exp\left(\omega\omega'\sin(\frac{b}{\epsilon})\right) - \exp\left(\omega\omega'\sin(\frac{a}{\epsilon})\right)\right)\right]\right)^{-1}. \end{aligned}$$

As in the example with the separable coefficient we consider the RVEs $[a_l, b_l] = [0, 0.5^{L+1-l}]$ for the MLMC approach. To choose the number of realizations m_l for each level appropriately we estimate the convergence rate β numerically. The reference is

$$K^*(\omega) = \lim_{b-a\to\infty} K_{a,b}^*(\omega,\omega') = \frac{1}{C(1+\omega)}.$$

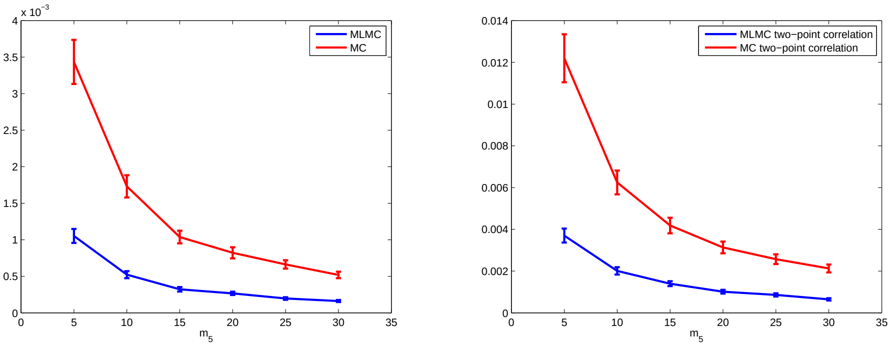
Again we use 15 different levels and the rate β is approximately 2 (cf. Figure 21). Again we compare the accuracy to MC with equated costs on the RVE $[a_L, b_L] = [0, 0.5]$. As in the previous example we use the practical reference value

$$K_{ref}^* = \lim_{b-a\to\infty} E[K_{a,b}^*(\omega,\omega')] = \frac{2\ln\frac{4}{3}}{C}.$$



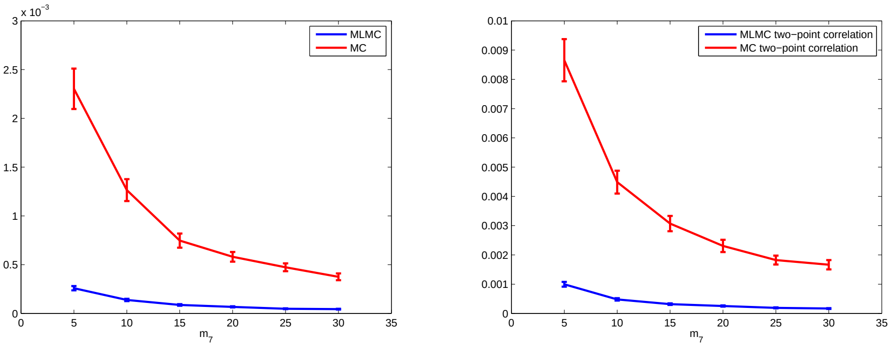
(a) Relative mean square errors of the expected value of the effective coefficients for 3 levels.

(b) Relative mean square errors of the two-point correlation of the effective coefficients for 3 levels.



(c) Relative mean square errors of the expected value of the effective coefficients for 5 levels .

(d) Relative mean square errors of the two-point correlation of the effective coefficients for 5 levels.



(e) Relative mean square errors of the expected value of the effective coefficients for 7 levels .

(f) Relative mean square errors of the two-point correlation of the effective coefficients for 7 levels.

Figure 20: Relative mean square errors with equated costs and $\mathfrak{m} = (2^{L-l}m_L, \cdots, 2m_L, m_L)$. Example 2 separable stationary coefficient.

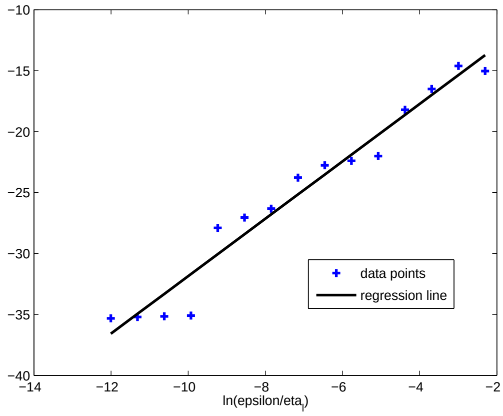


Figure 21: Computed data points with corresponding regression line with slope β. 15 levels, $\epsilon = \frac{0.5^L}{10}$, mean over 50000 samples, $\beta = 2.3548$ and $\ln C = -8.3103$.

In Figure 22 we illustrate the relative mean square errors $\left(\left(e^{rel}_{MLMC}\right)^2, \left(e^{rel}_{MC}\right)^2\right)$ for three different levels for the expected value and the two-point correlation of the effective coefficient with the corresponding confidence intervals for $Nb = 10000$ different sets of random numbers. We can observe that MLMC yields smaller errors for the same amount of work. If we choose $\mathfrak{m} = (2^{L-1}m_L, \cdots, 2m_L, m_L)$ for the MLMC approach both the relative mean square error for the mean and for the two-point correlation is 1.3 times larger for the standard MC approach than for the MLMC approach (cf. Fig. 22(c), 22(d)). However, if we choose the number of realizations according to β, i.e $\mathfrak{m} = (4^{L-1}m_L, \cdots, 4m_L, m_L)$ for the MLMC approach, both the relative mean square error for the mean and for the two-point correlation is two times larger for the standard MC approach than for the MLMC approach (cf. Fig. 22(a), 22(b)).

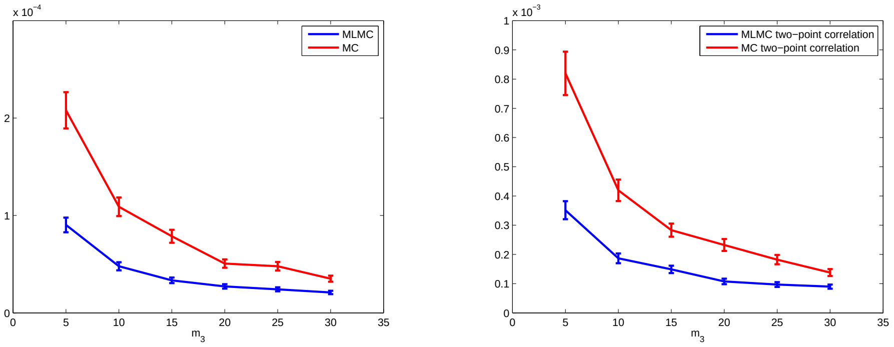
Independent samples In Section 10.2 we consider for only two levels the question, whether the realizations in each term of the MLMC approach should be independent or not. For two levels it is more accurate with the same amount of costs to choose the same realizations (cf. Section 10.2). In this paragraph we show numerical results for more than two levels. As previously we compute m_l realizations of the homogenized coefficient at level l with $m_l \leq m_{l-1}$, i.e., we compute $K^*_l(\omega_j, \omega'_j)$ for $1 \leq j \leq m_l$ with RVE $[a_l, b_l]$. Then we define as in Section 10.2 the MLMC approach with reusing the samples (cf. (10.5))

$$E^L_{same}(K^*_L) := \sum_{l=1}^{L} \frac{1}{m_l} \sum_{j=1}^{m_l} K^*_l(\omega_j, \omega'_j) - K^*_{l-1}(\omega_j, \omega'_j)$$

and with independent samples (cf. (10.6))

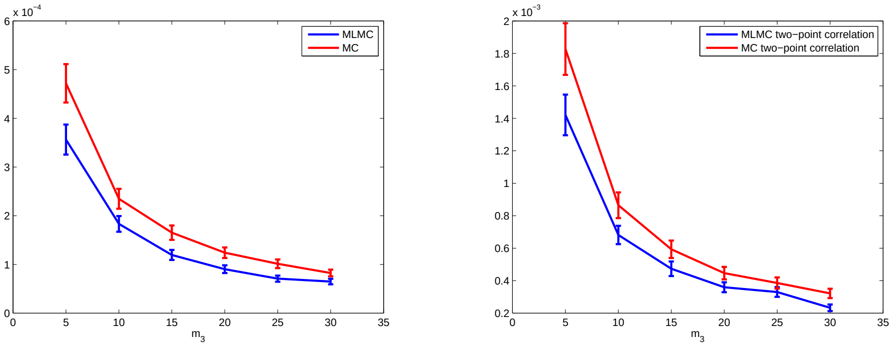
$$E^L_{ind}(K^*_L) := \sum_{l=1}^{L} \frac{1}{\widetilde{m}_l} \sum_{j=m_{l+1}+1}^{m_l} K^*_l(\omega_j, \omega'_j) - K^*_{l-1}(\omega_j, \omega'_j)$$

with $\widetilde{m}_l = m_l - m_{l+1}$ and $m_{L+1} = 0$. An important assumption for the MLMC approach is that the number of realizations decreases as the level l increases. Therefore we assume $\widetilde{m}_l \geq \widetilde{m}_{l+1}$. These two approximations share the same computational costs. In the following we compare



(a) Expected value of the effective coefficients, $\mathfrak{m} = (16m_3, 4m_3, m_3)$.

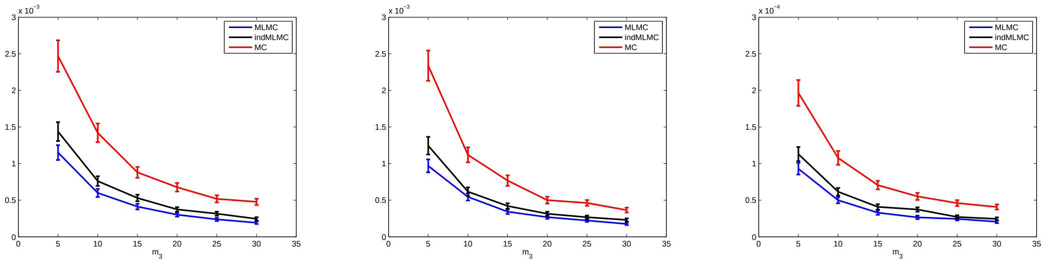
(b) Two-point correlation of the effective coefficients, $\mathfrak{m} = (16m_3, 4m_3, m_3)$.



(c) Expected value of the effective coefficients, $\mathfrak{m} = (4m_3, 2m_3, m_3)$.

(d) Two-point correlation of the effective coefficients, $\mathfrak{m} = (4m_3, 2m_3, m_3)$.

Figure 22: Relative mean square errors with equated costs. Example 3 non separable coefficient.



(a) Separable coefficient (b) Stationary coefficient (c) Non separable coefficient

Figure 23: Relative mean square errors for MLMC with independent samples, reused samples and the standard MC approach with $L = 3$, $Nb = 1000$ and $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$.

	separable	stationary	non separable
$\frac{Er_{MC}}{Er^{same}_{MLMC}}$	2.3	2.1	2.1
$\frac{Er_{MC}}{Er^{ind}_{MLMC}}$	1.8	1.7	1.7
$\frac{Er^{ind}_{MLMC}}{Er^{same}_{MLMC}}$	1.3	1.2	1.2

Table 5: Ratios between the relative mean square errors for MLMC with independent samples, reused samples and the standard MC approach with $L = 3$, $Nb = 1000$ and $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$.

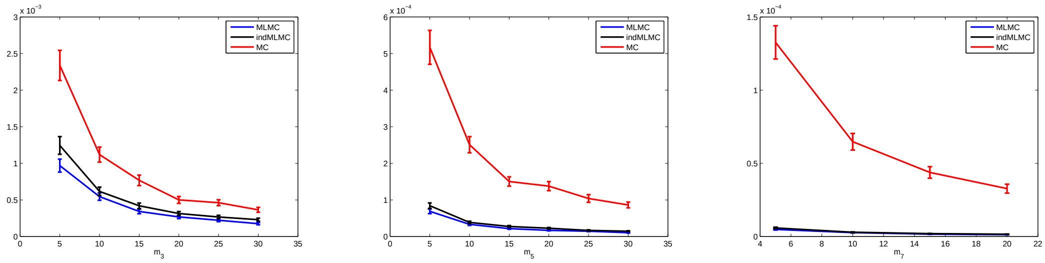
the accuracies of $E^L_{same}(K^*_L)$, $E^L_{ind}(K^*_L)$ and $E_{\widehat{m}}(K^*_L)$ by equated costs. We consider the same coefficients as in the previous paragraphs. If not otherwise stated we use the same parameters (e.g., a_l, b_l, Nb). To ensure $\widetilde{m}_l \geq \widetilde{m}_{l+1}$ we choose in all examples $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$. In Figure 23 the considered relative mean square errors for $L = 3$ for the different examples are shown. Again we are interested in the quotient of the different errors, for example between the relative mean square error of the MC approach and the one of the MLMC approach with reusing the samples

$$\frac{\frac{1}{Nb}\sum_{j=1}^{Nb}[e^{rel}_{MC}(K^*_L(\omega_j, \omega'_j))]^2}{\frac{1}{Nb}\sum_{j=1}^{Nb}[e^{rel}_{MLMC}(K^*_L(\omega_j, \omega'_j))]^2}.$$

We denote these errors with Er_{MC}, Er^{same}_{MLMC} and Er^{ind}_{MLMC}. So we are interested in the ratios $\frac{Er_{MC}}{Er^{same}_{MLMC}}$, $\frac{Er_{MC}}{Er^{ind}_{MLMC}}$ and $\frac{Er^{ind}_{MLMC}}{Er^{same}_{MLMC}}$. These are summarized in Table 5. For all example coefficients the behavior is about the same, the MLMC approach with reused samples is always better than the one with independent samples. The error with independent samples is approximately 1.2 times larger. In Table 6 and Figure 24 are shown the results for the example with a stationary coefficient for different numbers of levels. Compared to the error of the standard MC approach the difference in the two MLMC ansatzes decreases, but the ratio between the MLMC approach with independent and reused samples stays fixed, the error for independent samples is about 1.2 times larger than the one with reused samples.

	$L = 3$	$L = 5$	$L = 7$
$\frac{Er_{MC}}{Er^{same}_{MLMC}}$	2.1	7.6	26.1
$\frac{Er_{MC}}{Er^{ind}_{MLMC}}$	1.7	6.1	21.9
$\frac{Er^{ind}_{MLMC}}{Er^{same}_{MLMC}}$	1.2	1.3	1.2

Table 6: Ratios between the relative mean square errors for MLMC with independent samples, reused samples and the standard MC approach for the stationary example for different levels with $Nb = 1000$ and $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$.



(a) $L = 3$

(b) $L = 5$

(c) $L = 7$

Figure 24: Relative mean square errors for MLMC with independent samples, reused samples and the standard MC approach for the stationary example for different levels with $Nb = 1000$ and $\mathfrak{m} = (4^{L-l}m_L, \cdots, 4m_L, m_L)$.

14.1.3. Two dimensional example

In this section we deal with a two-dimensional microscale problem. To calculate the homogenized coefficient we have to solve two cell problems in each coarse grid block in general, i.e.,

$$\operatorname{div}(K_\epsilon(x,\omega)\nabla\chi_i) = 0, \text{ in } Y_\eta^x,\ i=1,2$$

with a representative volume centered at a macroscopic point x with size η, $Y_\eta^x = [x-\eta/2, x+\eta/2]^2$ subject to some boundary conditions. As a first step we consider a coefficient when there is a separation in the randomness at the macroscopic and the microscopic level and no dependency on the macroscopic space variable, more particular we study

$$K_\epsilon(x,\omega) = K(\frac{x}{\epsilon},\omega,\omega') := A(\omega)B(\frac{x}{\epsilon},\omega').$$

In this case we do not have an analytical expression for the homogenized coefficient like the arithmetic mean in the one-dimensional examples. Therefore we solve the cell problems with the cell-centered finite volume scheme (cf. Section 7) and the parameters as in Table 4. In the second two-dimensional problem we consider the more difficult case, when there is no separation between uncertainties at the macro- and microscopic levels. However, we study a special case where we can separate the space dimensions, i.e., we have ($x=(x_1,x_2)$)

$$K(x,\frac{x}{\epsilon},\omega,\omega') := K_1(x_1,\frac{x_1}{\epsilon},\omega,\omega')K_2(x_2,\frac{x_2}{\epsilon},\omega,\omega').$$

Here the local problems reduce to one-dimensional problems and we get an analytical expression for the homogenized coefficient. The presented results below refer to the first entry K_{11}^* of the homogenized coefficient matrix $(K_{ij}^*)_{1\le i,j\le 2}$.

Separable coefficient For the two-dimensional case, we first study coefficients where there is a separation in the randomness at the macro-level and micro-level, i.e.,

$$K(\frac{x}{\epsilon},\omega,\omega') = A(\omega)B(\frac{x}{\epsilon},\omega')$$

where A and B are both scalar valued. B is a random field with expected value $E[B]=10$ and the Gaussian covariance function

$$\operatorname{cov}(x,x') = Cov\left[B(\frac{x}{\epsilon},\cdot),B(\frac{x'}{\epsilon},\cdot)\right] = \sigma^2\exp\left(-\frac{|x-x'|^2}{\epsilon^2\tau_0^2}\right)$$

with standard deviation $\sigma=\sqrt{2}$ and $\tau=\epsilon\tau_0=0.04$. We generate samples of B with the Karhunen-Loève expansion. We take $A(\omega)=\exp(\omega)$, where ω is Gaussian distributed according to the law $N(0,1)$. The effective coefficient matrix is $K^*(\omega)=A(\omega)B^*$. We define levels l to approximate the expectation of B, i.e., we solve local problems, $1\le l\le L$,

$$\operatorname{div}(B(\frac{x}{\epsilon},\omega')\nabla\chi_i) = 0, \text{ in } [0,\eta_l]^2,\ i=1,2$$

to get a homogenized coefficient $B_l^*(\omega')$ at level l. Thus, at each level, we have $K_l^*(\omega,\omega') = A(\omega)B_l^*(\omega')$. For m_l realizations at level l, $1\le l\le L$, we define

$$E_{m_l}(K_l^*)(\tilde{\omega}) := \frac{1}{m_l}\sum_{j=1}^{m_l}K_l^*(\omega_j,\omega_j') = \frac{1}{m_l}\sum_{j=1}^{m_l}A(\omega_j)B_l^*(\omega_j')$$

with $\tilde{\omega} = (\omega_1, \cdots, \omega_{m_l})$. Note that we keep the dependence on the microscopic randomness ω', implicitly in this notation. We use the MLMC approach to approximate the expected value, as in (10.1) we have

$$E^L(K_L^*)(\tilde{\omega}) = \sum_{l=1}^{L} E_{m_l}(K_l^* - K_{l-1}^*)(\tilde{\omega}) = \sum_{l=1}^{L} \frac{1}{m_l} \sum_{j=1}^{m_l} A(\omega_j)(B_l^*(\omega_j') - B_{l-1}^*(\omega_j')).$$

We are interested in the errors

$$\begin{aligned} e_{MLMC}(K_L^*) &= \sqrt{E\left[(E[K_L^*] - E^L(K_L^*))^2\right]} \\ e_{MC}(K_L^*) &= \sqrt{E\left[(E[K_L^*] - E_{\widehat{m}}(K_L^*))^2\right]}. \end{aligned}$$

The theoretical reference value is $E[K_L^*]$, which we cannot access in practice and since it is computationally expensive to solve local problems on RVE $[0, \eta_L]^2$ we cannot afford many realizations of $B_L^*(\omega_j')$. So the statistical error would be too large, if we approximate $E[K_L^*]$ with the empirical mean at level L. That is why we introduce a biased estimator K_{ref}^* ($E[K_{ref}^*] \neq E[K_L^*]$), namely

$$\begin{aligned} E[K_L^*] &\approx K_{ref}^* \\ &:= \frac{1}{L} \sum_{l=1}^{L} E_n(E_{m_l^{ref}}(K_l^*)(\tilde{\omega})) \\ &:= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{n} \sum_{i=1}^{n} E_{m_l^{ref}}(K_l^*)(\tilde{\omega}^i) \\ &= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_l^{ref}} \sum_{j=1}^{m_l^{ref}} K_l^*(\omega_j^i, \omega_j') \\ &= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_l^{ref}} \sum_{j=1}^{m_l^{ref}} A(\omega_j^i) B_l^*(\omega_j') \end{aligned}$$

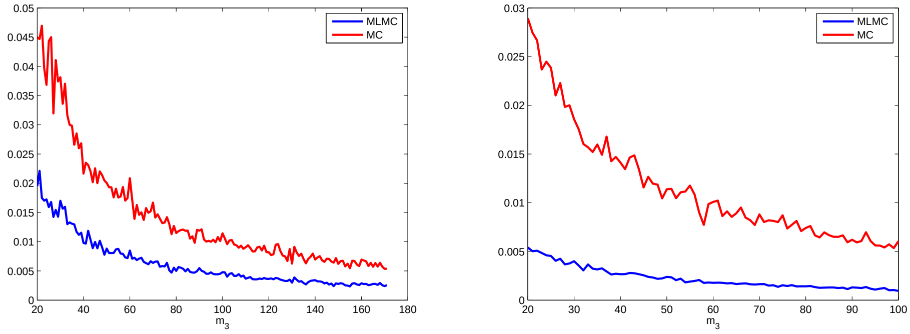
with $\tilde{\omega}^i = (\omega_1^i, \cdots, \omega_{m_1^{ref}}^i)$, $m_l^{ref} \leq m_{l-1}^{ref}$, where we have taken into account all the realizations on the RVEs $[0, \eta_l]^2$. To approximate the remaining expectation we use the randomness of the coarse scale, i.e.,

$$\begin{aligned} e_{MLMC}(K_L^*) &\approx \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left[K_{ref}^* - E^L(K_L^*)(\tilde{\omega}_i)\right]^2} \\ e_{MC}(K_L^*) &\approx \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left[K_{ref}^* - E_{\widehat{m}}(K_L^*)(\tilde{\omega}_i)\right]^2}. \end{aligned}$$

As mentioned above we equate the computational work of the MLMC and the MC approach and compare the errors. For the MC approach the work is $\hat{m}\left(\frac{\eta_L}{\epsilon}\right)^2$ and for the MLMC approach we have $\sum_{l=1}^{L} m_l \left(\frac{\eta_l}{\epsilon}\right)^2$. Therefore we get for the MC approach with equated costs

$$\hat{m} = \frac{\sum_{l=1}^{L} m_l \left(\frac{\eta_l}{\epsilon}\right)^2}{\left(\frac{\eta_L}{\epsilon}\right)^2}.$$

We work with three different levels, i.e $L = 3$ and we denote the vector of the number of realizations at each level with $\mathfrak{m}$, i.e., $\mathfrak{m} = (m_1, m_2, m_3)$. In our example we have $n = 500$ and $\mathfrak{m}^{ref} = (2000,\ 1000,\ 300)$. In Figure 25(a) we show the errors $e^2_{MC}(K^*_L)$ and $e^2_{MLMC}(K^*_L)$ on the first entry of the homogenized coefficient matrix for $\mathfrak{m} = (4\,m_3, 2\,m_3,\ m_3)$. In this case we have chosen the number of realizations per level m_l according to the numerical estimated convergence rate β (cf. Section 14.1.1). From these simulations we observe that MLMC provides smaller errors compared to standard MC for the same amount of work. In the mean the MC square error is approximately 2.3 times larger than the MLMC square error. In Figure 25(b) we show the errors $e^2_{MC}(K^*_L)$ and $e^2_{MLMC}(K^*_L)$ where we use the theoretical rate β^{theo} to choose $\mathfrak{m}$. Here the MC square error is approximately 5.2 times larger than the MLMC square error.

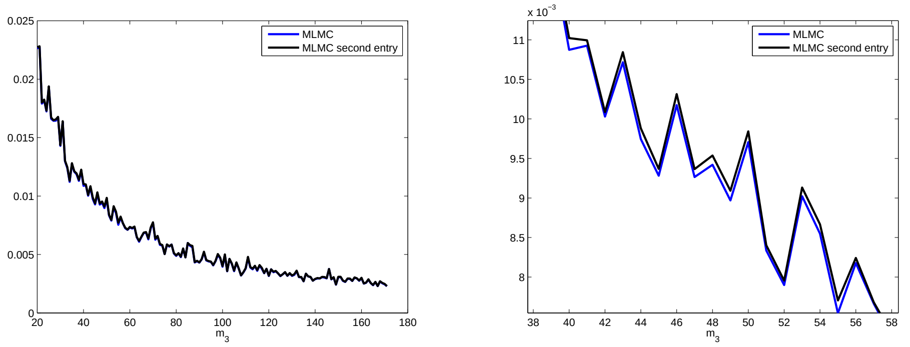


(a) $\mathfrak{m} = (4\,m_3, 2\,m_3,\ m_3)$

(b) $\mathfrak{m} = (16\,m_3, 4\,m_3,\ m_3)$

Figure 25: Errors $e^2_{MC}(K^*_L)$ and $e^2_{MLMC}(K^*_L)$. Separable coefficient.

In Figure 26 we compare the MLMC results for the first entry $[K^*_l]_{11}$ of the effective coefficient and the second one $[K^*_l]_{22}$. The behavior is the same and only if we consider only a small range of M_3 one can see a slight difference (cf. Fig. 26(b)). That is why we consider the first entry only in the other examples.



(a) $\mathfrak{m} = (4\,m_3, 2\,m_3,\ m_3)$

(b) $\mathfrak{m} = (4\,m_3, 2\,m_3,\ m_3)$

Figure 26: Errors between the first entry $[K^*_l]_{11}$ of the effective coefficient for MLMC and the second one $[K^*_l]_{22}$. Separable coefficient.

Non separable coefficient Next, we consider the case when there is no separation between macro- and micro-level uncertainties. In general it is very expensive to have a sufficiently large number of samples to reduce the statistical noise. Since in the limit of infinitely large RVEs the effective coefficient does not depend on the choice of the boundary conditions of the local problems (cf. ([16])), we consider a special case where we can solve the local problems and thus the effective coefficient analytically. Therefore we look at local problems with Dirichlet and no-flow boundary conditions, i.e.,

$$\begin{aligned} -\mathrm{div}\left(K(x,\frac{x}{\epsilon},\omega,\omega')\nabla\chi_i\right) &= 0 \text{ in } Y_{\eta_l} \\ \chi_i &= x_i \text{ at } \partial Y_{\eta_l}^D \\ n\cdot\nabla\chi_i &= 0 \text{ at } \partial Y_{\eta_l}\setminus\partial Y_{\eta_l}^D \end{aligned}$$

with $\partial Y_{\eta_l}^D = \{x \in \partial Y_{\eta_l} | x_i = 0 \text{ or } x_i = \eta_l\}$. Furthermore we assume separation of the dimensions for the coefficient, i.e.,

$$K(x,\frac{x}{\epsilon},\omega,\omega') = K_1(x_1,\frac{x_1}{\epsilon},\omega,\omega')K_2(x_2,\frac{x_2}{\epsilon},\omega,\omega'). \tag{14.1}$$

Then the local problem reduces to a one-dimensional problem in the direction x_i for the function χ_i that only depends on x_i. Analogous to the one-dimensional case the solution reads,

$$\chi_i(x_i,\omega,\omega') = \left(\frac{1}{\eta_l}\int_0^{\eta_l} K_i^{-1}(x_i,\frac{\bar{x}_i}{\epsilon},\omega,\omega')\,d\bar{x}_i\right)^{-1}\frac{1}{\eta_l}\int_0^{x_i} K_i^{-1}(x_i,\frac{\bar{x}_i}{\epsilon},\omega,\omega')\,d\bar{x}_i.$$

We get for the first entry of the upscaled coefficient

$$(K_{\eta_l}^*)_{11}(x,\omega,\omega') = \left(\frac{1}{\eta_l}\int_0^{\eta_l} K_1^{-1}(x_1,\frac{\bar{x}_1}{\epsilon},\omega,\omega')\,d\bar{x}_1\right)^{-1}\frac{1}{\eta_l}\int_0^{\eta_l} K_2(x_2,\frac{\bar{x}_2}{\epsilon},\omega,\omega')\,d\bar{x}_2.$$

In our example we choose

$$\begin{aligned} K_1^{-1}(\frac{x_1}{\epsilon},\omega,\omega') &= C(1+\omega)+\exp\left(\omega\omega'\sin\left(\frac{x_1}{\epsilon}\right)\right)\cos\left(\frac{x_1}{\epsilon}\right) \\ K_2(x_2,\frac{x_2}{\epsilon},\omega,\omega') &= C(1+\exp(5\omega))x_2+(1+x_2)\exp\left((1+x_2)\omega\omega'\sin\left(\frac{x_2}{\epsilon}\right)\right)\cos\left(\frac{x_2}{\epsilon}\right) \end{aligned}$$

where ω and ω' are i.i.d. random variables distributed in $[0.5,1]$ and $C = 2e$.
As in the example with the separable coefficient we consider the RVEs $[0,\eta_l] = [0,0.5^{L+1-l}]$ and $\epsilon = \frac{0.5^L}{10}$. To choose the number of realizations m_l in the MLMC approach for each level appropriately we estimate the convergence rate β numerically. The reference is

$$K^*(x,\omega) = \lim_{\eta\to\infty} K_\eta^*(x,\omega,\omega') = (1+\exp(5\omega))/(1+\omega)x_2.$$

Again we use 15 different levels and the rate β is approximately 2 (cf. Figure 27). As reference we use the MC approach with $m^{ref} = 400000$ realizations (ω,ω') on the largest RVE $[0,\eta_L]^2 = [0,0.5]^2$. Thanks to the special expression (14.1) of the coefficient we can compute a large number of samples in this two-dimensional example. For the MLMC approach we consider three different levels ($L = 3$) and according to β we choose $\mathfrak{m} = (16m_3, 4m_3, m_3)$ realizations. To determine a confidence interval we repeat it for $Nb = 2000$ different sets of realizations. In Figure 28 we compare the accuracies of the MC and MLMC approaches for the mean and the two-point correlation with equal costs, i.e., $\hat{m} = \frac{\sum_{l=1}^L m_l\left(\frac{\eta_l}{\epsilon}\right)^2}{\left(\frac{\eta_L}{\epsilon}\right)^2}$. Again, the MLMC approach is more accurate, we find that the relative mean square error for the MC approach is approximately 5 times larger than the one for the MLMC approach.

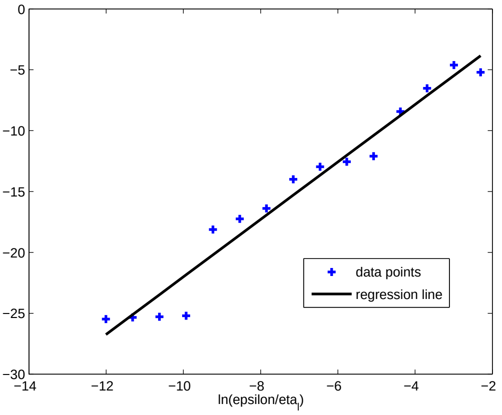
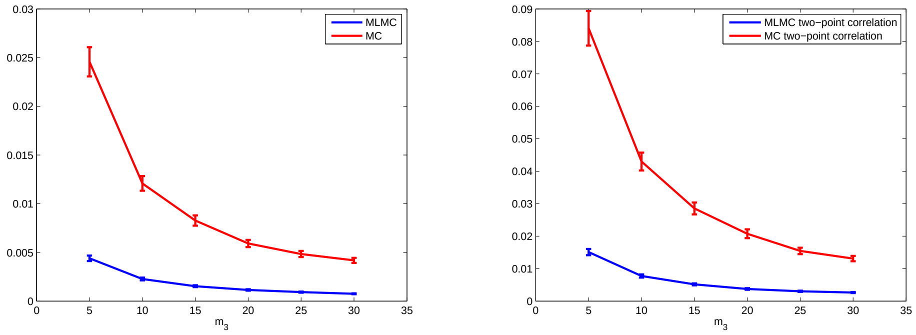


Figure 27: Computed data points with corresponding regression line with slope β. 15 levels, $\epsilon = \frac{0.5^L}{10}$, mean over 50000 samples, $\beta = 2.3600$ and $\ln C = 1.5888$.



(a) Relative mean square errors of the expected value of the effective coefficients.

(b) Relative mean square errors of the two-point correlation of the effective coefficients.

Figure 28: Relative mean square errors with equated costs and $\mathfrak{m} = (16m_3, 4m_3, m_3)$ for the two-dimensional case.

14.2. Numerical results of the homogenized solution

Next we present results for the coarse solution. This is an application of the weighted MLMC approach (cf. Section 12). Again we start with an one-dimensional example, after that we consider two-dimensional problems.

14.2.1. One dimensional example

As before we start with the one dimensional case where we know the reference solution exactly. We consider the coarse problem

$$\frac{d}{dx}\left(K^*(x,\omega,\omega')\frac{d}{dx}u\right) = f(x) \text{ in } [0,1]$$

with zero Neumann boundary conditions, $u(0) = 0$ and $\int_0^1 f = 0$. For each realization the analytical solution is

$$u(x,\omega,\omega') = \int_0^x (K^*(y,\omega,\omega'))^{-1} \int_0^y f(z)\,dz\,dy.$$

To use an MLMC approach we need an approximation of the coarse solution on different coarse grids. Therefore we denote the vertices of the grid with x_i^H, $0 \leq i \leq N$, i.e., the mesh size is $H = x_i - x_{i-1} = \frac{1}{N}$ and $g_i^H = g(x_i^H)$ for any function g. We approximate the solution with

$$\begin{aligned} u_i^H &= u_{i-1}^H + (K^*(x_i^H))^{-1} \int_{x_{i-1}^H}^{x_i^H} \int_0^x f(y)\,dy\,dx \\ &= \sum_{j=1}^{i} (K^*(x_j^H))^{-1} \int_{x_{j-1}^H}^{x_j^H} \int_0^x f(y)\,dy\,dx. \end{aligned}$$

Therefor $u_i^{H_l}$ is an approximation of $u_{\infty,H_l}(x_i)$, i.e., there is no error due to the homogenization of the coefficient K. If we use $K^*_{\eta_k}$ instead of K^* in the above formula we have an approximation of $u_{\eta_k,H_l}(x_i)$. For $H_l = cH_{l+1}$, $c \in \mathbb{N}$, $1 \leq l \leq L$, it is true

$$\{x_i^{H_l} | 1 \leq i \leq \frac{1}{H_l}\} \subset \{x_i^{H_{l+1}} | 1 \leq i \leq \frac{1}{H_{l+1}}\}.$$

So we are in the setting described in Section 13.3 and we can apply the weighted MLMC approach. We consider the following example

$$\begin{aligned} f(x) &= e^x - e + 1 \\ (K^*(x,\omega,\omega'))^{-1} &= C(1+\exp(5\omega))x \\ &\quad + \frac{1}{b-a}\left[\frac{\epsilon}{\omega\omega'}\left(\exp\left((1+x)\omega\omega'\sin\left(\frac{b}{\epsilon}\right)\right) - \exp\left((1+x)\omega\omega'\sin\left(\frac{a}{\epsilon}\right)\right)\right)\right] \end{aligned}$$

where ω and ω' are random variables, uniformly distributed in $[0.5,1]$ and $C = 2e$. As before we use the different RVEs $[a_l, b_l] = [0, 0.5^{L+1-l}]$. As reference we use the mean of the analytical solution for $b = \infty$

$$E[u] = C(1+\exp(5\cdot 0.75))(\exp(x)x - \exp(x) - (e-1)\frac{x^3}{3} - 0.5x^2 + 1).$$

We consider three different levels and choose all weights equal to 1 and $\epsilon = \frac{b_1}{100}$. We equate the costs of solving the coarse grid problem for a fixed ratio of realizations per level $\mathfrak{M} = (16M_3, 4M_3, M_3)$ and the grid sizes $\mathfrak{H} = (\frac{1}{4}, \frac{1}{8}, \frac{1}{16})$. As in the example for the coefficient we repeat the calculation for $Nb = 20000$ different sets of random numbers and determine the mean and the confidence intervals. We consider the relative mean square errors for the L^2-norm, i.e., $\|E[u] - E^{L*}(\tilde{u})\|^2 = E\left[\|E[u] - E^{L*}(\tilde{u})\|^2_{L^2}\right]$. In Figure 29 you find a larger relative weighted MC L^2-error compared to MLMC, namely the MC error is about 1.2 times larger.

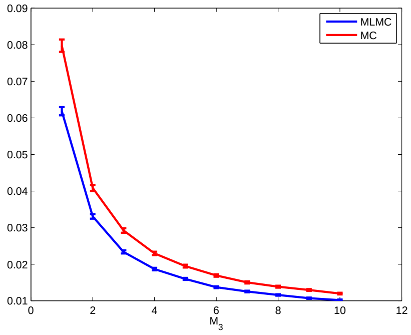


Figure 29: Relative mean square L^2-errors of the solution for 10 different realizations, the weight $(1,1,1)$, $\mathfrak{M} = (16M_3, 4M_3, M_3)$ and $\mathfrak{H} = (\frac{1}{4}, \frac{1}{8}, \frac{1}{16})$.

14.2.2. Two dimensional example

We consider

$$-\text{div}\left[K(x, \omega, \frac{x}{\epsilon}, \omega')\nabla u\right] = f \text{ in } D = (0,1)^2$$

with homogeneous Dirichlet boundary conditions and source term

$$f(x) = f(x_1, x_2) = 100(x_1 + x_2).$$

First we assume separated scales of the coefficient of the microscale problem (cf. Section 13.2). After that we consider a more general case where we apply weighted MLMC in two dissensions.

Separable case In this section we present numerical results for

$$K(x, \frac{x}{\epsilon}, \omega, \omega') = A(x, \omega)B(\frac{x}{\epsilon}, \omega')$$

where B is a log-normal distributed random field $B = e^K$ with K having zero mean and covariance function $\text{cov}(x, x') = \sigma^2 \exp(-\frac{|x-x'|^2}{\tau^2})$ with $\sigma = \sqrt{2}$, $\tau = 0.04$ For A we choose

$$A(x, \omega) = 2 + |\omega_1 \sin(2\pi x_1)| + |\omega_2 \sin(2\pi x_2)| + |\omega_3 \sin(\pi x_1)|$$

with independently normal distributed ω_k. Because of separability, it is sufficient to calculate the homogenized coefficient of B in only one arbitrary macroscopic point. For MLMC for each level $1 \leq l \leq L$ we solve the coarse problem

$$-\text{div}\left(A_l^k(x)E_{m_l}(B_l^*)\nabla u_l^k\right) = f \text{ in } D,\ 1 \leq k \leq M_l \qquad (14.2)$$

with grid size H_l and for MC

$$-\text{div}\left(A_L^k(x)E_{\hat{m}}(B_L^*)\nabla u_L^k\right) = f \text{ in } D,\, 1 \leq k \leq \hat{M}$$

with grid size H_L. As reference we solve (14.2) with $m_l = m^{ref} = 50$ and $M_l = M^{ref} = 1000$ for all levels and calculate the mean over both the levels and the number of realizations, i.e,

$$E^i_{M^{ref},L} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{M^{ref}}\sum_{k=1}^{M^{ref}} u_l^k.$$

As before we have three different levels and we equate the computational costs for the coefficient as well as solving the coarse scale problems, i.e., $\hat{m} = \frac{m_1\eta_1^2+m_2\eta_2^2+m_3\eta_3^2}{\eta_3^2}$ and $\hat{M} = \frac{M_3H_3^2+M_2H_2^2+M_1H_1^2}{H_3^2}$ respectively. We denote by $\mathfrak{M} = (M_1, M_2, M_3)$ and $\mathfrak{m} = (m_1, m_2, m_3)$. In

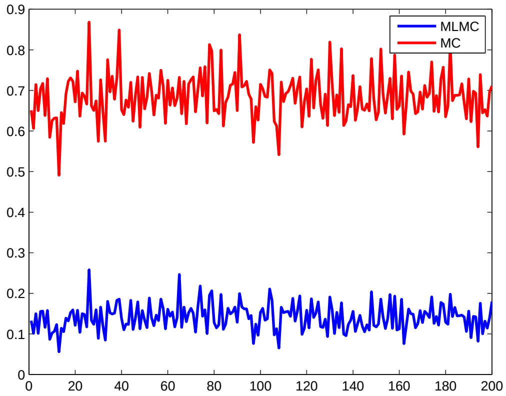


Figure 30: Relative L^2-errors e^i_{MC} and e^i_{MLMC} of the solution for 200 different realizations, $\mathfrak{M} = (32, 32, 16)$, $\mathfrak{m} = (50, 40, 20)$.

Figure 30, we compare the relative L^2-errors for MLMC and MC with $\mathfrak{M} = (32, 32, 16)$ and $\mathfrak{m} = (50, 40, 20)$ for 200 independent sets of realization for the coarse problem, i.e., we consider

$$e^i_{MLMC}(u_L) = \frac{\|E^i_{M^{ref},L}(u_L) - E^{i,L}(u_L)\|_{L^2(D)}}{\|E_{M^{ref},L}(u_L)\|_{L^2(D)}}$$
$$e^i_{MC}(u_L) = \frac{\|E^i_{M^{ref},L}(u_L) - E^i_{\hat{M}}(u_L)\|_{L^2(D)}}{\|E_{M^{ref},L}(u_L)\|_{L^2(D)}}$$

with $1 \leq i \leq 200$. Note that $\mathfrak{M}$ is chosen based on the calculations presented in [10] (we have also verified these calculation for finite volume methods) and we do not change the number of realizations $\mathfrak{M}$ and $\mathfrak{m}$ along the x-axes, here we use different random numbers. Therefore the error does not decrease like in the other figures. On average we gain almost a factor of 5

$$\begin{aligned} E_{200}(e_{MLMC}) &= 0.1411 \\ E_{200}(e_{MC}) &= 0.6851 \end{aligned}$$

and also the standard deviation for the MLMC approach is much smaller

$$\begin{aligned} std_{MLMC} &= 0.0324 \\ std_{MC} &= 0.0565. \end{aligned}$$

Note that the standard deviation of $e^i_{MLMC}(u_L)$ and $e^i_{MC}(u_L)$ coincides with the root mean square errors considered in the theoretical part (cf. Section 13.2). The standard variation for the MC approach is 1.7 times larger than the one for MLMC. If we consider the mean square errors as in the previous numerical examples, we gain a factor of 3.

Non separable case In this section, we consider a non-separable case when uncertainties at macro-scale and micro-scale are not separable. As we mentioned in this case we use a weighted MLMC approach. For comparison we will also use the standard MLMC approach using the elements in the diagonal of Table 3 where many upscaled computations are not taken into account (see discussions in Section 13.3). We consider

$$K(x, \frac{x}{\epsilon}, \omega, \omega') = \exp\Big(A(x,\omega)B(\frac{x}{\epsilon}, \omega')\Big),$$

where B is a random field with expected value $E[B] = 0$ and the Gaussian covariance function $\mathrm{cov}(x, x') = \sigma^2 \exp(-\frac{|x-x'|^2}{\tau^2})$, with $\sigma = \sqrt{2}$ and $\tau = 0.04$. We assume $A(x,\omega)$ is an independent (in space) discrete random variable that takes the values $A_i = 1 + 0.1i$ for $0 \le i \le 5$ uniformly. For each A_i we calculate the upscaled coefficient denoted by $K^*_i(\omega')$. As for the spatial correlation in macroscale, we choose

$$K^*(x,\omega,\omega') = K^*_i(\omega') \text{ if } i < \omega \frac{x_1 + x_2}{2} \le i + 1$$

with ω i.i.d. between 0 and 5. Since we do not know the exact solution of the coarse problem we use

$$E[u^*] \approx \frac{1}{L} \sum_{l=1}^{L} E_{M_{ref}}(u_{\eta_l, H_l})$$

as reference with $M_{ref} = 1000$. We consider the weighted MLMC and the MLMC approach. In both cases we use $\mathfrak{M} = (200, 100, 50)$. That guarantees the same costs for solving the coarse grid problems for the weighted MLMC and the MLMC approach. Note that the total costs for the MLMC approach are higher than for weighted MLMC, since we compute samples of the homogenized coefficients for M_l and $M_l - M_{l-1}$, respectively. For the MC approach we equate the costs for solving the coarse grid problems, i.e., $\hat{M} = \sum_{l=1}^{L} \frac{M_l H_l^{-2}}{H_L^{-2}}$.

For weighted MLMC we use the weights $(1, \frac{6}{7}, \frac{4}{7})$. This choice guarantees the same order of the systematic error for MC and MLMC and the constant in front of the exact solution is one, i.e., we have

$$\begin{aligned} \text{MLMC: } \|Cu^* - \tilde{u}\| &= O(H_L + \delta_L) \\ \text{MC: } \|u^* - u_{\eta_L, H_L}\| &= O(H_L + \delta_L) \end{aligned}$$

with $C = 1$. Our numerical results yield the following relative mean square L^2 errors of 200 samples

MLMC: 0.0050
weighted MLMC: 0.0016
MC: 0.0023

In Figure 31, L^2-errors for MLMC, weighted MLMC and MC are plotted. Note we do not change the number of realizations $\mathfrak{M}$ along the x-axes, we use here different random numbers. Therefor the error does not decrease as in the other figures. We see from this figure that weighted MLMC is more accurate.

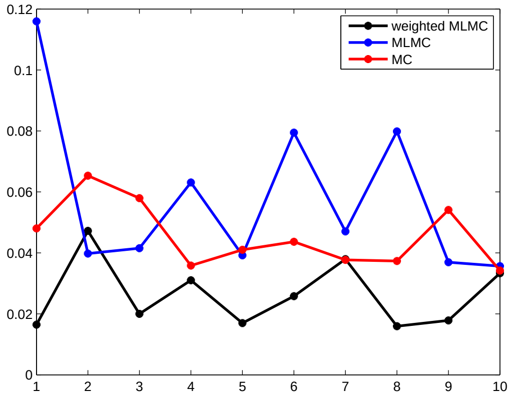
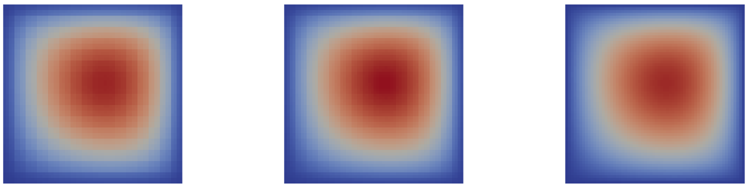


Figure 31: L^2-error for weighted MLMC, MLMC and MC with $\mathfrak{M} = (200, 100, 50)$ and the weights $(1, \frac{6}{7}, \frac{4}{7})$.

The reference solution, the weighted MLMC and the MC solution are plotted in Figure 32.



(a) reference solution (b) weighted MLMC (c) MC

Figure 32: Plots of the reference solution and the solutions calculated with weighted MLMC and MC.

Part III.
Using multi-level Monte Carlo with two-phase flow and transport

In this part we consider two-phase flow and transport in heterogeneous subsurface formations. In those simulations one has to deal with a rich hierarchy of spatial spaces and uncertainties. Typically, one approximates the solution on a coarse grid. These approaches include upscaling or multiscale methods [20, 29]. The previous parts are addressed to homogenization. In this part we discuss multiscale methods where we construct multiscale basis functions for each coarse-grid block to solve the problem on a coarse grid inexpensively. Due to the uncertainties this might still be expensive, though; one has to solve for a large number of samples to make accurate predictions. Techniques including ensemble level upscaling [19] or multiscale methods [4] are proposed to reduce the number of simulations in Monte Carlo methods. In this part we combine a multi-level Monte Carlo method with two different ensemble level multiscale mixed finite element methods (MsFEM) - no-local-solve-online (NLSO) and local-solve-online (LSO) ensemble level MsFEM - to reduce the computational costs ([30]). In both methods, the LSO and the NLSO multiscale methods, one selects a number of samples and generates for each of these the corresponding multiscale basis functions. With these basis functions a multiscale space is constructed. An advantage of such an approach (cf. [4]) is that one does not need to generate a coarse space for each new selected realization. One simply projects the global solution onto the multiscale space. The accuracy of the coarse-grid simulations depends on the number of multiscale basis functions used to construct the multiscale space. In particular, the simulation is more accurate if more realizations are selected to generate multiscale basis functions.

Additionally, the accuracy is influenced by the choice of boundary conditions for the multiscale basis. We consider local boundary conditions and boundary conditions using limited global information. The global boundary conditions are more accurate, but computationally more expensive (cf. [5])

We apply multi-level Monte Carlo as follows. Each level is represented by a multiscale space generated with a different number of samples. We run more forward coarse-grid simulations using the smaller dimensional multiscale spaces and less simulations using the higher dimensional space. We show that we achieve a higher accuracy by combining these approaches than for standard MC for the same computational costs. The number of selected realizations per level in MLMC depends on the convergence with respect to the coarse space dimension. One can estimate this rate based on a small number of apriori computations.

The part is organized as follows. In Section 15.2 we introduce the considered model problem. Section 15.3 is addressed to the mixed multiscale finite element method, Section 16 to the two different ensemble level multiscale approaches (NLSO and LSO). In Section 17 we combine these approaches with multi-level Monte Carlo and in Section 18 we present numerical results.

15. Preliminaries

15.1. Physical quantities

In this part we consider two-phase flow and transport in porous media. For further understanding we introduce some physical quantities (cf. [39]). In an unsaturated media one distinguishes three different phases in general, the wetting phase, the non-wetting phase and a gas phase. To build different phases the fluids may not mix. Fluids with different physical properties (e.g., hydrophilic and hydrophobical) can build different phases. Since gases are always miscible there exists one gas phase only. We consider only saturated zones with two different phases, the wetting and non-wetting phase. In the following referred to as water (w) and oil (o) phase, respectively. With a porous media we denote a solid body with pores, which are at least partly connected. The rock properties, determined as volume fractions and distributions of the pores, are parameters of the multi-phase flow in a reservoir. The void volume fraction of the medium is the rock porosity, i.e.,

$$\phi := \frac{\text{void volume}}{\text{medium volume}}, \quad 0 \leq \phi \leq 1.$$

In a multi-phase setting the porosity ϕ_α of phase α ($\alpha = w$ or $\alpha = o$) denotes the ratio of the void volume filled with phase α and the total volume. It is assumed that all pores are filled with the different phases and the volume fraction occupied by each phase is the saturation of the corresponding phase, i.e.,

$$S_\alpha := \frac{\phi_\alpha}{\phi}, \quad 0 \leq S_\alpha \leq 1.$$

Therefore we have

$$\sum_\alpha S_\alpha = 1.$$

The permeability k indicates the ability of the medium to transmit a single fluid. In general the permeability in the different directions depends on the other directions, i.e., k is a tensor. If there is more than one phase, the permeability of each phase depends on the presence of the other fluids. Thus we introduce the so-called relative permeability $k_{r\alpha}(S_w)$, $0 \leq k_{r\alpha} \leq 1$, as a dimensionless in general nonlinear function of the saturation. Due to the tension at the interface of two different phases the phase pressures p_α are different, the capillary pressure is defined as this difference

$$p_c := p_o - p_w.$$

Usually it is assumed that the capillary pressure depends on the saturation only. With ρ_α and μ_α we denote the density and viscosity of phase α, respectively.

15.2. Two-phase flow and transport model

We consider the flow of two phases (cf. [39]), one phase water and one phase oil, denoted with the subscripts w and o, respectively. The flow takes place in a reservoir denoted with D. For each phase α the continuity equation reads

$$\frac{\partial \left(S_\alpha \phi \rho_\alpha\right)}{\partial t} + \operatorname{div}\left(\rho_\alpha v_\alpha\right) = \rho_\alpha q_\alpha \tag{15.1}$$

with the phase source q_α which models sources for $q_\alpha > 0$ and sinks for $q_\alpha < 0$ and the phase velocity v_α. To model the phase velocity a generalized Darcy law is used, namely

$$v_\alpha = -\frac{k_{r\alpha}(S_w)}{\mu_\alpha} k \left(\nabla p_\alpha - \rho_\alpha G\right), \tag{15.2}$$

with the gravitational pull-down force G depending on the gravitational constant g. In the following we rewrite the two continuity equations into a system consisting of a pressure and a saturation or fluid-transport equation. First we introduce the phase mobility

$$\lambda_\alpha(S_w) = \frac{k_{r\alpha}(S_w)}{\mu_\alpha}.$$

Pressure equation After expanding the derivatives in space and time and dividing by the phase densities, we get for (15.1)

$$\frac{\partial\phi}{\partial t}S_\alpha+\phi\frac{\partial S_\alpha}{\partial t}+\phi\frac{S_\alpha}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial t}+\text{div }v_\alpha+\frac{v_\alpha\cdot\nabla\rho_\alpha}{\rho_\alpha}=q_\alpha. \tag{15.3}$$

If we sum the equations (15.3) for each phase with

$$\begin{aligned} v &= v_w+v_o\\ q &= q_w+q_o \end{aligned}$$

we get

$$\begin{aligned} q &= \frac{\partial\phi}{\partial t}(S_w+S_o)+\phi\frac{\partial(S_w+S_o)}{\partial t}+\phi\frac{S_w}{\rho_w}\frac{\partial\rho_w}{\partial t}+\phi\frac{S_o}{\rho_o}\frac{\partial\rho_o}{\partial t}+\text{div }v+\frac{v_w\cdot\nabla\rho_w}{\rho_w}+\frac{v_o\cdot\nabla\rho_o}{\rho_o}\\ &= \frac{\partial\phi}{\partial t}+\phi\frac{S_w}{\rho_w}\frac{\partial\rho_w}{\partial t}+\phi\frac{S_o}{\rho_o}\frac{\partial\rho_o}{\partial t}+\text{div }v+\frac{v_w\cdot\nabla\rho_w}{\rho_w}+\frac{v_o\cdot\nabla\rho_o}{\rho_o}. \end{aligned}$$

Here we used $1=S_w+S_o$. For simplicity we assume that the rock and the fluid phases are incompressible, i.e all terms with derivatives (spatial and time) of the porosity ϕ and the phase densities ρ_α vanish. The above equation reduces to

$$\text{div }v=q$$

with

$$\begin{aligned} v &= v_w+v_o\\ &= -\left(\frac{k_{rw}(S_w)}{\mu_w}k\left(\nabla p_w-\rho_w G\right)+\frac{k_{ro}(S_w)}{\mu_o}k\left(\nabla p_o-\rho_o G\right)\right)\\ &= -\left(\lambda_w(S_w)k\left(\nabla p_w-\rho_w G\right)+\lambda_o(S_w)k\left(\nabla p_o-\rho_o G\right)\right). \end{aligned}$$

With the capillary pressure we can write $p_w=p_o-p_c(S_w)$. It follows

$$\begin{aligned} v &= -\left(\lambda_w(S_w)k\left[\nabla(p_o-p_c(S_w))-\rho_w G\right]+\lambda_o(S_w)k\left[\nabla p_o-\rho_o G\right]\right)\\ &= -\left[\lambda_w(S_w)+\lambda_o(S_w)\right]k\nabla p_o+k\left[\lambda_w(S_w)\rho_w+\lambda_o(S_w)\rho_o\right]G+\lambda_w(S_w)k\nabla p_c(S_w)\\ &= -\lambda(S_w)k\nabla p_o+k\left[\lambda_w(S_w)\rho_w+\lambda_o(S_w)\rho_o\right]G+\lambda_w(S_w)k\nabla p_c(S_w) \end{aligned}$$

with the total mobility $\lambda(S_w)=\lambda_w(S_w)+\lambda_o(S_w)$. So we end with the following elliptic equation for the pressure

$$-\text{div}\left(\lambda(S_w)k\nabla p_o-k\left[\lambda_w(S_w)\rho_w+\lambda_o(S_w)\rho_o\right]G-\lambda_w(S_w)k\nabla p_c(S_w)\right)=q \tag{15.4}$$

with some boundary conditions. It is common to impose no-flow boundary conditions.

Saturation equation To complete the model we derive equations for the phase saturations S_w and S_o with the help of the continuity equation (15.1) of each phase. Since we have $1=S_w+S_o$ we need only one saturation equation. We pick S_w as second unknown. The continuity equation for the water phase gives

$$\begin{aligned}
0 &= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w v_w\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o}\left(1 + \frac{\lambda_o}{\lambda_w}\right) v_w\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o}\left(v_o + v_w - v_o + \frac{\lambda_o}{\lambda_w} v_w\right)\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o}\left(v - v_0 + \frac{\lambda_o}{\lambda_w} v_w\right)\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o}\left(v + \lambda_o k\left(\nabla p_o - \rho_o G\right) - \frac{\lambda_o}{\lambda_w} \lambda_w k\left(\nabla p_w - \rho_w G\right)\right)\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o}\left(v + \lambda_o k\left(\nabla(p_o - p_w) - \rho_o G + \rho_w G\right)\right)\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w \frac{\lambda_w}{\lambda_w + \lambda_o} v\right) + \operatorname{div}\left(\rho_w \frac{\lambda_o \lambda_w}{\lambda_w + \lambda_o} k\left(\nabla p_c - \rho_w G + \rho_o G\right)\right) - \rho_w q_w \\
&= \frac{\partial\left(S_w \phi \rho_w\right)}{\partial t} + \operatorname{div}\left(\rho_w f v\right) + \operatorname{div}\left(\rho_w \lambda_o f k\left(\nabla p_c - \rho_w G + \rho_o G\right)\right) - \rho_w q_w
\end{aligned}$$

with

$$f(S_w) = \frac{\lambda_w(S_w)}{\lambda(S_w)}.$$

As before we assume incompressibility, this leads to

$$\phi \frac{\partial S_w}{\partial t} + \operatorname{div}\left(f v\right) + \operatorname{div}\left(\lambda_o f k \nabla p_c\right) + \left(\rho_o G - \rho_w G\right) \operatorname{div}\left(\lambda_o f k\right) = q_w. \tag{15.5}$$

To have a complete description of the model boundary, e.g., no-flow, and initial conditions have to be imposed. This saturation equation (15.5) is a parabolic equation in general, but on a reservoir scale the effects of the capillary pressure are usually dominated by the viscous and gravity forces, represented by fv and Gv, respectively. Then the saturation equation becomes hyperbolic.

Two-phase flow and transport equation In the following we assume that the displacement is dominated by viscous effects, i.e., we neglect the effects of gravity, compressibility, and capillary pressure, and consider the porosity to be constant. Since $p_o - p_w = const$ we neglect the subscript and with S we denote the water saturation. Then Darcy's law of each phase reads $v_\alpha = -\frac{k_{r\alpha}(S)}{\mu_\alpha} k \cdot \nabla p$. Thus the above equations (15.5) and (15.4) simplify to

$$-\operatorname{div}(\lambda(S) k \nabla p) = q, \quad \forall x \in D \tag{15.6}$$

$$\phi \frac{\partial S}{\partial t} + \operatorname{div}(v f(S)) = q_w, \quad \forall x \in D,\ t \in [0, T] \tag{15.7}$$

with $v = -\lambda(S) k \cdot \nabla p$, almost everywhere in Ω. In our application we assume the permeability to be random. To ensure the existence of a unique solution p of (15.6) we assume the permeability k is uniformly bounded and coercive, in the sense that there exist two positive deterministic numbers $0 < k_{\min} \leq k_{\max}$ such that for any $\xi \in \mathbb{R}^d$ and any $1 \leq i, j \leq d$

$$k_{\min} |\xi|^2 \leq \xi^T k(x, \omega) \xi, \quad \left| [k(x, \omega)]_{ij} \right| \leq k_{\max}$$

almost everywhere in D and almost surely. We use a single set of relative permeability curves. The above equations (15.6) and (15.7) are non-linearly coupled, mainly through the saturation-dependent mobilities λ_α in the pressure equation (15.6) and through the pressure-dependent velocity v in the saturation equation (15.7). For each time step we solve (15.6) for the velocity with the saturation of the previous time step. With this velocity we solve (15.7) for the saturation. In the following Sections 15.3, and 16 we show how we solve the pressure equation 15.6 for many different realizations of the permeability and in Section 15.4 we introduce an implicit upwind scheme in combination with the Newton method to determine the solution of the hyperbolic equation (15.7). In Section 17 we combine these techniques with a multi-level Monte Carlo approach to estimate the expectation of the water saturation S. In our numerics (cf. Sec. 18) we will consider two different cases: single-phase flow ($\lambda(S) = 1$ and $f(S)$ nonlinear) and two-phase flow. In both cases, we will compare the saturation field at a certain time instant. Note that in this context the considered equation in Parts I and II can be seen as the pressure equation for single-phase flow.

15.3. Mixed multiscale finite element methods

In this section we present a mixed multiscale finite element method (MsFEM) following [3, 29] which we use to solve the elliptic pressure equation (15.6). First we discuss how to solve a deterministic pressure equation, which corresponds to a permeability $k(x, \omega)$ for a fixed ω. Later we extend this method to a stochastic version. For a fixed permeability realization we write the equation in a mixed form

$$\begin{aligned} (\lambda k)^{-1} v + \nabla p = 0 \quad &\text{in} \quad D \\ \operatorname{div}(v) = q \quad &\text{in} \quad D \\ \lambda(x) k(x) \nabla p \cdot n = g(x) \quad &\text{on} \quad \partial D. \end{aligned} \tag{15.8}$$

For simplicity, we assume Neumann boundary conditions. With $V_h \subset H(\operatorname{div}, D) := \{u : u \in (L^2(D))^d, \operatorname{div} u \in L^2(D)\}$ and $Q_h \subset L^2(D)/\mathbb{R}$ we denote finite dimensional spaces and let $V_h^0 = V_h \cap H_0(\operatorname{div}, D)$, where $H_0(\operatorname{div}, D)$ is $H(\operatorname{div}, D)$ with Neumann homogeneous boundary conditions. By multiplying with test functions $(u_h, b_h) \in V_h \times Q_h$ and integrating by parts we get the numerical approximation of (15.8) on the fine grid. It reads: Find $\{v_h, p_h\} \in V_h \times Q_h$ such that $v_h \cdot n = g_h$ on ∂D and

$$\begin{aligned} ((\lambda k)^{-1} v_h, u_h) - (\operatorname{div} u_h, p_h) &= 0 & \forall u_h \in V_h^0 \\ (\operatorname{div} v_h, b_h) &= (q, b_h) & \forall b_h \in Q_h, \end{aligned} \tag{15.9}$$

with the usual L^2 inner product, $(\cdot, \cdot)$. The idea behind the mixed MsFEM is to approximate the velocity using multiscale basis functions which contain the small-scale features. Therefor these multiscale basis functions are constructed for each edge of every block. To approximate the pressure field we use piecewise constant functions. To construct the multiscale basis we define a partitioning of the domain in polyhedral elements $\bigcup D_i = D$. Let $\mathcal{I}$ be the multi index set of index pairs of two neighboring blocks, i.e if $D_i \cap D_{i'} \neq \emptyset$, $\iota = \{ii'\} \in \mathcal{I}$. This interface we denote with $\Gamma_\iota := D_i \cap D_{i'}$ for $\iota \in \mathcal{I}$. For Γ_ι we define a multiscale basis function $\Psi_{\iota,k} = -k(x) \nabla w_{\iota,k}$

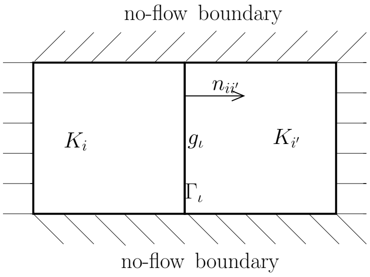


Figure 33: Local problem to solve for multiscale basis.

where $w_{\iota,k}$ is the solution of the auxiliary problem

$$
\begin{aligned}
(-\mathrm{div}(k(x)\nabla w_{\iota,k}))\,|_{D_i} &= \begin{cases} \frac{1}{|D_i|} & \text{if } \int_{D_i} q = 0 \\ \frac{q}{\int_{D_i} q} & \text{else,} \end{cases} \\
(-\mathrm{div}(k(x)\nabla w_{\iota,k}))\,|_{D_{i'}} &= \begin{cases} \frac{-1}{|D_{i'}|} & \text{if } \int_{D_{i'}} q = 0 \\ \frac{-q}{\int_{D_{i'}} q} & \text{else,} \end{cases} \\
-k(x)\nabla w_{\iota,k} \cdot n_{ii'} &= \begin{cases} g_\iota & \text{on } \Gamma_\iota \\ 0 & \text{else,} \end{cases}
\end{aligned}
\tag{15.10}
$$

where the choice of g_ι will be discussed later and $n_{ii'}$ is the normal pointing from D_i to $D_{i'}$ (see Figure 33).

Then the finite dimensional approximation space of the velocity is defined as

$$
\begin{aligned}
V_h(k) &:= \bigoplus_{\iota \in \mathcal{I}} \{\Psi_{\iota,k}\}, \\
V_h^0(k) &:= V_h(k) \cap H_0(\mathrm{div}, D).
\end{aligned}
$$

Note that this space $V_h(k)$ corresponds to a fixed realization of the permeability. In the stochastic framework we determine $V_h(k_j)$ for many realizations k_j of the permeability and construct the approximation space V_h in two different ways, which we introduce in Sections 16.1 and 16.2. The approaches are based on ensemble level methods.

The accuracy of the mixed MsFEM can be affected by the choice of boundary conditions g_ι in (15.10). In the following subsections we introduce two different kinds of boundary conditions, local and global ones. Note that one has to solve a single-phase flow problem for the global boundary conditions.

15.3.1. Mixed MsFEM using local boundary conditions

Piecewise constant coarse-scale fluxes on the boundary of the coarse elements, i.e., $g_\iota = \frac{1}{|\Gamma_\iota|}$ are used in [21]. However, the choice of piecewise constant boundary conditions can lead to large errors between the solution of the original problem and the mixed MsFEM solution. In general, it is possible to consider any boundary conditions that involve local information, e.g., permeabilities in local domains. In this case the boundary condition does not contain any fine-scale features that are in the velocity of the reference solution.

15.3.2. Mixed MsFEM using global boundary conditions

For accurate solutions the boundary conditions should contain fine-scale features. In [21, 40] the authors introduce an oversampling method. Here the main idea is to use larger regions for local problem computations and then for the computation of the multiscale basis functions they use only interior information. However, these approaches are not conservative on the fine grid, which is the reason why we follow [29, 3, 4, 5, 47] where limited global information is used. As in [4] as boundary condition for a fixed realization of the permeability $k_j(x) = k(x, \omega_j)$ and edge $\Gamma_\iota = D_i \cap D_{i'}$ we choose

$$g_\iota(k_j) = \frac{v_j \cdot n_{ii'}}{\int_{\Gamma_\iota} v_j \cdot n_{ii'} ds} \tag{15.11}$$

with the "global" velocity solution v_j which solves (15.8) with the coefficient $k_j(x)$. We note that the construction of the coarse space requires a solution of a single-phase flow problem besides the local solves for basis functions for each coefficient $k_j(x)$. That is the reason why global approaches are computationally more expensive, but more accurate for a given permeability field ([5]). In Sections 19.1 and 20.1 we discuss the accuracies of the two different types of boundary conditions within ensemble level mixed MsFEM context.

15.4. Implicit scheme for solving the saturation equation

After we have introduced the mixed MsFEM in the previous section to solve the pressure equation (15.6) we consider a numerical scheme to solve the time dependent hyperbolic transport equation (15.7) in this section. We combine an implicit upwind scheme with the Newton method. The implicit upwind scheme coincides with a first order discontinuous Galerkin scheme (cf. [48]). First we approximate the time derivative with the difference quotient, i.e.,

$$\frac{\partial S(x,t)}{\partial t} \approx \frac{S^{k+1}(x) - S^k(x)}{\Delta t}$$

with $S^k(x) := S(x, t^k)$ and $\Delta t := t^{k+1} - t^k$. Then we use the following approximation instead of equation (15.7)

$$\phi \frac{S^{k+1}(x) - S^k(x)}{\Delta t} + \theta \text{div } (f(S^{k+1})v) + (1-\theta) \text{div } (f(S^k)v) = q_w \tag{15.12}$$

with $0 \le \theta \le 1$. Thereby the choice $\theta = 0$ corresponds to a fully explicit scheme and $\theta = 1$ to an implicit scheme. If we assume that the saturation is piecewise constant we can write

$$S(x,t) \quad = \quad \sum_{i=1}^{N} S_i(t) \varphi_i(x)$$

with

$$\varphi_i(x) = \begin{cases} 1, & \text{if } x \in D_i \\ 0, & \text{else.} \end{cases}$$

If we multiply (15.12) with a piecewise constant basis function φ_i and integrate over the domain D we get by applying the Gauss theorem:

$$\phi \frac{|D_i|}{\Delta t} \left(S_i^{k+1} - S_i^k \right) + \theta \int_{\partial D_i} [f(S^{k+1})v] \cdot n + (1-\theta) \int_{\partial D_i} [f(S^k)v] \cdot n = Q_i$$

with $Q_i = \int_{D_i} q_w$. Since the flux $f(S)$ over the boundary may be discontinuous we replace it with a numerical consistent and conservative flux

$$\hat{f}(S)_{ii'} = \begin{cases} f(S_i), & \text{if } v \cdot n_{ii'} \geq 0 \\ f(S_{i'}), & \text{if } v \cdot n_{ii'} < 0 \end{cases}$$

for the normal $n_{ii'}$ pointing from D_i to the neighbor cell $D_{i'}$. The boundary integral reads

$$\begin{aligned} \int_{\partial D_i} [f(S^k)v] \cdot n &= \sum_{\Gamma_{ii'} = D_i \cap D_{i'}} \int_{\Gamma_{ii'}} [f(S^k)v] \cdot n_{ii'} \\ &\approx \sum_{\Gamma_{ii'} = D_i \cap D_{i'}} \hat{f}(S^k)_{ii'} \int_{\Gamma_{ii'}} v \cdot n_{ii'} \\ &= \sum_{\Gamma_{ii'} = D_i \cap D_{i'}} \hat{f}(S^k)_{ii'} v_{ii'} \end{aligned}$$

with $v_{ii'} := \int_{\Gamma_{ii'}} v \cdot n_{ii'}$. Then our numerical upwind scheme reads

$$S_i^{k+1} - S_i^k + \frac{\Delta t}{|D_i|\phi} \sum_{\Gamma_{ii'}} \left(\theta \hat{f}(S^{k+1})_{ii'} v_{ii'} + (1-\theta) \hat{f}(S^k)_{ii'} v_{ii'} \right) = \frac{\Delta t}{|D_i|\phi} Q_i.$$

We consider a fully implicit discretization, i.e., $\theta = 1$ to allow arbitrary time steps. In the following we assume that S_i^k is known for all i, $1 \leq i \leq N$ and we are interested in the unknown saturation in the whole reservoir for $t = t^{k+1}$. We define the vector $\mathcal{S} = (S_i^{k+1})_{1 \leq i \leq N}$ and the function

$$H(\mathcal{S}) = \left(S_i^k - \mathcal{S}_i - \frac{\Delta t}{|D_i|\phi} \sum_{\Gamma_{ii'}} \left(\hat{f}(\mathcal{S})_{ii'} v_{ii'} \right) + \frac{\Delta t}{|D_i|\phi} Q_i \right)_{1 \leq i \leq N}.$$

The unknown saturation vector is a zero of this function and we use the Newton method to compute the saturation for each time step. The Newton method reads

$$\mathcal{S}^{n+1} = \mathcal{S}^n - \left(D_H(\mathcal{S}^n)\right)^{-1} H(\mathcal{S}^n)$$

with the Jacobian matrix $D_H(\mathcal{S})$ of $H(\mathcal{S})$. Note that $\mathcal{S}^n$ denotes the Newton iteration after n steps and is does not coincide with S^n. However, if we use the start vector $\mathcal{S}^0 = S^k$, the Newton methods converges to the saturation of the next time step. Crucial for this method is to know the velocity in advance. To solve the system of the pressure and saturation equation (15.6) and (15.7) we solve (15.6) with mixed MsFEM for the velocity with the total mobility depending on the saturation of the previous time step $\lambda(S^k)$ and with this velcotiy we solve (15.7) with the above scheme to compute S^{k+1}.

16. Ensemble level methods for mixed multiscale finite element methods

The ensemble level method was introduced in [19]. Here the authors use it to statistically estimate upscaled two-phase functions, e.g., the total mobility or the flux functions. The main idea is to use a few realizations to cluster some coarse block attributes and to compute the cumulative distribution functions for the corresponding upscaled two-phase functions. They use this information to statistically estimate the upscaled two-phase functions for arbitrary realizations. We apply an ensemble level approach in combination with the above described mixed multiscale FEM (cf. Sec 15.3).

The main idea of ensemble level multiscale methods is to use precomputed multiscale basis functions depending on a few ensemble members to solve the problem for any member of the ensemble.

In this section we introduce two ensemble level methods for mixed MsFEM, the no-local-solve-online ensemble level method (NLSO) and the local-solve-online ensemble level method (NLSO). The computations are divided into offline and online computations. In both methods we construct sets of basis functions based on a few (N_l) realizations of the permeability in the offline stage. We choose the realizations randomly or we use proper orthogonal decomposition (POD). We use POD to find best local basis functions with respect to the velocity.

In general we can also choose offline realizations following the techniques used in reduced basis methods (cf.[51, 52]). For both methods we can use either boundary conditions using global single-phase flow information or local information.

16.1. No-local-solve-online ensemble level multiscale method (NLSO)

The no-local-solve-online ensemble level multiscale method was introduced in [4] to construct a multiscale basis for the velocity approximation space V_h. Again the main idea is to use precomputed quantities for a few realizations to compute the quantities for arbitrary realizations.

In particular, we solve (15.10) for a few coefficients $k_j(x) = k(x, \omega_j)$, $1 \leq j \leq N_l$ at each interior edge Γ_ι and compute the multiscale basis functions $\Psi_{\iota,k_j}(x) = -k_j(x)\nabla w_{\iota,k_j}$ These basis functions are used to solve the equation on a coarse grid without recomputing the basis functions for an arbitrary realization of the permeability. Following [4], we employ an approximation space for the velocity that is defined for multiple k, i.e., we use

$$V_h^{NLSO} := \bigoplus_{j=1}^{N_l} V_h(k_j).$$

In the NLSO approach one does not compute a basis for each realization, but the coarse space of all the precomputed basis functions is used to solve the equation on a coarse grid. Therefore we solve the equations on a $N_l \times |\mathcal{I}|$ coarse space (since we have N_l basis functions on each edge instead of one).

First we solve (15.8) on a coarse grid with the help of the precomputed multiscale basis. Then we use the velocity solution to solve the transport equation (15.7) with an implicit scheme to determine the saturation.

Since the basis functions do not change during the online stage, we can precompute the integrals such as $\int \Psi_{\iota,k_j}\Psi_{\iota',k_j'}$.

To identify the best local basis we use proper orthogonal decomposition. Therefore we compute the N_{POD} eigenvalues with the largest absolute value and the corresponding eigenvectors $\tilde{V}_i = (\tilde{V}_i^1, \cdots, \tilde{V}_i^{N_l})^T$, $1 \leq i \leq N_{POD}$, of the matrix $B_\iota^T B_\iota$. The matrix of precomputed multiscale basis functions for an edge ι we denote with B_ι, i.e., of $B_\iota = \left(\Psi_{\iota,k_j}\right)_{1\leq j\leq N_l}$. Let V be the matrix of the scaled eigenvectors, i.e., for each column of the matrix it holds: $V_i = \frac{1}{\sum_{j=1}^{N_l} \tilde{V}_i^j}\tilde{V}_i$. The POD multiscale basis functions are the columns of $\widetilde{B} = BV$. We have implemented the POD approaches using the L^2 inner product of the velocity. This approach increases the accuracy of the approximation of the velocity and the saturation (cf. Sections 19.1, 20.1). In the POD approach the improvement is larger for the velocity since we consider the L^2 inner product of the velocity.

It is not clear if this inner product is optimal for the quantity of interest: the water saturation. The choice of an optimal inner product for the saturation will be investigated in future work. If we combine the ensemble level mixed method with multi-level Monte Carlo we have not observed any gain using the POD approach (cf. Sections 19.3, 20.3).

16.2. Local-solve-online ensemble level multiscale method (LSO)

The local-solve-online ensemble level multiscale method is similar to a reduced basis approach (e.g., [28, 51]). We apply the LSO approach in the context of mixed multiscale FEM as follows. The offline part is to construct sets of basis functions based on a few (N_l) realizations of the coefficient $k_j(x) = k(x, \omega_j)$, $1 \leq j \leq N_l$. In comparison to reduced basis methods where the realizations fulfill special properties (e.g., [51]), we choose these realizations randomly. After we have solved (15.10) for each realization (cf. Section 15.3) we define the space $V_h^\iota := \bigoplus_{j=1}^{N_l} \Psi_{\iota,k_j}$ for every edge $\iota = \{ii'\}$. In this space V_h^ι we approximate the solution $\tilde{w}_{\iota,\tilde{k}}$ of the auxiliary problem (15.10) with the coefficient $\tilde{k}(x) = k(x, \tilde{\omega})$ for some random $\tilde{\omega}$ in the online phase. As approximation space for the velocity we now define

$$V_h^{LSO} := \bigoplus_{\iota \in \mathcal{I}} \{\tilde{\Psi}_{\iota,\tilde{k}}\},$$

for $\tilde{\Psi}_{\iota,\tilde{k}} = -\tilde{k}(x)\nabla \tilde{w}_{\iota,\tilde{k}}$.

For each computed multiscale velocity we solve the transport equation (15.7) to compute the saturation.

In this approach we use a different multiscale basis for each coefficient. That is the reason why the integrals $\int \tilde{\Psi}_{\iota,\tilde{k}} \tilde{\Psi}_{\iota',\tilde{k}}$ over a coarse block cannot be precomputed. However, since each basis function $\tilde{\Psi}_{\iota,\tilde{k}}$ is a linear combination of the precomputed basis functions Ψ_{ι,k_j}, $1 \leq j \leq N_l$, the calculations can be done inexpensively using precomputed quantities.

As in the previous approach, NLSO, we can use POD to find the best local basis. Again we observe an improvement in the numerical simulations for approximating the velocity and the saturation (cf. Sections 19.1, 20.1). We have not observed any gain of the POD approach in combination with the multi-level Monte Carlo approach.

16.3. Comparison of LSO and NLSO approaches

In contrast to the NLSO method, the online stage of the LSO method is divided into two parts. The first step is to construct a multiscale basis for a particular realization. Therefore we solve in each interior edge of the coarse domain a local problem of size N_l^2. The second step is to solve the global problem on a coarse grid by projecting the solution. In the NLSO approach we solve the equations on a N_l times larger coarse space (since we have N_l basis functions on each edge instead of one). In Figure 34 we summarized the main steps of the NLSO and LSO approach.

In the NLSO case we approximate the velocity with

$$v = \sum_{\iota \in \mathcal{I}} \sum_{j=1}^{N_l} c_{\iota j} \Psi_{\iota,k_j}$$

and in the LSO approach with

$$v = \sum_{\iota \in \mathcal{I}} c_\iota \tilde{\Psi}_{\iota,\tilde{k}} = \sum_{\iota \in \mathcal{I}} c_\iota \sum_{j=1}^{N_l} c_j \Psi_{\iota,k_j}.$$

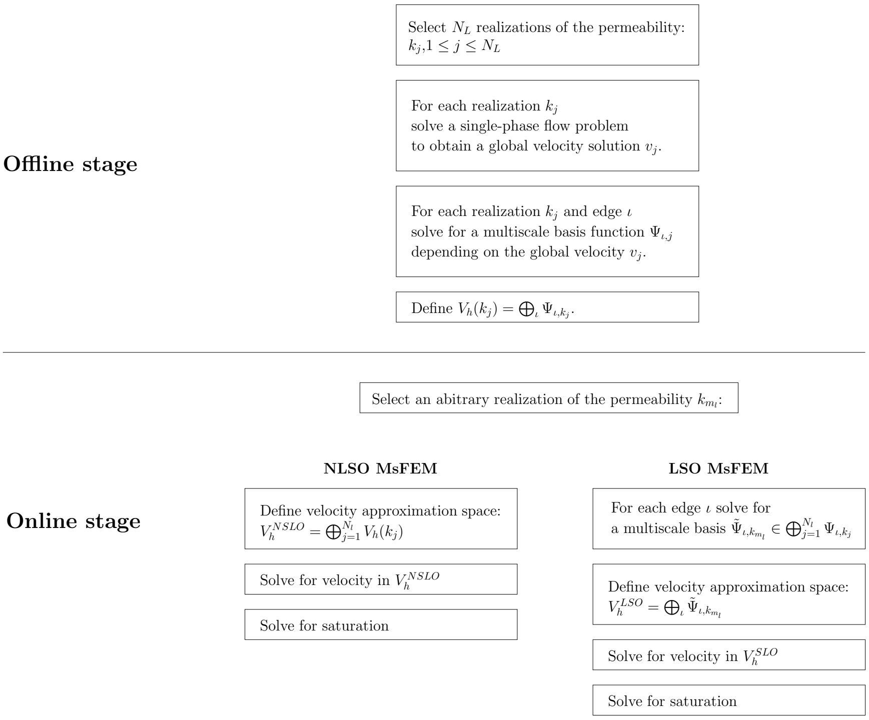

Offline stage
Select N_L realizations of the permeability: $k_j, 1 \leq j \leq N_L$
For each realization k_j solve a single-phase flow problem to obtain a global velocity solution v_j.
For each realization k_j and edge ι solve for a multiscale basis function $\Psi_{\iota,j}$ depending on the global velocity v_j.
Define $V_h(k_j) = \bigoplus_\iota \Psi_{\iota,k_j}$.
Select an abitrary realization of the permeability k_{m_l}:
Online stage
NLSO MsFEM
Define velocity approximation space: $V_h^{NSLO} = \bigoplus_{j=1}^{N_l} V_h(k_j)$
Solve for velocity in V_h^{NSLO}
Solve for saturation
LSO MsFEM
For each edge ι solve for a multiscale basis $\tilde{\Psi}_{\iota,k_{m_l}} \in \bigoplus_{j=1}^{N_l} \Psi_{\iota,k_j}$
Define velocity approximation space: $V_h^{LSO} = \bigoplus_\iota \tilde{\Psi}_{\iota,k_{m_l}}$
Solve for velocity in V_h^{SLO}
Solve for saturation

Figure 34: Steps of the two multiscale methods, of NLSO and LSO.

This is exactly the same if we have the freedom to choose c_j. Note that the coefficients c_j in LSO is determined from the solution of local problems that compute basis functions.

As mentioned above global boundary conditions are more expensive than local ones, but more accurate. That was demonstrated in [49, 29] theoretically and numerically. This is particularly true when the problem is solved multiple times. Next we state that we cannot use LSO type approaches due to the fact that we have use local boundary conditions in the online stage if an accurate solution is sought.

The costs of the online computations of the NLSO approach do not depend on the choice of the boundary conditions g_ι. That is why it is reasonable to choose limited global boundary conditions.

The first part of the online computations of the LSO method is to solve the auxiliary problem (15.10). Boundary conditions using global information would increase the computational cost of the method. Therefore we choose local boundary condition for the online part of the LSO method (global for the offline part). With this choice of boundary conditions the computational costs for NLSO and LSO are comparable. The NLSO approach is more accurate than the LSO method if global boundary conditions are used as we will show in Sections 19.1 and 20.1.

17. Multi-level Monte Carlo using ensemble level mixed multiscale finite element methods

In Section 11.1 we presented the multi-level Monte Carlo method in a general framework. In this section we apply it to approximate the expected value of the water saturation. Therefor we combine it with the previously (cf. Sections 16.1, 16.2) introduced numerical methods. We use the NLSO and LSO approach to calculate the water saturation S_l where N_l realizations of the permeability field are chosen to compute the basis for the whole ensemble. In this case different levels correspond to different accuracies in the velocity approximation space V_h. We are interested in the expected value of the saturation with a large number of precomputed basis functions N_L. We use levels which are less accurate and less computational expensive to approximate the quantity of interest. Particularly, we assume $\|S_l - S\| \sim \frac{1}{N_l^\beta}$, with $N_1 \leq N_2 \leq \cdots \leq N_L$ and $\beta > 0$. As introduced earlier we use a multi-level approximation to approximate the expectation $E[S_L]$

$$E^L(S_L) = \sum_{l=1}^{L} E_{M_l}\left(S_l - S_{l-1}\right),$$

where E_{M_l} denotes the arithmetic mean with M_l samples. Again, we consider the root mean square errors

$$\|E[S_L] - E^L(S_L)\| := \left(E[\left\vert\kern-0.25ex\left\vert\kern-0.25ex\left\vert E[S_L] - E^L(S_L)\right\vert\kern-0.25ex\right\vert\kern-0.25ex\right\vert^2]\right)^{\frac{1}{2}}$$

with an appropriate norm $|\!|\!|\cdot|\!|\!|$. Analogously to the analysis in Section 11 we get

$$\|E[S_L] - E^L(S_L)\| \lesssim \sum_{l=2}^{L} \frac{1}{\sqrt{M_l}} \frac{1}{N_l^\beta} + \frac{1}{\sqrt{M_1}}.$$

To equate the error terms we choose

$$M_l = C \begin{cases} N_L^{2\beta}, & l = 1 \\ \left(\frac{N_L}{N_l}\right)^{2\beta}, & 2 \leq l \leq L, \end{cases} \tag{17.1}$$

then we end with

$$\|E[S_L] - E^L(S_L)\| = O\left(\frac{1}{N_L^\beta}\right).$$

To predict the saturation field we select at each level l a different number of realizations of the permeability field N_l, $N_1 \leq N_2 \leq \cdots \leq N_L$, to build a low dimensional approximation space V_h for the velocity that captures both small scale (sub coarse-grid) spatial variability in the permeability data and stochastic variability due to uncertainties in the data. In particular, we calculate the velocity at level l for M_l realizations with one of the ensemble level mixed MsFEMs to compute the saturations $S_{l,m}$ with $1 \leq l \leq L$ and $1 \leq m \leq M_l$ with the upwind method. With these saturations we build the MLMC approximation of the expected value of the fine scale saturation. For clarity we summarize the basic steps below.

Offline computations

1. *Generation of coarse grid.*
    - Partition the domain into a coarse grid. The coarse grid is a partitioning of the fine grid where each cell in the fine grid belongs to a unique block in the coarse grid and each coarse grid block is connected. [7].
2. *Select N_L realizations from the stochastic permeability distribution $k_j(x)$, $1 \leq j \leq N_L$.*
3. *Construction of the multiscale approximation space $V_h(k_j)$, $1 \leq j \leq N_L$:*
    - For each selected realization $1 \leq j \leq N_L$:

  - Solve (15.8) on the fine grid using a suitable mass conservative numerical method to obtain a "global" velocity solution v_j.
  - Compute the multiscale basis functions: For each edge Γ_ι, $\iota = \{ii'\}$, set

$$g_\iota(k_j) = \frac{v_j \cdot n_{ii'}}{\int_{\Gamma_\iota} v_j \cdot n_{ii'} ds},$$

    and solve (15.10) to obtain w_{ι,k_j} and subsequently Ψ_{ι,k_j}.
  - Define $V_h(k_j) = \bigoplus_{\iota\in\mathcal{I}} \Psi_{\iota,k_j}$.

4. ***NLSO**: Construction of the multiscale approximation space V_h^{NLSO} at level l, $1 \le l \le L$:*
   - Define $V_h^{NLSO} = \bigoplus_{j=1}^{N_l} V_h(k_j)$.

Online computations

5. *Multi-level mixed MsFEM computations for estimating an expectation at level l, $1 \le l \le L$:*
   - Select M_l realizations of the permeability k_{m_l}, $1 \le m_l \le M_l$.
   - **LSO**: Construction of the approximation space V_h^{LSO}
     - Compute basis functions: $\tilde{\Psi}_{\iota,k_{m_l}} \in V_h^\iota = \bigoplus_{j=1}^{N_l} \Psi_{\iota,k_j}$ by solving (15.10) with k_{m_l} and local boundary conditions.
     - Define $V_h^{LSO} = \bigoplus_{\iota\in\mathcal{I}} \{\tilde{\Psi}_{\iota,k_{m_l}}\}$.
   - Solve two-phase flow and transport (15.6)-(15.7) for $S_{l,m}$. At each time step, the velocity field is constructed by solving (15.6) on a coarse grid using NLSO or LSO, respectively.
   - Calculate the arithmetic mean

$$E_{M_l}(S_l - S_{l-1}) = \frac{1}{M_l} \sum_{m=1}^{M_l} (S_{l,m} - S_{l-1,m}).$$

6. *MLMC approximation of $E[S_L]$*

$$E^L(S_L) = \sum_{l=1}^{L} E_{M_l}(S_l - S_{l-1}).$$

17.1. Computational cost

Next, we discuss the computational costs for the MLMC and the standard MC methods. Since the multiscale basis functions are fixed throughout the simulations, we ignore the cost of precomputations in all cases. We assume we solve the pressure equation optimally. Then, for the NLSO method, the computational costs for solving the pressure equation each time instant with N_l basis function sets (at level l) for a coarse grid with mesh size H_j are $2\left(H_j^{-1} - 1\right) N_l^2 H_j^{-1}$. Note that the number of interior edges is $2\left(H_j^{-1} - 1\right) H_j^{-1}$. For the LSO method we solve at each interior edge an N_l^2 sized problem and solving the pressure equation for only one set of basis functions we get $2\left(H_j^{-1} - 1\right)\left(N_l^2 + 1\right) H_j^{-1}$. Note that we have almost the same numerical costs for NLSO and LSO and in the following we neglect the costs of solving the pressure equation for one set of basis functions in the LSO case. So we end with the same numerical costs for the

NLSO and LSO approach. In the LSO approach the multiscale basis of each edge can be solved in parallel, such that the computational time for the LSO ansatz is much smaller. As in Part II, in our simulations, we equate the costs of solving the pressure equation for MC and MLMC and compare the accuracy of these approximations. Although for single-phase flow this comparison is accurate (up to the cost of computation of basis functions), one needs to take into account the cost of solving the saturation equation in two-phase flow and transport simulations. To solve for the saturation we use a coarse-grid velocity field so the computational costs at each time instant are the same at any level. However, the cost of solving the pressure equation is larger than that for the saturation equation on a coarse grid, because there are more degrees of freedom and the convergence of iterative solvers requires many iterations for multiscale problems. Since in the NLSO method we use several basis functions per coarse edge and in the LSO multiscale method we have to calculate the velocity approximation space online, the cost of computing the pressure solution can be several times larger than that of the saturation equation because the coarse system is several times larger for the pressure equation. That is why we will ignore the costs of saturation computation on a coarse grid in two-phase flow examples.

We have the following computational costs for MLMC based on pressure:

$$\begin{aligned} W_{MLMC} &= \sum_{l=1}^{L} \left(2\left(H_j^{-1}-1\right) N_l^2 H_j^{-1}\right) M_l \\ &= 2N_L^{2\beta} H_j^{-1}(H_j^{-1}-1)\left(\sum_{l=2}^{L} N_l^{2-2\beta} + N_1^2\right). \end{aligned}$$

We compare the costs with the computational costs for MC

$$W_{MC} = \left(2\left(H_j^{-1}-1\right)\widehat{N}^2 H_j^{-1}\right)\widehat{M}.$$

In our numerical simulations, we will equate the work and compare the accuracy of MLMC and MC.

If we increase the dimension of the coarse space (corresponds to N_l), we observe that the accuracy of the multiscale methods increases (see [4] and the discussions below). However, in general this accuracy cannot be estimated, so we propose an empirical procedure to estimate the convergence rate β based on simulations with few samples in Sections 19.2 and 20.2. With this estimated rate β we select the number of realizations, M_l based on (17.1), for the MLMC approach.

18. Experimental setup

As a numerical example we consider the traditional quarter-of-a-five-spot with no-flow boundary conditions in a squared domain $D \subset \mathbb{R}^2$. A water injector is placed at the upper left corner while a producer produces whatever (oil or water) reaches the lower right corner. We solve the pressure equation (15.6) with the two different mixed MsFEMs and for the saturation equation (15.7) we use an implicit scheme. In the numerics we solve on a 100 × 100 fine grid a single-phase flow problem (cf. Section 19) and a two-phase flow problem on a 5 × 5 coarse grid (cf. Section 20). For $Y(x,\omega) = \log[k(x,\omega)]$ with $E[Y(x,\omega)] = 0$ and $\operatorname{cov}(x,y) = E[Y(x,\omega)Y(y,\omega)]$ we consider two types of distributions. The first one is log-Gaussian

$$\operatorname{cov}(x,y) = \sigma^2 \exp\left(-\frac{|x_1-y_1|^2}{2\epsilon_1^2} - \frac{|x_2-y_2|^2}{2\epsilon_2^2}\right), \tag{18.1}$$

and the next one is log-Exponential

$$\operatorname{cov}(x,y)=\sigma^2\exp\Big(-\frac{|x_1-y_1|}{\epsilon_1}-\frac{|x_2-y_2|}{\epsilon_2}\Big). \tag{18.2}$$

As previously, we denote with ϵ_1 and ϵ_2 the correlation lengths in each dimension and $\sigma^2=E(Y^2)$ is a constant that represents the variance of the permeability field, which we choose $\sigma^2=2$ in all examples. We use the Karhunen-Loève expansion to parameterize these permeability fields. In the first case, we expect faster decay of the eigenvalues compared to the second case, log-Exponential, for a given set of correlation lengths. In both cases the permeability field $Y(x)$ is given on a 100×100 fine Cartesian grid. In particular we consider:

- Isotropic Gaussian field: correlation length $\epsilon_1=\epsilon_2=0.2$, stochastic dimension 10
- Anisotropic Gaussian field: correlation length $\epsilon_1=0.5$ and $\epsilon_2=0.1$, stochastic dimension 12
- Isotropic Exponential field: correlation length $\epsilon_1=\epsilon_2=0.2$, stochastic dimension 300
- Anisotropic Exponential field: correlation length $\epsilon_1=0.5$ and $\epsilon_2=0.1$, stochastic dimension 350

We apply the above described MLMC approach. To build the multiscale basis, we generate N_L independent realizations of the permeability and for these realizations we solve (15.10). The multiscale basis functions are not recomputed during the simulations, i.e., they are computed at time zero. We choose the realizations for the precomputed multiscale basis functions randomly and we use POD to find a best local basis.

We compare the MLMC accuracy of the saturation with the accuracy of standard MC at level L with the same amount of costs; therefore, we choose

$$\widehat{M}=\frac{\sum_{l=1}^{L}N_l^2M_l}{N_L^2}.$$

As reference S_{ref}, we use the arithmetic mean of M_{ref} samples of the saturation solved with a multiscale velocity. For this velocity the multiscale basis is recomputed for each permeability realization, such that we use exactly the same realization as in the pressure equation. We consider the square root of the arithmetic mean square of the relative L^2 errors, e.g., for MLMC we consider

$$\text{MLMC error}=\sqrt{\frac{1}{N_b}\sum_{j=1}^{N_b}\frac{\|S_{ref}-S^j_{MLMC}\|^2_{L^2}}{\|S_{ref}\|^2_{L^2}}}, \tag{18.3}$$

where S^j_{MLMC} denotes the MLMC approximation of the expectation for a given set of permeability realization $K^j_{off}=(k^j_1,\cdots,k^j_{N_L})$ to compute the multiscale basis functions (cf. (15.10)) and different realizations $K^j_{on}=(k^j_1,\cdots,k^j_{M_1})$ to solve the flow and transport equations (cf. (15.6)-(15.7)) to compute the saturation. In the following Section 19 we present our results for the single-phase flow on the fine grid and in Section 20 we consider two-phase flow on the coarse grid. This part is done with Matlab ([2]). The implementation is based on the code of Aarnes, where the solver for the saturation equation (15.7), the mixed MsFEM method and the ensemble level approach are implemented (cf. [3, 4, 6]). A description of the main part of the code can be found in [6, 45].

19. Numerical results for single-phase flow

As mentioned above, we denote the example as single-phase flow, if the total mobility $\lambda(S) = 1$. Since the mobility does not depend on the saturation, the pressure equation (15.6) does not, either. Therefore we solve (15.6) only once and update the transport equation for each time step. As flux term we use $f(S) = \frac{S^2}{S^2+(1-S)^2}$ (see (15.7)). In this example we consider only the 100×100 fine grid.

19.1. Comparison of the NLSO and the LSO approach for single-phase flow

In this section we study the differences of the two ensemble level mixed MsFEMs, the influence of the boundary conditions using local or limited global information and the influence of using proper orthogonal decomposition to determine a best basis with respect to the velocity.

We show that methods using local boundary conditions have a residual error no matter how many basis functions we pick.

The global boundary conditions are

$$g_\iota(k_j) = \frac{v_j \cdot n_{ii'}}{\int_{\Gamma_\iota} v_j \cdot n_{ii'} ds}$$

as defined in (15.11). In our case the local boundary conditions depend on the permeability realization in the considered cells, i.e.,

$$g_\iota(k_j) = \frac{G_\iota^j}{\int_{\Gamma_\iota^j} G_\iota ds}$$

with

$$G_\iota^j(x) = \frac{2}{\left(\frac{1}{k_j(x-n_{ii'}h)} + \frac{1}{k_j(x+n_{ii'}h)}\right)^{-1}}$$

with the fine mesh size h.

The quantity of interest is the water saturation. Therefore we consider the mean L^2-error for 100 selected realizations k_j of the water saturation computed with one of the ensemble level methods with either local or global boundary conditions and a reference saturation. As reference we use either the fine-scale water saturation S_{ref}^j or the multiscale saturation S_{refloc}^j, where the basis is calculated for the realization k_j with local boundary conditions.

Since the multiscale basis in the POD case is chosen with respect to the velocity we consider the errors for the velocities, also.

We compute the mean L^2-errors for 6 and 12 multiscale basis functions. For the POD we precompute 100 multiscale functions and use the first 6, 12 functions as basis.

We have summarized the results in Tables 7-14. Again, note that we solve the online problems for LSO approach with local boundary conditions.

From the tables we see that in the LSO method we introduce an additional error by solving the online problem with local boundary conditions. The NLSO method seems to be a good approximation of the fine-scale solution, while the solution of the LSO approach is closer to the local reference. For instance, we have for the saturation an error of 2% for randomly chosen realizations and of 1.3% if POD is used for the NLSO approach with global boundary conditions and 6 basis functions for the isotropic Gaussian distribution. For the LSO approach with local boundary conditions we have errors of 2.7% and 1.5% with the local reference without and with POD for the isotropic Gaussian distribution, respectively. The errors of the LSO approach to the

Isotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	2.0154	1.3175	16.2749	16.2132	6.3233	6.3794	16.1875	16.1169
	0.3265	0.2431	16.0556	16.0887	1.9055	1.7348	16.1615	16.1972
$\|S_{refloc} - S\|_{L^2}$	16.2392	16.1794	6.6100	3.1272	15.0033	15.3998	2.6582	1.4822
	16.2149	16.2162	1.8751	1.4486	16.0174	16.0938	0.7492	0.3746
$\|S_{ref} - S_{refloc}\|_{L^2}$	16.2200							

Table 7: Mean saturation errors (percent) of 100 realizations of the isotropic Gaussian distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Isotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	0.1440	0.0944	1.2821	1.3367	0.6470	0.5881	1.3707	1.3628
	0.0195	0.0176	1.3528	1.3603	0.1918	0.1707	1.3661	1.3669
$\|v_{refloc} - v\|_{L^2}$	1.3792	1.3719	0.6375	0.3101	1.3806	1.3745	0.2070	0.1041
	1.3671	1.3670	0.2041	0.1406	1.3686	1.3743	0.0610	0.0290
$\|v_{ref} - v_{refloc}\|_{L^2}$	1.3669							

Table 8: Mean velocity errors (percent) of 100 realizations of the isotropic Gaussian distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Isotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	19.7784	17.7025	32.1457	31.8803	24.2868	22.6637	31.7842	31.3074
	15.5608	12.9127	31.3271	30.8887	20.0685	17.7077	31.1171	31.0616
$\|S_{refloc} - S\|_{L^2}$	25.9793	27.3023	22.5036	18.6939	23.8879	23.7598	22.5670	16.0498
	26.8417	27.7851	18.5179	15.6346	24.5747	25.5206	16.1490	12.9676
$\|S_{ref} - S_{refloc}\|_{L^2}$	30.3272							

Table 9: Mean saturation errors (percent) of 100 realizations of the isotropic Exponential distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Isotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	2.8652	2.2179	3.2436	3.1131	3.6413	3.6586	3.5080	3.5085
	2.1377	1.5891	3.1698	3.0914	3.2082	2.8214	3.4817	3.4847
$\|v_{refloc} - v\|_{L^2}$	3.8502	3.6259	2.9904	2.6848	4.1118	4.1260	2.6649	2.0330
	3.6400	3.5168	2.5808	2.2091	3.9426	3.8258	2.0581	1.5502
$\|v_{ref} - v_{refloc}\|_{L^2}$	3.4769							

Table 10: Mean velocity errors (percent) of 100 realizations of the isotropic Exponential distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Anisotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\lVert S_{ref} - S \rVert_{L^2}$	4.4798	3.7621	28.0264	27.3662	10.6921	9.9636	27.5760	27.6249
	1.4889	1.0872	27.3692	27.3779	5.9529	5.4429	27.6672	27.6798
$\lVert S_{refloc} - S \rVert_{L^2}$	27.4733	27.5175	12.3253	6.2222	25.1565	25.3781	5.0260	2.9356
	27.6664	27.6771	4.2140	3.7559	26.6987	26.8279	2.0322	1.1568
$\lVert S_{ref} - S_{refloc} \rVert_{L^2}$	27.7428							

Table 11: Mean saturation errors (percent) of 100 realizations of the anisotropic Gaussian distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Anisotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\lVert v_{ref} - v \rVert_{L^2}$	0.4539	0.3828	1.9906	2.2211	1.3898	1.2797	2.3359	2.3476
	0.1232	0.0931	2.2799	2.2914	0.7342	0.5897	2.3553	2.3527
$\lVert v_{refloc} - v \rVert_{L^2}$	2.3963	2.3707	1.3507	0.8364	1.3898	1.2797	2.3359	2.3476
	2.3536	2.3529	0.6093	0.5135	0.7342	0.5897	2.3553	2.3527
$\lVert v_{ref} - v_{refloc} \rVert_{L^2}$	2.3518							

Table 12: Mean velocity errors (percent) of 100 realizations of the anisotropic Gaussian distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Anisotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\lVert S_{ref} - S \rVert_{L^2}$	20.7988	19.7001	35.6866	34.7461	26.3559	25.0294	35.6875	33.4379
	16.1815	14.0968	33.6023	33.1095	21.4738	20.5792	33.9996	33.2703
$\lVert S_{refloc} - S \rVert_{L^2}$	29.3508	30.0347	25.1490	21.6110	25.0723	25.5648	25.2204	16.2809
	30.1783	31.0464	19.8822	16.7638	27.1934	27.4769	17.5943	13.0112
$\lVert S_{ref} - S_{refloc} \rVert_{L^2}$	33.8186							

Table 13: Mean saturation errors (percent) of 100 realizations of the anisotropic Exponential distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

Anisotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\lVert v_{ref} - v \rVert_{L^2}$	3.5702	2.8821	3.7861	3.6260	4.5283	4.5108	4.0864	4.0154
	2.8207	2.2679	3.6015	3.5865	3.9577	3.8736	3.9667	3.9984
$\lVert v_{refloc} - v \rVert_{L^2}$	4.3484	3.9808	3.1189	2.7728	4.5251	4.5562	2.9389	2.1638
	4.1108	4.0028	2.6538	2.3411	4.3364	4.3447	2.2073	1.6922
$\lVert v_{ref} - v_{refloc} \rVert_{L^2}$	4.0063							

Table 14: Mean velocity errors (percent) of 100 realizations of the anisotropic Exponential distribution for the different methods and boundary conditions for the single-phase flow example, $6, 12$ basis functions.

global reference is of the same size as the error between the global reference and the local one, independent of the underlying distribution. In both approaches global boundary conditions give a better approximation of the fine-scale solution (at least if we choose the number of precomputed basis functions large enough). For the velocities the behavior of the mean errors is comparable, but the approximations are much more accurate. Furthermore, the approximation of the fine-scale velocity is more accurate with global boundary conditions independent of the number of basis functions even for the LSO approach. For this reason we will only consider global boundary conditions in the following. In Figure 35 we show the water saturations with global boundary conditions for one sample of the permeability for the considered distributions for all methods.

19.2. Numerical study of the convergence rate for single-phase flow

In this section we investigate if the assumption $\|S_l - S\| = C\frac{1}{N_l^\beta} =: \delta_l$, with $\delta_1 > \delta_2 > \cdots \delta_L$ (cf. Section 17) is fulfilled for the NLSO and the LSO approach. Here S denotes the saturation with the basis calculated for the permeability realization which is used in the pressure equation and S_l the saturation with a precomputed basis with N_l permeability realizations. With these δ_ls, it is possible to find appropriate choices of realizations M_l at each level, namely

$$M_l = C \begin{cases} \left(\frac{1}{\delta_L}\right)^2 (\mathrm{std}(S) + \delta_1^2), & l = 1 \\ \left(\frac{\delta_l}{\delta_L}\right)^2, & 2 \leq l \leq L. \end{cases}$$

To determine the convergence rate, we choose $\mathcal{N} = (3, 6, 12)$ and calculate the mean over $M = 100$ permeability realizations as follows. For 10 sets of permeability realizations we compute the multiscale basis and each of these sets we use to compute the error for 10 different permeability realizations. We compute the arithmetic mean of the 10×10 numbers. In Table 15 we summarize the error ratios $\frac{\delta_l}{\delta_L} = \frac{\|S_l - S\|_{L^2}}{\|S_L - S\|_{L^2}}$.

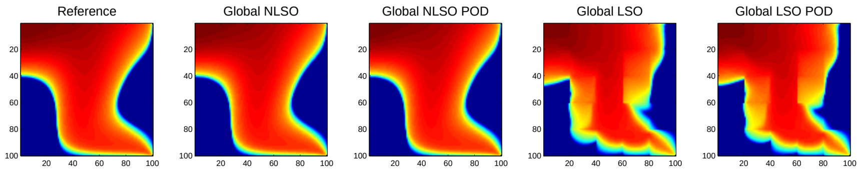
For the NLSO approach the resulting δs depend on the underlying distribution. If we precompute the multiscale basis functions with randomly chosen realizations we get for the isotropic Gaussian distribution for the different levels the δ ratios $(13.5, 5.1, 1)$ and for the anisotropic Gaussian $(5.2, 2.6, 1)$. For the two Exponential distributions the ratios are almost the same, namely $(1.6, 1.3, 1)$. For POD the resulting ratios are comparable. Since the δs do not decrease fast we choose larger N_ls in Section 19.3 where we combine the ensemble level mixed MsFEMs with MLMC.

For the LSO approach all ratios are close to one, i.e., the errors are almost independent of the number of used precomputed basis functions. However, since we observed in the previous section that the LSO approach does not converge to the fine scale saturation we could not expect anything. However, we choose the same numbers of precomputed basis functions N_l, $1 \leq l \leq L$, as in the NLSO approach in Section 19.3.

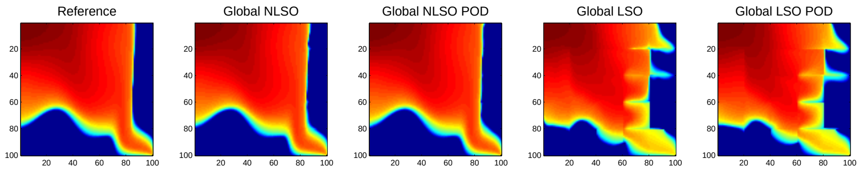
19.3. Ensemble level mixed MsFEM for single-phase flow

In this section we combine the introduced ensemble level mixed MsFEMs to solve the pressure equation (15.6), the NLSO and LSO approach, with multi-level Monte Carlo. We consider both, randomly chosen realizations for the precomputations of the multiscale basis functions and the POD approach.

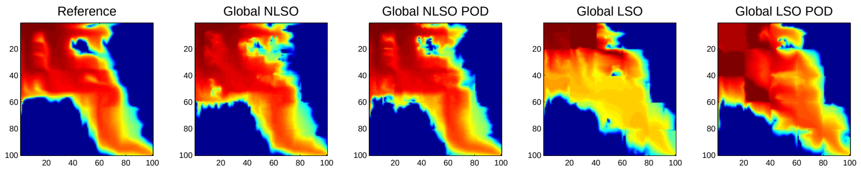
In Table 16 we summarize the parameters used and the resulting relative errors of our simulations for the different distributions and methods. We start with $N_b = 100$ in (18.3) for NLSO with randomly chosen realization. However, for computational reasons we decrease this number for



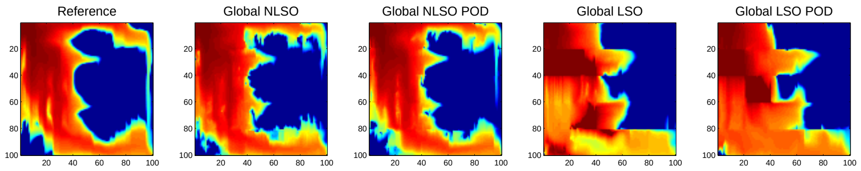
(a) Isotropic Gaussian



(b) Anisotropic Gaussian



(c) Isotropic Exponential



(d) Anisotropic Exponential

Figure 35: On realization of the water saturation for global boundary conditions for the different methods with 12 precomputed basis functions.

	isotropic Gaussian	anisotropic Gaussian	isotropic Exponential	anisotropic Exponential
NLSO				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	13.5458	5.2335	1.6024	1.6183
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	5.0930	2.6086	1.2746	1.3254
LSO				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.0773	1.0378	1.0991	1.1817
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.0018	1.0081	1.0087	1.0608
NLSO with POD				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	12.7403	5.0250	1.9021	2.0390
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	4.7024	2.2227	1.4007	1.4202
LSO with POD				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.0072	1.1122	1.2138	1.2974
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.0037	1.0046	1.0395	1.0510

Table 15: Convergence of the single-phase flow example for the different methods and distributions with $\mathcal{N} = (3, 6, 12)$.

the other methods, in particular, we use $N_b = 20$. If not otherwise stated we use $N_b = 20$. For the online computations we choose the number of samples of the permeability independent of the underlying distribution at each level as $\mathfrak{M} = (70, 20, 10)$. The dimension of the approximations (defined as the number of independent samples selected to construct the multiscale space, N_l at level l) for the Exponential distributions is eight times larger than for the Gaussian tests. For the proper orthogonal decomposition examples we precompute multiscale basis functions for a large number N_{POD} compared to N_L and determine the used N_l basis functions with POD. More precisely we use $N_{POD} = 100$ for underlying Gaussian distributions and $N_{POD} = 500$ for the Exponential cases. As mentioned before we equate the computational costs and compare the resulting relative errors for the MLMC and MC approaches. With this choice of realizations for the MLMC method, we get for MC with equated costs $\widehat{M} = 20$, where $\widehat{M}$ is the number of permeability realizations needed for forward simulations.

	isotropic Gaussian	anisotropic Gaussian	isotropic Exponential	anisotropic Exponential
(N_1, N_2, N_3)	$(3, 6, 12)$	$(3, 6, 12)$	$(24, 48, 96)$	$(24, 48, 96)$
(M_1, M_2, M_3)	$(70, 20, 10)$	$(70, 20, 10)$	$(70, 20, 10)$	$(70, 20, 10)$
$\widehat{N}$	12	12	96	96
$\widehat{M}$	20	20	20	20
M_{MCref}	500	500	500	500
N_{POD}	100	100	500	500
MLMC error NLSO $N_b = 100$	0.0769	0.0853	0.0952	0.0879
MC error NLSO $N_b = 100$	0.1386	0.1282	0.1418	0.1321
$\frac{\text{MC error}}{\text{MLMC error}}$ NLSO $N_b = 100$	1.80	1.50	1.49	1.50
MLMC error NLSO	0.0758	0.0847	0.0934	0.0889
MC error NLSO	0.1439	0.1222	0.1389	0.1343
$\frac{\text{MC error}}{\text{MLMC error}}$ NLSO	1.90	1.44	1.49	1.51
MLMC error LSO	0.1101	0.1189	0.1781	0.1711
MC error LSO	0.3027	0.4244	0.2112	0.1839
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	2.75	3.57	1.19	1.07
MLMC error POD NLSO	0.0789	0.0824	0.0947	0.0864
MC error POD NLSO	0.1332	0.1343	0.1532	0.1348
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	1.67	1.63	1.62	1.56
MLMC error POD LSO	0.1020	0.1228	0.1819	0.1757
MC error POD LSO	0.1462	0.1447	0.2024	0.2029
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	1.4	1.18	1.11	1.15

Table 16: Parameters and errors for the single-phase flow example.

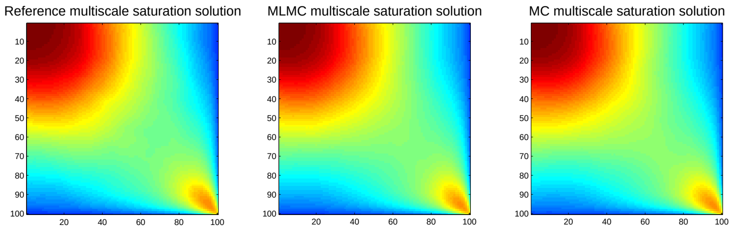
In the NLSO case with $N_b = 100$ we get for MLMC relative errors of approximately 9 percent and for MC approximately 13 percent. This result is almost independent of the underlying distribution, but note that for the Exponential test cases the computational work is eight times larger than for the Gaussian ones. For $N_b = 20$ the errors do not change significantly. So it makes sense to reduce the computational time and use $N_b = 20$ for the other cases.

For all considered combinations -NLSO, LSO with POD or without POD for the different distributions- we increase the accuracy with the help of MLMC in comparison to MC at the largest level with equated costs. As expected the errors for the NLSO method are smaller than the errors for the LSO approach independent of the underlying distributions. For instance, we have for the isotropic Exponential distribution for MLMC an error of 9% for the NLSO case and 18% otherwise (cf. Table 16). This coincides with our results in 19.1. However, note that LSO approach provides good results although the δ ratios are close to one (cf. Section 19.2).

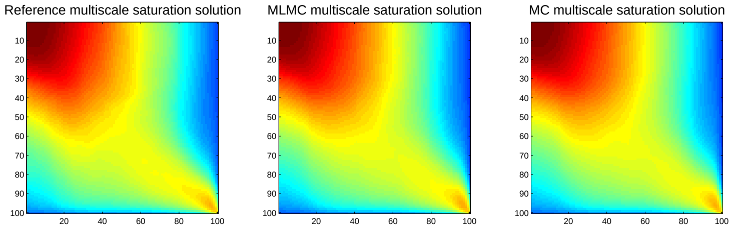
There is no significant influence of using POD. It seems to decrease the influence of the underlying distribution. Particularly, without POD the ratio of the MC and the MLMC error is between 1.44% and 1.90% for NLSO. For LSO it lies in the interval $[1.07, 3.57]$. If we use POD we get $[1.56, 1.67]$ for NLSO and $[1.11, 1.40]$ for LSO.

The resulting mean water saturations for the different covariance functions (isotropic and anisotropic Gaussian, isotropic and anisotropic Exponential) and the different methods (MLMC, MC, reference) for the considered ensemble level mixed MsFEMs we illustrate in Figures 36-39. In contrast to the LSO method one observes no differences for MLMC, MC and the reference in the NLSO approach (cf. Fig.36, 38). In the figures corresponding to the LSO approach (Fig. 37, 39) one can see the coarse grid due to the local boundary conditions in the online stage. Furthermore one can observe differences for the MLMC and the MC approach, e.g., 37(a). The error of the MC approach for the Gaussian distributions is significantly smaller if one uses POD in the case of LSO (cf. Table 16). This can be seen in the corresponding Figure 39 as well.

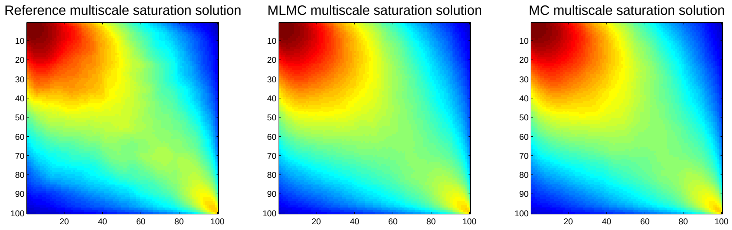
If one is interested in the expected value of the water saturation at the producer only, the MLMC approach increases the accuracy in comparison to the MC approach with equated costs. The improvement of MLMC is approximately the same, but the errors are much smaller. For instance, we get an error of 0.6% with the MLMC approach and on of 1.3% for MC in the case of the NLSO method with POD for an isotropic Gaussian distribution. For the LSO approach the corresponding errors are 0.7% and 1.2% for MLMC and MC, respectively.



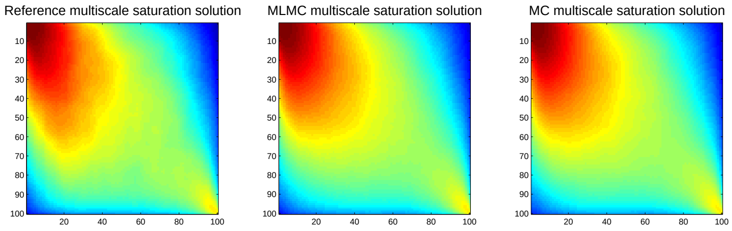
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

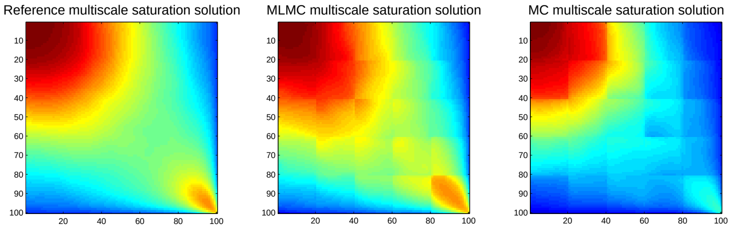


(c) Isotropic Exponential

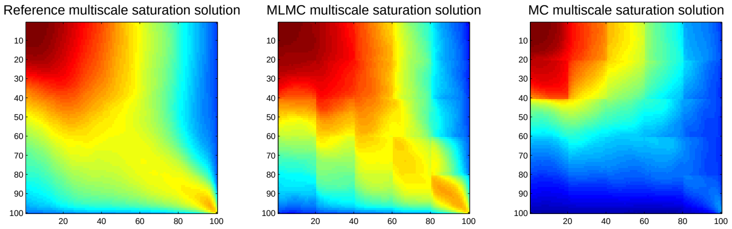


(d) Anisotropic Exponential

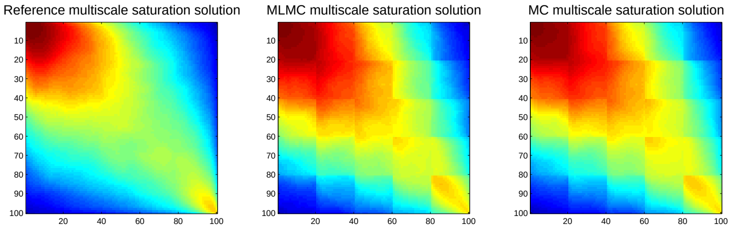
Figure 36: Water saturation for NLSO for MLMC and MC for the different distributions for single-phase flow, $N_b = 100$.



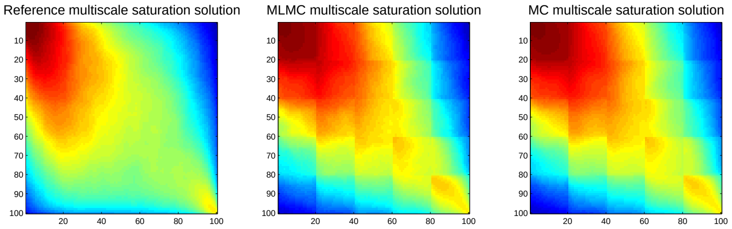
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

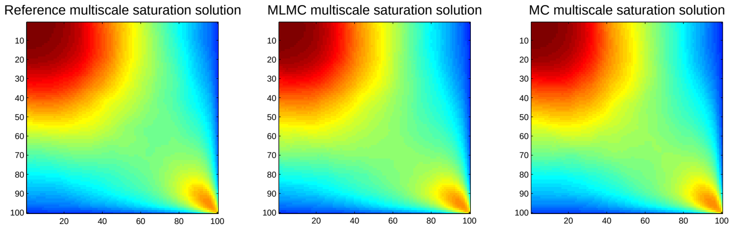


(c) Isotropic Exponential

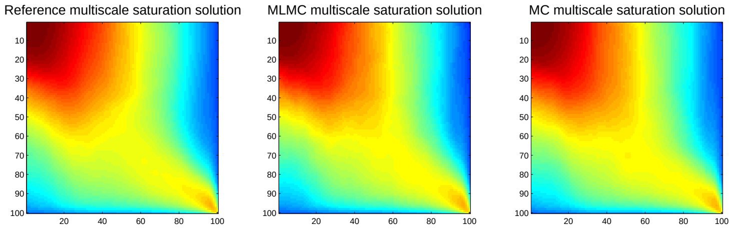


(d) Anisotropic Exponential

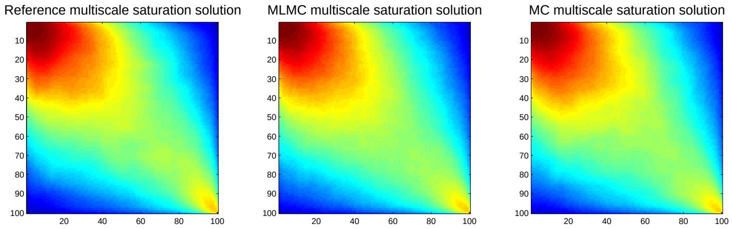
Figure 37: Water saturation for LSO for MLMC and MC for the different methods and distributions for single-phase flow.



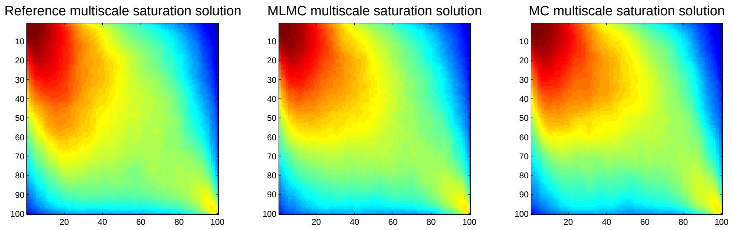
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

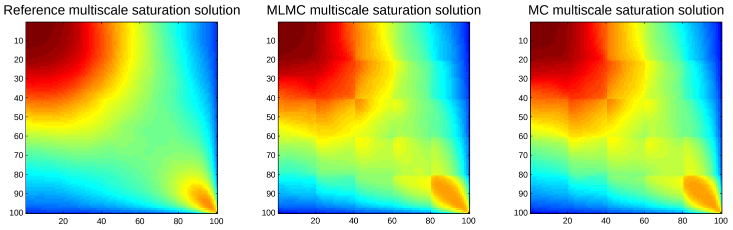


(c) Isotropic Exponential

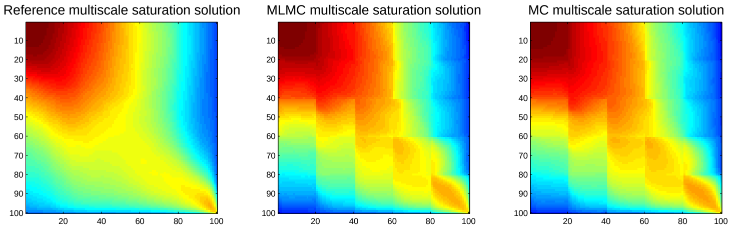


(d) Anisotropic Exponential

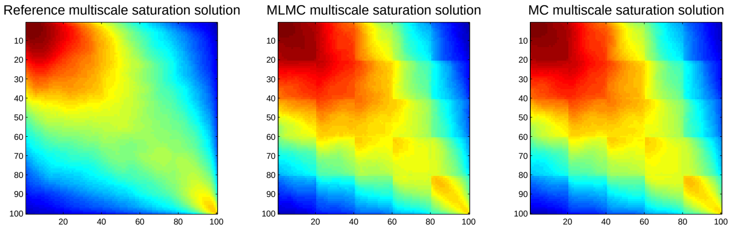
Figure 38: Water saturation for NLSO with POD for MLMC and MC for the different methods and distributions for single-phase flow.



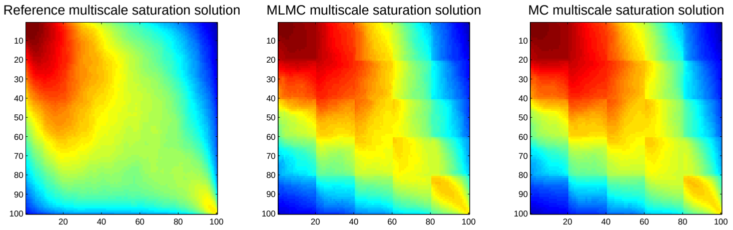
(a) Isotropic Gaussian



(b) Anisotropic Gaussian



(c) Isotropic Exponential



(d) Anisotropic Exponential

Figure 39: Water saturation for LSO with POD for MLMC and MC for the different methods and distributions for single-phase flow.

20. Numerical results for two-phase flow

In the two-phase flow example we consider a 5×5 coarse grid, such that every grid block contains a 20×20 cell partition from the fine grid. In equation (15.6) and (15.7) we choose the following quantities $\mu_w = 1$, $\mu_0 = 1$, $k_{rw}(S) = S^2$ and $k_{ro}(S) = (1-S)^2$. In this case the total mobility $\lambda(S) = \frac{k_{rw}(S)}{\mu_w} + \frac{k_{ro}(S)}{\mu_o} = S^2 + (1-S)^2$ depends on the water saturation and we solve the pressure equation on the 5×5 coarse grid with mixed MsFEM for each time step.

20.1. Comparison of the NLSO and the LSO approach for two-phase flow

Analogously to the single-phase flow case we study the differences of the two ensemble level mixed MsFEMs, the influence of boundary conditions using local or limited global information and the influence of using proper orthogonal decomposition to determine a best basis with respect to the velocity.

Again we observe that methods using local boundary conditions have a residual error no matter how many basis functions we pick.

In this case we consider the multiscale saturations S^j_{ref} and S^j_{refloc} as references where the basis is calculated for the realization k_j with global boundary conditions or with local boundary conditions, respectively, as well as the corresponding velocities.

As before, we compute the mean L^2-errors for 6 and 12 multiscale basis functions. For the POD we precompute 100 multiscale functions and use the first 6, 12 functions as basis.

	Isotropic Gaussian							
	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	1.3792	1.2134	7.6178	7.8785	4.4814	3.4160	7.9852	8.1864
	0.8252	0.8815	7.9849	8.0405	1.3310	1.1110	8.1728	8.1900
$\|S_{refloc} - S\|_{L^2}$	8.3251	8.3319	1.8642	1.0486	8.4147	8.6531	0.6369	0.2410
	8.2087	8.2358	0.5198	0.3121	8.2615	8.1302	0.1040	0.0436
$\|S_{ref} - S_{refloc}\|_{L^2}$	8.1947							

Table 17: Mean saturation errors (percent) of 100 realizations of the isotropic Gaussian distribution for the different methods and boundary conditions for the two-phase flow example, 6, 12 basis functions.

	Isotropic Gaussian							
	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	0.1132	0.1036	0.3684	0.3843	0.2614	0.2222	0.3964	0.4006
	0.0542	0.0552	0.3874	0.3935	0.0776	0.0647	0.4028	0.4034
$\|v_{refloc} - v\|_{L^2}$	0.4081	0.4005	0.1000	0.0499	0.4655	0.4573	0.0337	0.0156
	0.4059	0.4027	0.0308	0.0178	0.4014	0.3987	0.0071	0.0034
$Vertv_{ref} - v_{refloc}\|_{L^2}$	0.4040							

Table 18: Mean velocity errors (percent) of 100 realizations of the isotropic Gaussian distribution for the different methods and boundary conditions for the two-phase flow example, 6, 12 basis functions.

We have summarized the results in Tables 17-24. Again, note that we solve the online problems of the LSO method with local boundary conditions.

Similar to the single-flow case we observe an additional error due to the online problem in the LSO method. Again the NLSO approach seems to be a good approximation of the solution with

Isotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	10.2557	8.2078	19.7424	19.4531	13.3052	12.5890	19.6968	19.7828
	7.0350	5.2391	19.7486	19.7901	10.1616	8.8308	19.9663	20.2782
$\|S_{refloc} - S\|_{L^2}$	18.7326	19.5738	11.8344	10.3493	17.2977	17.2377	10.4369	8.0460
	19.5572	20.1890	10.0678	8.4615	18.4222	18.8266	7.9656	5.6863
$VertS_{ref} - S_{refloc}\|_{L^2}$	20.7051							

Table 19: Mean saturation errors (percent) of 100 realizations of the isotropic Exponential distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

Isotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	0.7057	0.5706	1.3037	1.1934	0.8832	0.8323	1.2297	1.1719
	0.4180	0.3244	1.2228	1.1397	0.6529	0.5720	1.1815	1.1889
$\|v_{refloc} - v\|_{L^2}$	1.2120	1.1911	0.8416	0.6359	1.1373	1.1307	0.6899	0.5008
	1.1680	1.1784	0.6700	0.4968	1.1306	1.1950	0.4901	0.3718
$\|v_{ref} - v_{refloc}\|_{L^2}$	1.2040							

Table 20: Mean velocity errors (percent) of 100 realizations of the isotropic Exponential distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

Anisotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	2.6836	2.6356	16.5501	17.1777	5.3143	5.2178	17.6067	17.8641
	1.5851	1.4145	17.6444	17.6032	3.4046	2.9868	17.9567	17.9591
$\|S_{refloc} - S\|_{L^2}$	18.2849	18.3574	4.8377	2.6597	17.9094	17.3160	1.3827	0.5543
	18.1816	18.2574	1.2725	1.0496	17.9992	17.9370	0.3430	0.1005
$\|S_{ref} - S_{refloc}\|_{L^2}$	17.9892							

Table 21: Mean saturation errors (percent) of 100 realizations of the anisotropic Gaussian distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

Anisotropic Gaussian

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	0.1339	0.1353	0.7849	0.8243	0.2944	0.2868	0.8141	0.8259
	0.0739	0.0713	0.8187	0.8108	0.1823	0.1491	0.8302	0.8303
$\|v_{refloc} - v\|_{L^2}$	0.8518	0.8531	0.2384	0.1328	0.8307	0.8175	0.0582	0.0269
	0.8428	0.8388	0.0636	0.0483	0.8438	0.8418	0.0161	0.0048
$\|v_{ref} - v_{refloc}\|_{L^2}$	0.8317							

Table 22: Mean velocity errors (percent) of 100 realizations of the anisotropic Gaussian distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

Anisotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|S_{ref} - S\|_{L^2}$	10.9965	11.0470	22.1169	21.5021	15.4963	14.4225	21.6356	21.7126
	7.4683	6.4064	21.2290	21.2887	11.1717	10.9566	22.1763	22.1820
$\|S_{refloc} - S\|_{L^2}$	21.2597	21.1819	13.3213	11.5873	17.9521	18.1224	12.3714	8.3137
	22.1649	22.3475	11.0198	9.4439	20.0669	19.9342	8.4631	5.8418
$\|S_{ref} - S_{refloc}\|_{L^2}$				23.7082				

Table 23: Mean saturation errors (percent) of 100 realizations of the anisotropic Exponential distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

Anisotropic Exponential

	gNLSO		gLSO		lNLSO		lLSO	
	no POD	POD	no POD	POD	no POD	POD	no POD	POD
$\|v_{ref} - v\|_{L^2}$	0.7689	0.7371	1.4086	1.3380	1.1682	0.9662	1.4078	1.3604
	0.5040	0.4776	1.2925	1.2982	0.6848	0.6967	1.3351	1.3844
$\|v_{refloc} - v\|_{L^2}$	1.3688	1.2647	0.8417	0.6131	1.2812	1.1923	0.8408	0.4765
	1.4036	1.4278	0.6704	0.5211	1.2220	1.2361	0.5335	0.3568
$\|v_{ref} - v_{refloc}\|_{L^2}$				1.4636				

Table 24: Mean velocity errors (percent) of 100 realizations of the anisotropic Exponential distribution for the different methods and boundary conditions for the two-phase flow example, $6, 12$ basis functions.

global boundary conditions, while the solution of LSO method is closer to the local reference. The errors in the LSO method are of the same size as the error between the reference solutions using local or global boundary conditions, i.e., the LSO approach does not remove the residual error. We will consider global boundary conditions only, since in both approaches global boundary conditions give a better approximation of the solution using global information if the number of precomputed basis functions is large enough.

For the velocities the behavior of the mean errors is comparable, but the approximations are much more accurate.

In Figure 40 we show the water saturations with global boundary conditions for one sample of the permeability for the considered distributions for all methods.

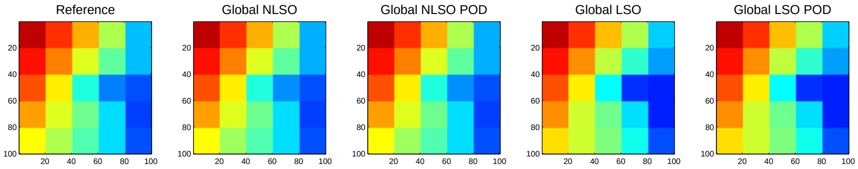
20.2. Numerical study of the convergence rate for two-phase flow

As in Section 19.2 we investigate if the assumption $\|S_l - S\| = C\frac{1}{N_l^\beta} =: \delta_l$, with $\delta_1 > \delta_2 > \cdots \delta_L$ (cf. Section 17) is fulfilled for the NLSO and the LSO approach.

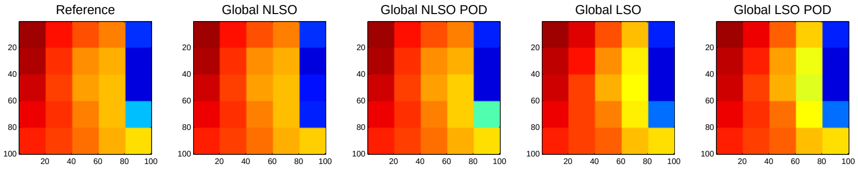
We choose the same parameters as in the single-phase case, namely $\mathcal{N} = (3, 6, 12)$ and $M = 100 = 10 \times 10$.

In the two-phase flow case δ ratios are almost independent of the underlying distribution. For the NLSO approach the ratios for the Gaussian distributions are slightly larger than the Exponential ones, if we do not apply POD. With POD the ratios of the isotropic Gaussian distributions decrease in comparison to the corresponding ratio without POD. This is reasonable because the errors for the smallest level, $N_1 = 3$, are significantly smaller due to POD.

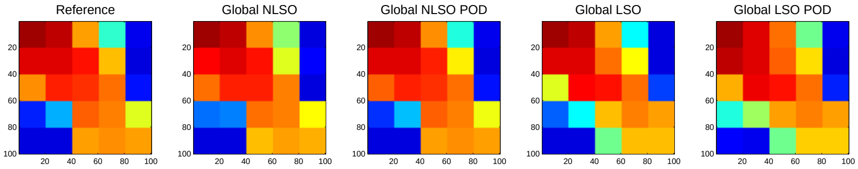
Again the LSO method does not give appropriate δs. In many cases the ratios are even smaller than one. That means, the assumption for our error estimate in Section 17 are not valid. Since the LSO saturation does not converge to coarse reference saturation (cf. Section 20.1), we choose the number of precomputed basis functions as for the NLSO approach.



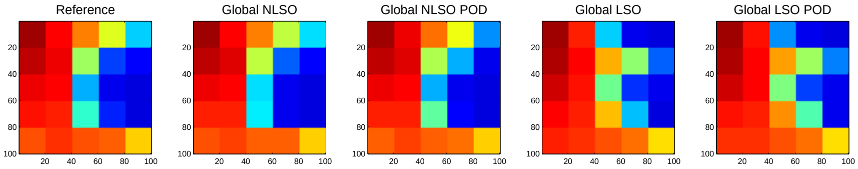
(a) Isotropic Gaussian



(b) Anisotropic Gaussian



(c) Isotropic Exponential



(d) Anisotropic Exponential

Figure 40: One realization of the water saturation for global boundary conditions for the different methods for two-phase flow with 12 basis functions.

	isotropic Gaussian	anisotropic Gaussian	isotropic Exponential	anisotropic Exponential
NLSO				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	2.8265	2.5523	2.0302	1.9180
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.6382	1.6738	1.5291	1.4913
LSO				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	0.9970	0.9244	1.0891	0.9357
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	0.9491	0.9492	1.0010	0.9559
NLSO with POD				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.8794	3.4429	3.0525	3.5156
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	1.3096	1.7965	1.6624	1.9182
LSO with POD				
$\frac{\|S_1-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	0.8749	0.8903	1.1059	1.0248
$\frac{\|S_2-S\|_{L^2}}{\|S_3-S\|_{L^2}}$	0.9806	0.9951	1.0024	1.0251

Table 25: Convergence of the two-phase flow example for the different methods and distributions with $\mathcal{N} = (3, 6, 12)$.

20.3. Ensemble level mixed MsFEM for two-phase flow

In this section we combine the introduced ensemble level mixed MsFEMs to solve the pressure equation (15.6), the NLSO and LSO approach, with multi-level Monte Carlo. As in the single-phase flow example we consider both, randomly chosen realizations for the precomputation of the multiscale basis functions and the POD approach.

The parameters we use for our simulations and the resulting relative errors are summarized in Table 26. We choose all parameters as in the single-phase flow case. Note that for the Exponential test cases the computational work is eight times larger than for the Gaussian ones.

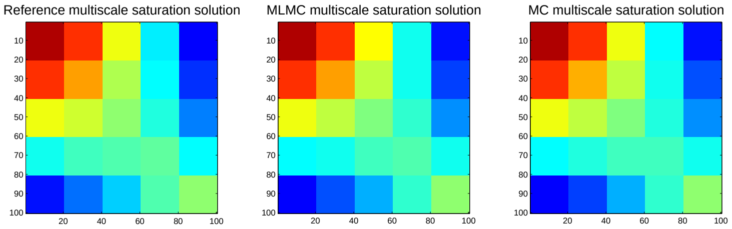
Again we observe no significant difference in the errors with $N_b = 100$ or $N_b = 20$ for the NLSO method with randomly chosen realizations for the precomputation of multiscale basis function. For this reason we consider $N_b = 20$ for the other examples.

As in the single-phase flow example we increase the accuracy by using MLMC in comparison to MC with equated costs independent of the underlying distribution or method, NLSO or LSO. Here the difference of the NLSO and the LSO method seems to depend on the underlying distribution. If it is anisotropic the gain of using the NLSO approach is larger, e.g., we have a relative MLMC error of 5.43% for NLSO and 11.61% for LSO without POD for the anisotropic Exponential distribution. This is consistent with the errors with POD (cf. Table 26). For the isotropic distributions some errors of the LSO approach are even smaller than for the NLSO method. If one is interested in coarse quantities only, both methods -NLSO and LSO- give comparable results.

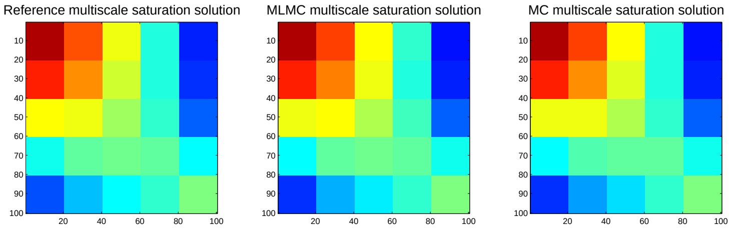
However, note that the LSO approach provides results as good as the NLSO, although the assumption for our error estimate is not valid.

In Figures 41-44 one can find the mean water saturations for the different covariance functions (isotropic and anisotropic Gaussian, isotropic and anisotropic Exponential) and the different methods (MLMC, MC, reference) for the considered ensemble level mixed MsFEMs. For some distributions there is a slight difference between the reference and MLMC or the reference and MC. However, it is difficult to see any difference between the MLMC and MC figures. For instance in Figure 42(d) the figures for MLMC and MC are slightly different. However, these differences are decreased by the use of POD.

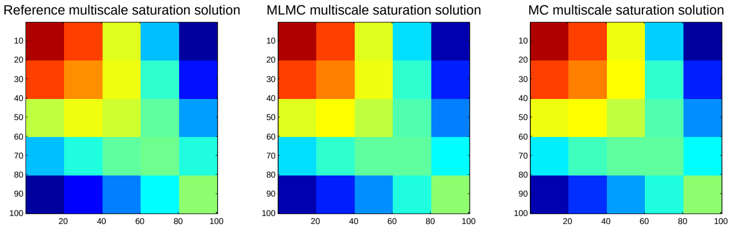
As in the single-phase example we observe a similar behavior if we consider the water saturation in the outlet instead of the saturation in the whole reservoir. The improvement of MLMC is approximately the same, but the errors are much smaller. For instance, we observe an error of 0.6% with the MLMC approach and on of 1.3% for MC in the case of the NLSO method with



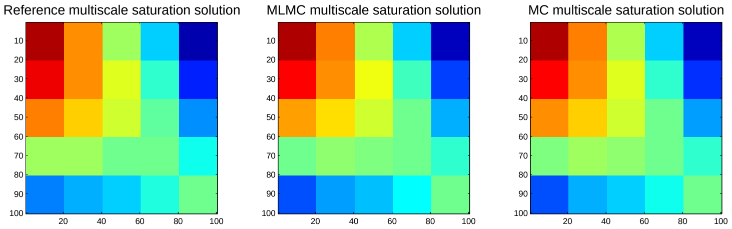
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

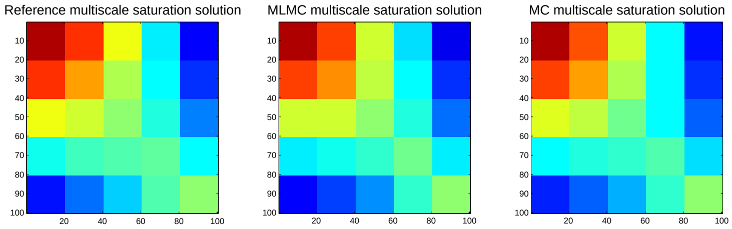


(c) Isotropic Exponential

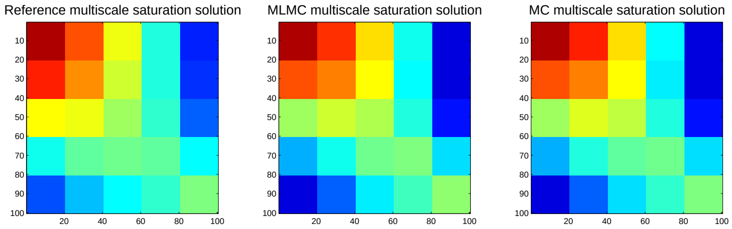


(d) Anisotropic Exponential

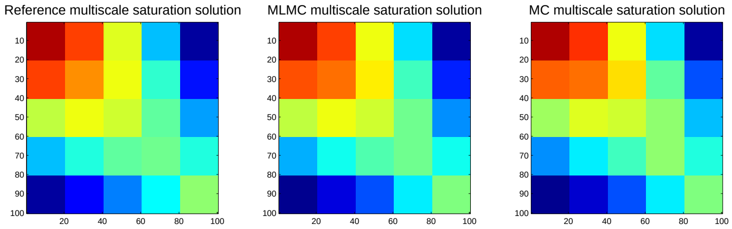
Figure 41: Water saturation for NLSO for MLMC and MC for the different methods and distributions for two-phase flow, $N_b = 100$.



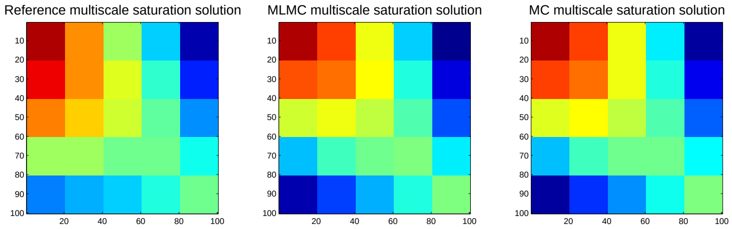
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

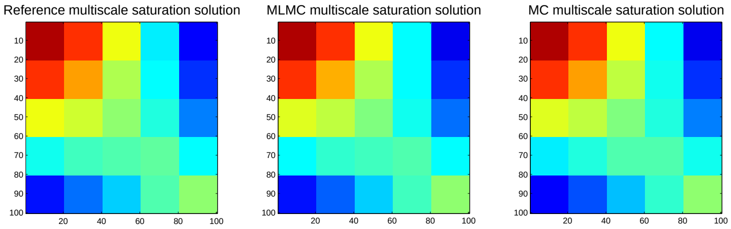


(c) Isotropic Exponential

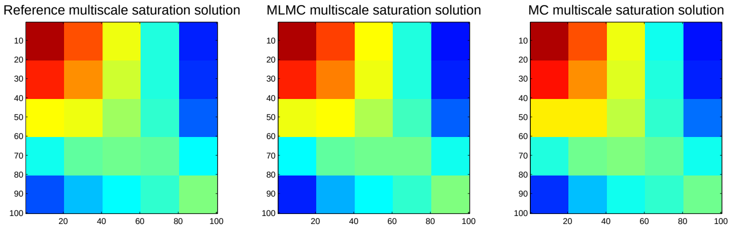


(d) Anisotropic Exponential

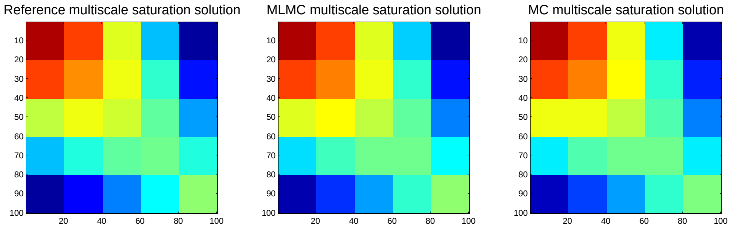
Figure 42: Water saturation for LSO for MLMC and MC for the different methods and distributions for two-phase flow.



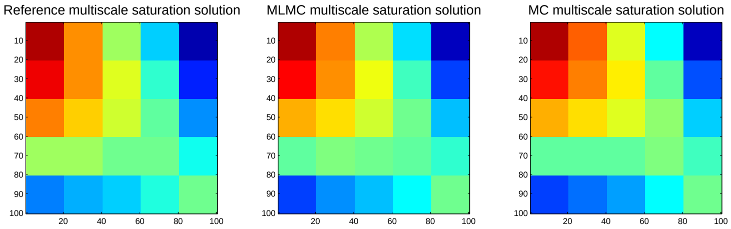
(a) Isotropic Gaussian



(b) Anisotropic Gaussian

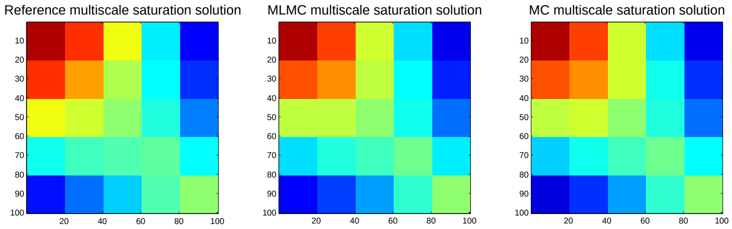


(c) Isotropic Exponential

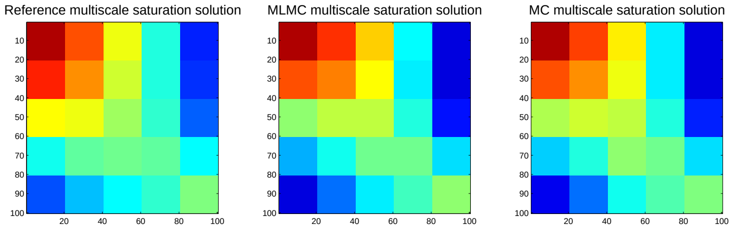


(d) Anisotropic Exponential

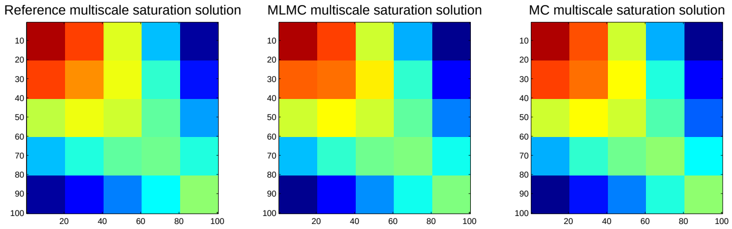
Figure 43: Water saturation for NLSO with POD for MLMC and MC for the different methods and distributions for two-phase flow.



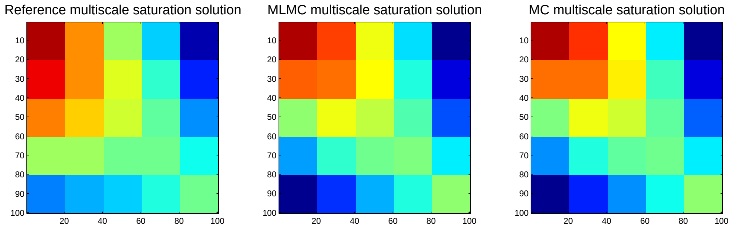
(a) Isotropic Gaussian



(b) Anisotropic Gaussian



(c) Isotropic Exponential



(d) Anisotropic Exponential

Figure 44: Water saturation for LSO with POD for MLMC and MC for the different methods and distributions for two-phase flow.

	isotropic Gaussian	anisotropic Gaussian	isotropic Exponential	anisotropic Exponential
(N_1, N_2, N_3)	$(3, 6, 12)$	$(3, 6, 12)$	$(24, 48, 96)$	$(24, 48, 96)$
(M_1, M_2, M_3)	$(70, 20, 10)$	$(70, 20, 10)$	$(70, 20, 10)$	$(70, 20, 10)$
$\widehat{N}$	12	12	96	96
$\widehat{M}$	20	20	20	20
M_{MCref}	500	500	500	500
N_{POD}	100	100	500	500
MLMC error NLSO $N_b = 100$	0.0566	0.0557	0.0537	0.0529
MC error NLSO $N_b = 100$	0.0970	0.0889	0.0943	0.0840
$\frac{\text{MC error}}{\text{MLMC error}}$ NLSO $N_b = 100$	1.71	1.60	1.76	1.59
MLMC error NLSO	0.0522	0.0497	0.0529	0.0543
MC error NLSO	0.0989	0.0849	0.0947	0.0898
$\frac{\text{MC error}}{\text{MLMC error}}$ NLSO	1.90	1.71	1.79	1.65
MLMC error LSO	0.0448	0.0854	0.0581	0.1161
MC error LSO	0.0839	0.1056	0.1039	0.1274
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	1.72	1.24	1.79	1.10
MLMC error POD NLSO	0.0479	0.0563	0.0568	0.0583
MC error POD NLSO	0.0884	0.0937	0.0972	0.0911
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	1.85	1.67	1.71	1.56
MLMC error POD LSO	0.0565	0.0843	0.0652	0.1299
MC error POD LSO	0.0992	0.1083	0.0952	0.1524
$\frac{\text{MC error}}{\text{MLMC error}}$ LSO	1.76	1.29	1.46	1.17

Table 26: Parameters and errors for the two-phase flow example.

POD for an isotropic Exponential distribution. For the LSO approach the corresponding errors are 0.8% and 1.0% for MLMC and MC, respectively.

21. Conclusions

In this work we consider multiscale problems with stochastic coefficients. We combine multiscale methods for deterministic problems, such as mixed multiscale finite elements and homogenization with stochastic methods, such as multi-level Monte Carlo methods and Karhunen-Loève expansions to increase the accuracy in comparison to a standard approach -the Monte Carlo method. Our objective is to rapidly compute the expectation of the macroscale quantities, such as macroscale solutions, homogenized coefficients, or functionals of these quantities.

Part I and Part II are devoted to the study of numerical homogenization with different stochastic methods. We consider elliptic stationary diffusion equations with stochastic coefficients which vary on the macroscale and the microscale.

In Part I we decouple the high-dimensional local problems with the help of the Karhunen-Loève expansion and a polynomial chaos approach. The gain in the computational work depends on the underlying distribution of the coefficient. In general, we cannot state our developed approach is more appropriate than Monte Carlo. Additionally, we introduce a method to speed-up the approximation of the eigenpairs of the covariance operator needed for the Karhunen-Loève expansion. This method is based on low-rank approximations of the matrix. Since the Karhunen-Loève expansion is widely used to approximate random fields the applicability of the low-rank approximation approach is not restricted to this part of the work.

In Part II we combine numerical homogenization with multi-level Monte Carlo methods. We consider different levels of coarse-grid meshes and representative volumes. We combine the results from a few expensive computations that involve smallest coarse meshes and largest representative volumes with many less expensive computations with larger coarse mesh and smaller representative volume sizes. The larger the coarse mesh or the smaller the representative volume size the more computations are used. We show that by selecting the number of realizations at each level carefully we can achieve a speed-up in the computations. For the computations of homogenized solutions, we propose weighted multi-level Monte Carlo where weights are chosen at each level such that it optimizes the accuracy at a given cost.

In Part III we consider multi-phase flow and transport equations. Here we combine multiscale finite element methods and multi-level Monte Carlo techniques to speed-up Monte Carlo simulations. In particular, we consider no-local-solve-online (NLSO) and local-solve-online (LSO) ensemble level mixed multiscale finite element approaches. We precompute for a few realizations of the random field multiscale basis functions. An ensemble of these basis functions is used to solve the multi-phase flow equation for an arbitrary realization. We show that NLSO provides better accuracy, since it is reasonable to choose local boundary conditions for the computation of the online multiscale basis in LSO.

Different sizes of the ensemble are related to different levels in the multi-level Monte Carlo approach. The use of larger ensembles yields more accurate solutions. We run more accurate (and expensive) forward simulations with fewer samples while less accurate (and inexpensive) forward simulations are run with a larger number of samples. Selecting the number of expensive and inexpensive simulations carefully, one can show that multi-level Monte Carlo can provide better accuracy at the same costs than Monte Carlo.

In this work we presented different methods to deal with the stochastic nature of the problems in combination with different approaches to handle the many scales. However, this gives only a small view insight the topic. A natural next step would be to consider a random mobility in the two-phase flow problem, for example. In Part III we have chosen the permeability realizations of the offline stage randomly or with proper orthogonal decomposition. Since we do POD with respect to the velocity, the influence on the saturation is of interest. Techniques as in reduced basis methods to select the samples and other POD approaches need to be studied. Of course, we have not considered any possible combination of multiscale approaches and stochastic methods. Another interesting case is to combine the multiscale methods with collocation methods.

A. Notation

Domains:

D	bounded domain, $D \subseteq \mathbb{R}^d$
∂D	smooth boundary of D
∂D^D	boundary part with Dirichlet boundary conditions, $\partial D^D \subseteq \partial D$
∂D^N	boundary part with Neumann boundary conditions, $\partial D^N \subseteq \partial D$
Y	periodicity cell, $Y = \{y = (y_1, \dots, y_d) : 0 < y_i < 1 \text{ for } i = 1, \dots, d\}$
Y_η	RVE, $Y_\eta = (0, \eta)^d$
Y_η^x	RVE, $Y_\eta^x = (x - \frac{\eta}{2}, x + \frac{\eta}{2})^d$

Spaces:

$L^p(D)$	L^p space
$L^p(\Omega, B)$	Bochner space with a Banach space B
$L^p_{loc}(\mathbb{R}^d, L^p(\Omega))$	$u \in L^p(\mathfrak{K}, L^p(\Omega)),\ \forall \mathfrak{K} \subset \mathbb{R}^d, \mathfrak{K}$ compact
$L^p_{unif}(\mathbb{R}^d, L^p(\Omega))$	$\lVert u \rVert_{L^p(B_x, L^p(\Omega))} \leq C,\ \forall x \in \mathbb{R}^d$, with C independent of x and the unit ball B_x with center x
$\mathcal{C}^0(\bar{D})$	space of continuous functions
$\mathcal{C}^1(\bar{D})$	space of continuously differentiable functions
$\mathcal{C}^t(\bar{D})$	Hölder space, $0 < t < 1$
$H^k(D)$	Sobolev space, $k \in \mathbb{N}$
$H^r(D)$	$r \notin \mathbb{N}$, $u \in H^k(D)$, such that $\lvert u \rvert^2_{H^r(D)} < \infty$, $r = k + t$, $k \in \mathbb{N}$, $0 < t < 1$
$H^p_0(D)$	$H^p_0(D) \subset H^p(D)$
$H^{-p}(D)$	dual space of the Sobolev space $H^p_0(D)$
$H(\text{div}, D)$	$H(\text{div}, D) = \{u : u \in (L^2(D))^d, \text{div } u \in L^2(D)\}$
$H_0(\text{div}, D)$	$H(\text{div}, D)$ with zero Neumann boundary

Norms:

$\lVert u \rVert_{L^p(D)}$	$\lVert u \rVert_{L^p(D)} = \begin{cases} \left(\int\limits_D \lvert u(x) \rvert^p \, dx \right)^{1/p}, & p < \infty \\ \text{esssup}_{x \in D} \lvert u(x) \rvert, & p = \infty \end{cases}$
$\lVert u \rVert_{L^p(\Omega, B)}$	$\lVert u \rVert_{L^p(\Omega, B)} = \begin{cases} \left(\int\limits_\Omega \lVert u \rVert^p_B \, dP(\omega) \right)^{1/p}, & p < \infty \\ \text{esssup}_{\omega \in \Omega} \lVert u \rVert, & p = \infty \end{cases}$
$\lVert u \rVert_{\mathcal{C}^0(\bar{D})}$	$\lVert u \rVert_{\mathcal{C}^0(\bar{D})} = \sup\limits_{x \in \bar{D}} \lvert u(x) \rvert$
$\lVert u \rVert_{\mathcal{C}^1(\bar{D})}$	$\lVert u \rVert_{\mathcal{C}^1(\bar{D})} = \sum\limits_{\lvert \alpha \rvert \leq 1} \lVert D^\alpha u \rVert_{\mathcal{C}^0(\bar{D})}$
$\lvert u \rvert_{\mathcal{C}^t(\bar{D})}$	$\lvert u \rvert_{\mathcal{C}^t(\bar{D})} = \sup\limits_{x, y \in \bar{D} : x \neq y} \frac{\lvert u(x) - u(y) \rvert}{\lvert x - y \rvert^t}$, $0 < t < 1$, semi-norm

$\lVert u\rVert_{\mathcal{C}^t(\bar{D})}$	$\lVert u\rVert_{\mathcal{C}^t(\bar{D})} = \sup_{x\in\bar{D}} \lvert u(x)\rvert + \lvert u\rvert_{\mathcal{C}^t(\bar{D})}$
$\lvert u\rvert_{H^k(D)}$	$\lvert u\rvert_{H^k(D)} = \left(\int_D \sum_{\lvert\alpha\rvert=k} \lvert D^\alpha u\rvert^2 \, dx \right)^{1/2}$, semi-norm
$\lVert u\rVert_{H^k(D)}$	$\lVert u\rVert_{H^k(D)} = \left(\int_D \sum_{\lvert\alpha\rvert\le k} \lvert D^\alpha u\rvert^2 \, dx \right)^{1/2}$
$\lvert u\rvert_{H^r(D)}$	$\lvert u\rvert_{H^r(D)} = \left(\iint_{D\times D} \sum_{\lvert\alpha\rvert=k} \frac{[D^\alpha u(x) - D^\alpha u(y)]^2}{\lvert x-y\rvert^{d+2t}} \, dx \, dy \right)^{1/2}$, semi-norm
$\lVert u\rVert_{H^r(D)}$	$\lVert u\rVert_{H^r(D)} = \left(\lVert u\rVert^2_{H^k(D)} + \lvert u\rvert^2_{H^r(D)} \right)^{1/2}$
$\lVert u\rVert$	$\lVert u\rVert = \lVert u\rVert_{L^2(\Omega,H)}$, with an appropriate Hilbert space H
$\vert\!\vert\!\vert u\vert\!\vert\!\vert$	$\vert\!\vert\!\vert u\vert\!\vert\!\vert = \lVert u\rVert_H$, with an appropriate Hilbert space H
$\lvert x\rvert$	$\lvert x\rvert = \sqrt{\sum_{i=1}^{n} x_i^2}$, $x \in \mathbb{R}^n$ for some $n \in \mathbb{N}$, e.g., $n = 1$ or $n = d$

Stochastics:

(Ω, F, P)	probability space
$E[G]$	expected value of a random variable G, $E[G] = \int_\Omega G(\omega)\, dP(\omega)$
E_K	expected value of the coefficient K, $E_K(y) = E[K(y)]$
μ	expected value of a Gaussian distributed random variable
μ_{log}	expected value of a lognormal distribution, $\mu_{log} = \exp(\mu + \frac{\sigma^2}{2})$
$\mathrm{Cov}(G_1, G_2)$	covariance of two random variables G_1 and G_2, $\mathrm{Cov}(G_1, G_2) = E[(G_1 - E[G_1])(G_2 - E[G_2])]$
$\mathrm{Var}(G)$	variance of a random variable G, $\mathrm{Var}(G) = \mathrm{Cov}(G, G)$
$\mathrm{std}(G)$	standard deviation of a random variable G, $\mathrm{std}(G) = \sqrt{\mathrm{Var}(G)}$
σ	standard deviation of the normal distribution
Σ	standard distribution of the lognormal distribution, $\Sigma^2 = \mu^2_{log} \left(\exp(\sigma^2) - 1\right)$
$\mathrm{cov}(y, y')$	covariance function of the coefficient K, $\mathrm{cov}(y, y') = \int_\Omega K(y,\omega) K(y',\omega)\, dP(\omega) - E_K(y) E_K(y') = \mathrm{Cov}(K(y,\omega), K(y',\omega))$
cov_G	Gaussian covariance, $\mathrm{cov}_G(y, y') = \sigma^2 \exp\left(\frac{-\lvert y-y'\rvert^2}{2\epsilon^2}\right)$
cov_{log}	covariance of lognormal distribution, $\mathrm{cov}_{log}(y, y') = \mu^2_{log} \left(\exp(\mathrm{cov}_G(y, y')) - 1\right)$
ϵ	correlation length
τ	$\tau = \sqrt{2}\epsilon$

Homogenization:

K	microscale coefficient
K^*	exact effective coefficient $[K^*]_{ij} = \lim\limits_{m\to\infty} \frac{1}{(2m)^d} \int_{\lvert y\rvert\leq m}(e_i + \nabla\chi_i(y,\cdot))^T K(y,\omega)e_j\,dy$
K^*_η	effective coefficient calculated in the RVE Y^x_η, (4.13), $e_j \cdot K^*_\eta(x,\omega)e_i = \frac{1}{\eta^d}\int_{Y^x_\eta} \nabla\chi_j \cdot K(x,\frac{y}{\epsilon},\omega)\nabla\chi_i$
χ_i	solution of the corresponding cell problem (cf. (4.8), (4.12)) in i^{th} direction
u_ϵ	solution of the microscale problem
u^*	solution of the homogenized equation
$\hat{u}$	reconstructed fine scale solution
u^{edge}	solution of a PDE on ∂V, needed as boundary condition for reconstruction

A.1. Notation Part I

MC:

N	number of realizations used for MC, $1 \leq n \leq N$
X_n	random variable with $E[X_n] = 0$ and given covariance
K^{MC}_n	microscale coefficient in Monte Carlo simulation, $K^{MC}_n(y) = E_K(y) + X_n$
K^{*MC}_n	effective tensor Monte Carlo simulation, $\left(K^{*MC}_n\right)_{ij} = \int_Y \nabla\chi^{MC,n}_i \cdot K^{MC}_n(y)\nabla\chi^{MC,n}_j\,dy$
K^{*MC}_N	Monte Carlo approximation of the effective coefficient, (5.2), $K^{*MC}_N = \frac{1}{N}\sum_{n=1}^N K^{*MC}_n$
k_N	$k_N = \left((K^{*MC}_N)_{11}, (K^{*MC}_N)_{22}\right)^T$
$\chi^{MC,n}_i$	solution of the Monte Carlo cell problem (5.1) in i^{th} direction

KL:

K^{KL}	Karhunen-Loève expansion of $K(y,\omega)$, (5.3), $K^{KL}(y,\omega) = E_K(y) + \sum_{m=1}^\infty \sqrt{\lambda_m}\phi_m(y)X_m(\omega)$
K^{KL}_M	truncated Karhunen-Loève expansion, (5.6), $K^{KL}_M(y,\omega) = E_K(y) + \sum_{m=1}^M \sqrt{\lambda_m}\phi_m(y)X_m(\omega)$
$\tilde{K}^{KL}_M$	polynomial chaos expansion, (5.7) $\tilde{K}^{KL}_M(y,z) = E_K(y) + \sum_{m=1}^M \sqrt{\lambda_m}\phi_m(y)z_m$, coefficient of the cell problems (5.8) and (5.9)
$K^{KL,\mathbf{j}}_M$	$K^{KL,\mathbf{j}}_M(y) = E_K(y) + \sum_{m=1}^M \sqrt{\lambda_m}\phi_m(y)\mu^{r_m}_{j_m}$, (5.13) coefficient in the decoupled Karhunen-Loève cell problem (5.12)
$\bar{K}^{KL,\mathbf{j}}_M$	$\bar{K}^{KL,\mathbf{j}}_M = E_K(y) + \sum_{m=1}^M \sqrt{\lambda_m}\phi_m(y)\frac{\int_I P^{r_m}_{j_m}(z_m)z_m\,d\rho_m(z_m)}{\int_I P^{r_m}_{j_m}(z_m)\,d\rho_m(z_m)}$, (5.14)
$\tilde{K}^{KL}_{\det}$	$\tilde{K}^{KL}_{\det} = \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y \nabla\tilde{\chi}^{KL}_{l,\mathbf{j}}(y) \cdot K^{KL,\mathbf{j}}_M(y)\nabla\tilde{\chi}^{KL}_{k,\mathbf{j}}(y)\,dy$

$\hat{K}_{\text{det}}^{KL}$	$\hat{K}_{\text{det}}^{KL} = \sum_{\mathbf{j}\leq\mathbf{r}} \int_Y \nabla\hat{\chi}_{l,\mathbf{j}}^{KL}(y) \cdot K_M^{KL,\mathbf{j}}(y)\nabla\hat{\chi}_{k,\mathbf{j}}^{KL}(y)\,dy$
K^{*KL}	effective coefficient of (5.8), $e_l K^{*KL}(z) e_k = \int_Y \nabla\chi_l^{KL}\tilde{K}_M^{KL}(y,z)\nabla\chi_k^{KL}\,dy$
$K_{KL}^{*\mathbf{j}}$	effective coefficient of (5.22), $\left(K_{KL}^{*\mathbf{j}}\right)_{lk} = \int_Y \nabla\tilde{\chi}'^{KL}_{l,\mathbf{j}}\tilde{K}_M^{KL,\mathbf{j}}(y,z)\nabla\tilde{\chi}'^{KL}_{k,\mathbf{j}}\,dy$
K_{KL}^{*}	$K_{KL}^{*} = \frac{\sum_{\mathbf{j}\leq\mathbf{r}} K_{KL}^{*\mathbf{j}}}{\lvert\mathbf{r}\rvert}$, (5.23)
$\chi_i^{KL}(y,z)$	solution of the Karhunen-Loève cell problem (5.8)
$\hat{\chi}_i^{KL}(y,z)$	solution of the Karhunen-Loève cell problem (5.9), $\hat{\chi}_i^{KL} = \chi_i^{KL} - y_i$
$\tilde{\chi}_{i,\mathbf{j}}^{KL}$	solution of the decoupled Karhunen-Loève cell problem (5.12)
$\hat{\chi}_{i,\mathbf{j}}^{KL}$	$\hat{\chi}_{i,\mathbf{j}}^{KL}(y) = \tilde{\chi}_{i,\mathbf{j}}^{KL}(y) + y \cdot e_i$, (5.21)
$\tilde{\chi}'^{KL}_{i,\mathbf{j}}$	solution of modified decoupled Karhunen-Loève cell problem (5.22)
Eigenproblem:	
$X_m(\omega)$	centered at 0, pairwise uncorrelated random variables in (5.3)
$(\lambda_m, \phi_m)_{1\leq m\leq\infty}$	sequence of eigenpairs of the covariance operator $(\mathcal{K}\phi)(y) = \int_Y \operatorname{cov}(y,y')\phi(y')\,dy'$ in (5.3)
$(\lambda_m^h, \phi_m^h)_{1\leq m\leq\tilde{N}}$	sequence of eigenpairs of the discrete eigenproblem, (5.4)
$\varphi_n \in S_h$	$1 \leq n \leq \tilde{N}$ basis of finite dimensional space S_h
Φ_m	coefficient vector of $\phi_m^h(y) = \sum_{j=1}^{\tilde{N}} \Phi_m^j \varphi_j(y)$
$\tilde{\Phi}_m$	$\tilde{\Phi}_m = \mathbf{M}^{1/2}\Phi_m$
$\mathbf{K}$	$(\mathbf{K})_{ij} = \int_{Y\times Y} \operatorname{cov}(y,y')\varphi_j(y')\varphi_i(y)\,dy'dy$
$\tilde{\mathbf{K}}$	$\tilde{\mathbf{K}} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$
$\mathbf{M}$	$(\mathbf{M})_{ij} = \int_Y \varphi_j(y)\varphi_i(y)\,dy$
Decoupling:	
$\mathcal{P}_r$	space of polynomials of degree at most r, $\mathcal{P}_r = \operatorname{span}\{1, t, t^2, \dots, t^r\} \subseteq L^2(I)$
$\mathbf{r}$	$\mathbf{r} = (r_1, r_2, \dots, r_M) \in \mathbb{N}_0^M$
$\left(\mu_j^{r_m}, P_j^{r_m}\right)_{0\leq j\leq r_m}$	eigenpairs of the symmetric bilinear form $(u,v) \to \int_{-1/2}^{1/2} u(t)v(t)t\,d\rho_m(t)$ over $\mathcal{P}_{r_m}$
ρ	$\rho := \rho_1 \times \rho_2 \times \cdots \rho_M$, $\rho_m(\mathcal{B}) := P(X_m \in \mathcal{B})$ for any Borel set $\mathcal{B} \subseteq I$
$\{\psi_j\}_{0\leq j\leq r_m}$	basis of $\mathcal{P}_{r_m}$
$\mathbf{A}$	$(\mathbf{A})_{ij} = \int_I \psi_i(t)\psi_j(t)t\,d\rho_m(t)$
$\mathbf{B}$	$(\mathbf{B})_{ij} = \int_I \psi_i(t)\psi_j(t)\,d\rho_m(t)$
$P_{\mathbf{j}}^{\mathbf{r}}$	$P_{\mathbf{j}}^{\mathbf{r}}(z) = P_{j_1}^{r_1}(z_1)\,P_{j_2}^{r_2}(z_2)\cdots P_{j_M}^{r_M}(z_M) \quad \text{for } \mathbf{j} \leq \mathbf{r}$
$\mathcal{P}_{\mathbf{r}}$	$\mathcal{P}_{\mathbf{r}} = \mathcal{P}_{r_1} \otimes \mathcal{P}_{r_2} \otimes \cdots \mathcal{P}_{r_M}$, $\mathcal{P}_{\mathbf{r}} = \operatorname{span}\{P_{\mathbf{j}}^{\mathbf{r}} \;\vert \mathbf{j} \leq \mathbf{r}\} \subseteq L^2(I^M)$
$\bar{P}_{\mathbf{j}}^{\mathbf{r}}$	$\bar{P}_{\mathbf{j}}^{\mathbf{r}} = \int_{I^M} P_{\mathbf{j}}^{\mathbf{r}}(z)\,d\rho(z) = \prod_{m=1}^{M} \int_I P_{j_m}^{r_m}(z_m)\,d\rho_m(z_m)$, (5.15)
$f_{i,\mathbf{j}}^{KL}$	$f_{i,\mathbf{j}}^{KL}(y) = \bar{P}_{\mathbf{j}}^{\mathbf{r}} \operatorname{div}\left(e_i \cdot \bar{K}_M^{KL,\mathbf{j}}(y)\right)$, (5.16), right-hand side of the decoupled Karhunen-Loève cell problem (5.12)

$\bar{E}_K$	$\bar{E}_K = \int_Y E_K(y)\,dy$

H-matrices:

T	$T := (\mathcal{N}, r, S)$ tree with nodes $\mathcal{N}$ and root$(T) = r$, S mapping from $\mathcal{N}$ into subsets of $\mathcal{N}$
T_L	$T := (\mathcal{N}, r, S, m, L)$ labeled tree, L label set of T, $m : \mathcal{N} \mapsto L$
$T_{\mathcal{I}\times\mathcal{J}}$	block cluster tree
$\hat{t}$	$\hat{t} = m(t) \in \mathcal{I}$ label of $t \in \mathcal{N}$
$\mathcal{L}(T)$	$\mathcal{L}(T) := \{t \in \mathcal{N} \,:\, \text{sons}(t) = \emptyset\}$ set of leaves
sons$^*(t)$	set of descendants of a node $t \in \mathcal{N}$, eq. (5.25)
$\mathbf{K}'$	low-rank approximation of $\mathbf{K}$, eq. (5.29)
$\tilde{\mathbf{R}}$	$\tilde{\mathbf{R}} = \mathbf{A}\mathbf{B}^T \in \mathbb{R}^{\hat{t}\times\hat{s}}$ and $\mathbf{A} \in \mathbb{R}^{\hat{t}\times k}$ and $\mathbf{B} \in \mathbb{R}^{\hat{s}\times k}$, $k \in \mathbb{N}$

Finite Volume:

C_i	cell of the grid with number i
∂C_i	boundary of cell C_i
γ_{ij}	face common to C_i and C_j
ν	outer normal
ν_{ij}	normal pointing from C_i to C_j
A	matrix of the finite volume scheme
F	right-hand side of the finite volume scheme
U	$U_i = u(\bar{x}_i)$ solution vector of the finite volume scheme
V_i	volume of cell i
v_{ij}	volume of γ_{ij} (or boundary face)
$\bar{x}_i$	cell center of cell i
$\bar{x}_{ij}$	center of the intersection
x_i^{ul}	upper left node of cell C_i
x_i^{ur}	upper right node of cell C_i
x_i^{bl}	bottom left node of cell C_i
x_i^{br}	bottom right node of cell C_i
h_j	$h_j = \sum_{k=1}^{2} \lvert(\bar{x}_i)_k - (\bar{x}_j)_k\rvert$
h_{ij}	$h_{ij} = \sum_{k=1}^{2} \lvert(\bar{x}_i)_k - (\bar{x}_{ij})_k\rvert$
N_i	set of the global numbers of the neighbors of cell i
B_i^D	index set of Dirichlet boundary condition intersections
B_i^N	index set of Neumann boundary condition intersections
B_i	$B_i = B_i^D + B_i^N$
$K(\bar{x}_{ij})$	$K(\bar{x}_{ij}) = \frac{2}{\frac{1}{K(\bar{x}_i)} + \frac{1}{K(\bar{x}_j)}}$
n_x	number of cells in x-direction

A.2. Notation Part II

MLMC:

L	level of interest
l	level, $1 \leq l \leq L$
η_l	RVE size at level l, $\eta_1 < \eta_2 < \cdots < \eta_L$
$\hat{\eta}$	RVE size used for MC, normally $\hat{\eta} = \eta_L$
H_l	coarse mesh size at level l, $H_1 > H_2 > \cdots > H_L$

$\hat{H}$	coarse mesh used for MC, normally $\hat{H} = H_L$
h_l	fine mesh size at level l, $h_1 > h_2 > \cdots > h_L$
m_l	number of realizations used at level l with RVE size η_l, $m_1 > m_2 > \cdots > m_L$
m_{lj}	number of realizations used with RVE size η_l and fine mesh size h_j
$\hat{m}$	number of realizations used for MC with RVE size η_L
M_l	number of realizations used at level l with mesh size H_l, $M_1 > M_2 > \cdots > M_L > M_{L+1} = 0$
$\widetilde{M_l}$	$\widetilde{M_l} = M_l - M_{l+1}$
$\hat{M}$	number of realizations used for MC with mesh size H_L
M_{ref}	number of realizations used to calculate the reference
$\mathfrak{m}$	$\mathfrak{m} = (m_1, m_2, \cdots, m_L)$
$\mathfrak{M}$	$\mathfrak{M} = (M_1, M_2, \cdots, M_L)$
$\mathfrak{H}$	$\mathfrak{H} = (H_1, H_2, \cdots, H_L)$
$\mathcal{P}_l$	set of coarse grid points where we solve the RVE problems, $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \cdots \subset \mathcal{P}_L$
P_l	number of RVE problem in each direction, $P_l = \lvert\mathcal{P}_l\rvert$
G_l	quantity of interest at level l, e.g., effective coefficient, upscaled solution
G_l^i	realization of G_l, $G_l^i = G_l(\omega_i)$
$E^L(G_L)$	MLMC approximation of $E[G_L]$, $E^L(G_L) = \sum_{l=1}^{L} E_{M_l}(G_l - G_{l-1})$ (10.1)
$E^L_{same}(G_L)$	$E^L_{same}(G_L) = \sum_{l=1}^{L} \frac{1}{M_l} \sum_{j=1}^{M_l} (G_l - G_{l-1}) = E^L(G_L)$
$E^L_{ind}(G_L)$	$E^L_{ind}(G_L) = \sum_{l=1}^{L} \frac{1}{\widetilde{M_l}} \sum_{j=M_{l+1}+1}^{M_l} (G_l - G_{l-1})$
$E_{M_l}(G_l)$	arithmetic mean, $E_{M_l}(G_l) = \frac{1}{M_l} \sum_{i=1}^{M_l} G_l^i$
$e_{MLMC}(G_L)$	$e_{MLMC}(G_L) = \sqrt{E\left[\vert\!\vert\!\vert E[G_L] - E^L(G_L)\vert\!\vert\!\vert^2\right]}$, (10.2)
$e^{ref}_{MLMC}(G_L)$	$e^{ref}_{MLMC}(G_L) = \frac{e_{MLMC}(G_L)}{\vert\!\vert\!\vert E[G_L]\vert\!\vert\!\vert}$
$e^{same}_{MLMC}(G_L)$	$e^{same}_{MLMC}(G_L) = e_{MLMC}(G_L)$
$e^{ind}_{MLMC}(G_L)$	$e^{ind}_{MLMC}(G_L) = \sqrt{E\left[\vert\!\vert\!\vert E[G_L] - E^L_{ind}(G_L)\vert\!\vert\!\vert^2\right]}$
$e_{MC}(G_L)$	$e_{MC}(G_L) = \sqrt{E\left[\vert\!\vert\!\vert E[G_L] - E_{\hat{M}}(G_L)\vert\!\vert\!\vert^2\right]}$, (10.3)
$e^{ref}_{MC}(G_L)$	$e^{ref}_{MC}(G_L) = \frac{e_{MC}(G_L)}{\vert\!\vert\!\vert E[G_L]\vert\!\vert\!\vert}$
Er_{MLMC}	$Er_{MLMC} = \frac{1}{Nb} \sum_{j=1}^{Nb} [e^{rel}_{MLMC}(K^*_L(\omega_j))]^2$
Er_{MC}	$Er_{MC} = \frac{1}{Nb} \sum_{j=1}^{Nb} [e^{rel}_{MC}(K^*_L(\omega_j))]^2$
K^*_l	effective coefficient with RVE size η_l, $K_l := K_{\eta_l}$ (cf. (4.13))
K^*_{l,h_j}	effective coefficient with RVE size η_l and fine mesh size h_j
$K^*_{\eta_l,H_j}$	effective coefficient with RVE $Y^x_{\eta_l}$ with $x \in \mathcal{P}_j$
K^*_{ref}	reference effective coefficient
δ_l	$\delta_l = \sqrt{E\left[\vert\!\vert\!\vert K^* - K^*_l\vert\!\vert\!\vert^2\right]} \leq C\left(\frac{\epsilon}{\eta_l}\right)^{\frac{\beta}{2}}$

$\delta_{l,j}$	$\delta_{l,j} = \sqrt{E\left[\left\|\left\|K^* - K^*_{l,h_j}\right\|\right\|^2\right]} \leq C\sqrt{\left(\frac{\epsilon}{\eta_l}\right)^\beta + \left(\frac{h_j}{\epsilon}\right)^\gamma}$
β	convergence rate of the effective coefficient w.r.t the RVE size
γ	convergence rate of the effective coefficient w.r.t the fine mesh size
N_l	costs to solve the RVE problem (4.12) in all directions
W^{MLMC}_{RVE}	costs for MLMC concerning the RVE problems
W^{MC}_{RVE}	costs for MC concerning the RVE problems
W^{MLMC}_{coarse}	costs for MLMC concerning the coarse grid problems
W^{MC}_{coarse}	costs for MC concerning the coarse grid problems
Cor_{K^*}	two-point correlation function of the homogenized matrix, $Cor_{K^*}(x,y) = E\left([K^*(x,\omega)]_{ij}[K^*(y,\omega)]_{qp}\right)$
$Cor_{m_l}(K^*_l)$	empirical estimator of the two-point correlation, $Cor_{m_l}(K^*_l) = \frac{1}{m_l}\sum_{k=1}^{m_l}[K_l^{*,k}(x)]_{ij}[K_l^{*,k}(y)]_{qp}$
$Cor^L(K^*_L)$	MLMC approximation of the two-point correlation function, $Cor^L(K^*_L) = \sum_{l=1}^L\left(Cor_{m_l}(K^*_l) - Cor_{m_l}(K^*_{l-1})\right)$
w_l	weight at level l, $\sum_{l=1}^L = 1$
α_l	$\alpha_l = \sum_{i=l}^L w_l$
u^*	exact homogenized solution (no error), cf. (13.2)
$u^*_{H_l}$	homogenized solution with mesh size H_l (discretization error), (13.6)
u_l	homogenized solution with mesh size H_l and the coefficient calculated with RVE size η_l (discretization and homogenization error), (13.5)
u_{η_j,H_i}	homogenized solution with mesh size H_i and the coefficient calculated with RVE size η_j
$\tilde{u}$	$\tilde{u} = \sum_{l=1}^L \frac{\alpha_l}{M_l}\sum_{j=l}^L(M_j - M_{j+1})(u_{\eta_j,H_l} - u_{\eta_j,H_{l-1}})$, (13.7)
$E^{L*}(\tilde{u})$	weighted MLMC approximation, $E^{L*}(\tilde{u}) = \sum_{l=1}^L \alpha_l E^*_{M_l}(u_l - u_{l-1})$, (13.8)
$E^*_{M_l}(u_l - u_{l-1})$	$= \frac{1}{M_l}\sum_{j=l}^L(M_j - M_{j+1})E_{M_j - M_{j+1}}(u_{\eta_j,H_l} - u_{\eta_j,H_{l-1}})$
$E^i_{M^{ref},L}$	$E^i_{M^{ref},L} = \frac{1}{L}\sum_{l=1}^L \frac{1}{M^{ref}}\sum_{k=1}^{M^{ref}} u_l^k$
$e^i_{MLMC}(u_L)$	$e^i_{MLMC}(u_L) = \frac{\\|E^i_{M^{ref},L}(u_L) - E^{i,L}(u_L)\\|_{L^2(D)}}{\\|E_{M^{ref},L}(u_L)\\|_{L^2(D)}}$
$e^i_{MC}(u_L)$	$e^i_{MC}(u_L) = \frac{\\|E^i_{M^{ref},L}(u_L) - E^i_{\hat{M}}(u_L)\\|_{L^2(D)}}{\\|E_{M^{ref},L}(u_L)\\|_{L^2(D)}}$

A.3. Notation Part III

Two-phase flow:

index $_{w}$, $_{o}$	quantity of water or oil phase
index $_{\alpha}$	phase index, $\alpha \in \{w, o\}$
ϕ	rock porosity, $\phi := \frac{\text{void volume}}{\text{medium volume}}$,
ϕ_α	porosity of phase α, $\phi_\alpha := \frac{\text{void volume filled with phase } \alpha}{\text{medium volume}}$
S_α	saturation of phase α, $S_\alpha := \frac{\phi_\alpha}{\phi}$
S	water saturation, solution of eq. (15.7)
k	permeability

$k_{\min}$	coercivity constant
$k_{\max}$	bounded constant
$k_{r\alpha}$	relative permeability of phase α
p_α	pressure of phase α
p_c	capillary pressure, $p_c = p_o - p_w$
p	pressure, solution of eq. (15.6)
ρ_α	density of phase α
μ_α	viscosity of phase α
q_α	source ($q_\alpha > 0$) or sink ($q_\alpha < 0$) of phase α
q	total source, $q = q_w + q_o$
G	gravitational pull-down force
g	gravitational constant
v_α	phase velocity, $v_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} k\,(\nabla p_\alpha - \rho_\alpha G)$, eq. (15.2)
v	total velocity, $v = v_w + v_o$
t	time, $t \in [0, T]$
λ_α	phase mobility, $\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}$
λ	total mobility, $\lambda = \lambda_w + \lambda_o$
f	flux term, $f = \frac{\lambda_w}{\lambda}$

Mixed MsFEM:

V_h	finite dimensional velocity approximation space, $V_h \subset H(\mathrm{div}, D)$
V_h^0	$V_h^0 = V_h \cap H_0(\mathrm{div}, D)$
$V_h(k_j)$	$V_h(k_j) = \bigoplus_{\iota \in \mathcal{I}} \{\Psi_{\iota, k_j}\}$
$V_h^0(k_j)$	$V_h^0(k_j) = V_h(k_j) \cap H_0(\mathrm{div}, D)$
V_h^{NLSO}	velocity approximation space no-local-solve-online (NLSO) method, $V_h^{NLSO} = \bigoplus_{j=1}^{N_l} V_h(k_j)$
V_h^ι	$V_h^\iota = \bigoplus_{j=1}^{N_l} \Psi_{\iota, k_j}$
V_h^{LSO}	velocity approximation space local-solve-online (LSO) method, $V_h^{LSO} = \bigoplus_{\iota \in \mathcal{I}} \{\Psi_{\iota, \tilde{k}}\}$
Q_h	finite dimensional pressure approximation space, $Q_h \subset L^2(D)/\mathbb{R}$
function index $_h$	projection onto the corresponding finite dimensional subspace
D_i	partition of D in polyhedral elements, $\bigcup_{i=1}^N D_i = D$
Γ_ι	interface between two neighboring blocks, $\Gamma_\iota = D_i \cap D_{i'}$, $\iota = \{ii'\} \in \mathcal{I}$
N	number of blocks D_i, $1 \leq i \leq N$
N_l	number of permeability realizations $k_j(x)$, $1 \leq j \leq N_l$, per level used to build the velocity approximation space
$\mathcal{I}$	multi index set of index pairs of two neighboring blocks, i.e if $D_i \cap D_{i'} \neq \emptyset$, $\iota = \{ii'\} \in \mathcal{I}$
$\|\mathcal{I}\|$	cardinality of $\mathcal{I}$
n	outer normal

$n_{ii'}$	normal pointing from D_i to $D_{i'}$
g	Neumann boundary function in (15.8)
g_ι	boundary function of the auxiliary problem (15.10), e.g., global boundary condition: $g_\iota(k_j) = \frac{v_j \cdot n_{ii'}}{\int_{\Gamma_\iota} v_j \cdot n_{ii'} ds}$ ((15.11))
v_j	global velocity solution which solves (15.8) with the coefficient $k_j(x)$
k_j	realization of the random permeability for some random ω_j, $k_j(x) = k(x, \omega_j)$
K^j_{off}	given set of permeability realizations to compute the multiscale basis functions (cf. (15.10)) $K^j_{off} = (k^j_1, \cdots, k^j_{N_L})$, $1 \leq j \leq Nb$
K^j_{on}	set of permeability realizations to solve online the flow and transport equations (cf. (15.6)-(15.7)) to compute the saturation, $K^j_{on} = (k^j_1, \cdots, k^j_{M_1})$, $1 \leq j \leq Nb$
$\tilde{k}$	realization of the random permeability for some random $\tilde{\omega}$, $\tilde{k}(x) = k(x, \tilde{\omega})$
$w_{\iota,k}$	solution of the auxiliary problem (15.10) with the coefficient $k(x)$
$\tilde{w}_{\iota,\tilde{k}}$	approximation of the solution of (15.10) with local boundary conditions with the coefficient $\tilde{k}(x)$ in V^ι_h
$\Psi_{\iota,k}$	multiscale basis function $\Psi_{\iota,k} = -k(x)\nabla w_{\iota,k}$
$\tilde{\Psi}_{\iota,\tilde{k}}$	LSO basis multiscale basis function $\tilde{\Psi}_{\iota,\tilde{k}} = -\tilde{k}(x)\nabla \tilde{w}_{\iota,\tilde{k}}$

Saturation scheme:

t^k	discrete time value
Δt	time step, $\Delta t := t^{k+1} - t^k$
$S^k(x)$	water saturation at t^k, $S^k(x) = S(x, t^k)$
$S_i(t)$	time dependent coefficient in $S(x,t) = \sum_{i=1}^N S_i(t)\varphi_i(x)$
S^k_i	$S_i(t^k)$
$\mathcal{S}$	$\mathcal{S} = (S^{k+1}_i)_{1 \leq i \leq N}$
$\mathcal{S}^n$	Newton iteration
φ_i	piecewise constant basis function in space, $\varphi_i(x) = \begin{cases} 1, & \text{if } x \in D_i \\ 0, & \text{else.} \end{cases}, 1 \leq i \leq N$
θ	parameter in θ-scheme (15.12), $0 \leq \theta \leq 1$, $\theta = 0$: explicit scheme, $\theta = 1$: implicit scheme
Q_i	source in one cell D_i, $Q_i = \int_{D_i} q_w$
$\hat{f}(S)_{ii'}$	numerical flux, $\hat{f}(S)_{ii'} = \begin{cases} f(S_i), & \text{if } v \cdot n_{ii'} \geq 0 \\ f(S_{i'}), & \text{if } v \cdot n_{ii'} < 0 \end{cases}$

$v_{ii'}$	velocity flux across an intersection, $v_{ii'} = \int_{\Gamma_{ii'}} v \cdot n_{ii'}$
$H(\mathcal{S})$	$H(\mathcal{S}) = \left(S_i^k - \mathcal{S}_i - \frac{\Delta t}{\lvert D_i \rvert \phi} \sum_{\Gamma_{ii'}} \left(\hat{f}(\mathcal{S})_{ii'} v_{ii'}\right) + \frac{\Delta t}{\lvert D_i \rvert \phi} Q_i\right)_{1 \leq i \leq N}$
$D_H(\mathcal{S})$	Jacobian matrix of $H(\mathcal{S})$

MLMC:

L	level of interest
l	level, $1 \leq l \leq L$
M_l	number of realizations used at level l with N_l permeability realizations to compute the velocity approximation space, $M_1 > M_2 > \cdots > M_L$
$\hat{M}$	number of realizations used for MC with N_L permeability realizations
M_{ref}	number of realizations used to calculate a reference saturation
$\mathfrak{M}$	$\mathfrak{M} = (M_1, M_2, \cdots, M_L)$
S_l	water saturation at level l, i.e., N_l permeability realizations are used to compute the velocity approximation space
$S_{l,m}$	realization of the water saturation at level l, $S_{l,m} = S_l(x, t, \omega_m)$, $1 \leq m \leq M_l$
S_{ref}	reference saturation, arithmetic mean for M_{ref} realizations
$E^L(S_L)$	MLMC approximation of $E[S_L]$, $E^L(S_L) = \sum_{l=1}^{L} E_{M_l}(S_l - S_{l-1})$
S^j_{MLMC}	MLMC approximation of the expectation for given sets of permeability realization K^j_{off} and K^j_{on}, $E^L(S_L; K^j_{off}, K^j_{on})$
β	convergence rate, $\lVert S_l - S \rVert \sim \frac{1}{N_l^{\beta}}$
W_{MLMC}	computational costs for MLMC based on pressure
W_{MC}	computational costs for MC based on pressure
MLMC error	MLMC error $= \sqrt{\frac{1}{N_b} \sum_{j=1}^{N_b} \frac{\lVert S_{ref} - S^j_{MLMC} \rVert^2_{L^2}}{\lVert S_{ref} \rVert^2_{L^2}}}$
$Y(x, \omega)$	$Y(x, \omega) = \log[k(x, \omega)]$
H_j	coarse mesh size
ϵ_i	correlation length in direction i

B. Tables

σ	**Ansatz**	**coefficient**	**# cell problems in each direction**
0.0001	KL	$K^* = \begin{pmatrix} 0.973177 & -3.89458 \cdot 10^{-9} \\ -3.89458 \cdot 10^{-9} & 0.973177 \end{pmatrix}$	2
	MC	$K^* = \begin{pmatrix} 0.973179 & 1.00662 \cdot 10^{-8} \\ 3.18901 \cdot 10^{-8} & 0.973179 \end{pmatrix}$	100
	MC	$K^* = \begin{pmatrix} 0.973202 & -8.79283 \cdot 10^{-7} \\ -3.95239 \cdot 10^{-9} & 0.973201 \end{pmatrix}$	1
0.001	KL	$K^* = \begin{pmatrix} 0.973177 & \cdot 1.0132610^{-8} \\ 1.01326 \cdot 10^{-8} & 0.973177 \end{pmatrix}$	2
	MC	$K^* = \begin{pmatrix} 0.9732 & 9.64221 \cdot 10^{-8} \\ 4.06617 \cdot 10^{-7} & 0.9732 \end{pmatrix}$	100
	MC	$K^* = \begin{pmatrix} 0.97343 & -9.80544 \cdot 10^{-6} \\ 5.78373 \cdot 10^{-7} & 0.973411 \end{pmatrix}$	1
0.01	KL	$K^* = \begin{pmatrix} 0.973176 & -2.35257 \cdot 10^{-9} \\ -2.35257 \cdot 10^{-9} & 0.973176 \end{pmatrix}$	2
	MC	$K^* = \begin{pmatrix} 0.973389 & 1.47233 \cdot 10^{-6} \\ 4.66184 \cdot 10^{-6} & 0.973392 \end{pmatrix}$	100
	MC	$K^* = \begin{pmatrix} 0.972753 & -2.59393 \cdot 10^{-5} \\ 4.45803 \cdot 10^{-5} & 0.972688 \end{pmatrix}$	11
0.1	KL	$K^* = \begin{pmatrix} 0.973057 & -1.21852 \cdot 10^{-9} \\ -1.21852 \cdot 10^{-9} & 0.973057 \end{pmatrix}$	54
	MC	$K^* = \begin{pmatrix} 0.974138 & -3.80639 \cdot 10^{-5} \\ 9.22774 \cdot 10^{-6} & 0.973826 \end{pmatrix}$	100
	MC	$K^* = \begin{pmatrix} 0.973099 & -3.08822 \cdot 10^{-5} \\ 2.5945 \cdot 10^{-5} & 0.972769 \end{pmatrix}$	99

Table 27: Upscaled coefficient with $\alpha = 0.1$ and $\tau = 1$.

σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 0.999997 & -3 \cdot 10^{-14} \\ -3 \cdot 10^{-14} & 0.999997 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 1 \cdot 10^{-14} \\ 1 \cdot 10^{-14} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00003 & -4.17333 \cdot 10^{-7} \\ 1.21728 \cdot 10^{-7} & 1.00003 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00002 & -1.15326 \cdot 10^{-6} \\ -1.72008 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 1 & 1.3 \cdot 10^{-12} \\ 1.3 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & -3.43732 \cdot 10^{-8} \\ -3.43732 \cdot 10^{-8} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00031 & -3.77327 \cdot 10^{-6} \\ -3.77327 \cdot 10^{-6} & 1.00031 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00023 & -1.12308 \cdot 10^{-5} \\ -2.84615 \cdot 10^{-7} & 1.00022 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 1 & 3.015 \cdot 10^{-12} \\ 3.015 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & -2.81223 \cdot 10^{-8} \\ -2.81223 \cdot 10^{-8} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00112 & -3.40494 \cdot 10^{-6} \\ 6.51628 \cdot 10^{-5} & 1.00105 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00021 & 3.94357 \cdot 10^{-6} \\ 7.34104 \cdot 10^{-6} & 1.00021 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 0.999948 & 6.05926 \cdot 10^{-10} \\ 6.05926 \cdot 10^{-10} & 0.999948 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 0.999938 & 1.13249 \cdot 10^{-6} \\ 1.13249 \cdot 10^{-6} & 0.999938 \end{pmatrix}$	54
	MC1	$K^* = \begin{pmatrix} 0.976056 & -9.30823 \cdot 10^{-5} \\ 0.00054973 & 0.975952 \end{pmatrix}$	18
	MC2	$K^* = \begin{pmatrix} 0.999219 & 9.45786 \cdot 10^{-6} \\ 1.56579 \cdot 10^{-6} & 0.999235 \end{pmatrix}$	10000

Table 28: Upscaled coefficient with $\alpha = 1$ and $\tau = 1$.

σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 2 & -2.58898 \cdot 10^{-15} \\ -2.58898 \cdot 10^{-15} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00003 & -9.53789 \cdot 10^{-8} \\ 3.81182 \cdot 10^{-7} & 2.00003 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 2.00002 & -8.00069 \cdot 10^{-7} \\ -3.13272 \cdot 10^{-8} & 2.00002 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 2 & -1.7 \cdot 10^{-12} \\ -1.7 \cdot 10^{-12} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & -4.37498 \cdot 10^{-8} \\ -4.37498 \cdot 10^{-8} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00031 & -3.97768 \cdot 10^{-6} \\ 3.58652 \cdot 10^{-6} & 2.00031 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 2.00023 & -1.10975 \cdot 10^{-5} \\ -4.07172 \cdot 10^{-7} & 2.00022 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 2 & 3.11344 \cdot 10^{-9} \\ 3.11344 \cdot 10^{-9} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 6.25077 \cdot 10^{-8} \\ 6.25077 \cdot 10^{-8} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00112 & -2.63017 \cdot 10^{-6} \\ 6.59184 \cdot 10^{-5} & 2.00106 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 2.00022 & 3.66665 \cdot 10^{-6} \\ 7.03134 \cdot 10^{-6} & 2.00022 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 1.99998 & 6.87222 \cdot 10^{-10} \\ 6.87222 \cdot 10^{-10} & 1.99998 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 1.99997 & -3.72147 \cdot 10^{-7} \\ -3.72147 \cdot 10^{-7} & 1.99997 \end{pmatrix}$	54
	MC1	$K^* = \begin{pmatrix} 1.97657 & -8.74909 \cdot 10^{-5} \\ 5.55191 \cdot 10^{-4} & 1.97623 \end{pmatrix}$	18
	MC2	$K^* = \begin{pmatrix} 1.99954 & 9.51709 \cdot 10^{-6} \\ 1.76096 \cdot 10^{-6} & 1.99954 \end{pmatrix}$	10000

Table 29: Upscaled coefficient with coefficient= 2 and $\tau = 1$.

σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 1 & -3 \cdot 10^{-14} \\ -3 \cdot 10^{-14} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & -6.24999 \cdot 10^{-9} \\ -6.24999 \cdot 10^{-9} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00002 & 3.20372 \cdot 10^{-7} \\ 1.0951 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00001 & 1.20297 \cdot 10^{-6} \\ -4.37568 \cdot 10^{-7} & 1.00001 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 0.999999 & 1 \cdot 10^{-13} \\ 1 \cdot 10^{-13} & 0.999999 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 1.47002 \cdot 10^{-12} \\ 1.47002 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00021 & 3.20309 \cdot 10^{-6} \\ 1.60917 \cdot 10^{-6} & 1.00022 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00012 & 1.15273 \cdot 10^{-5} \\ -3.55854 \cdot 10^{-6} & 1.00013 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 0.999998 & 7.80573 \cdot 10^{-10} \\ 7.80573 \cdot 10^{-10} & 0.999998 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 7.81533 \cdot 10^{-9} \\ 7.81533 \cdot 10^{-9} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00055 & -5.19363 \cdot 10^{-6} \\ 6.7596 \cdot 10^{-5} & 1.0005 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00014 & -5.80497 \cdot 10^{-6} \\ 5.25817 \cdot 10^{-6} & 1.00011 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 0.999948 & 6.05926 \cdot 10^{-10} \\ 6.05926 \cdot 10^{-10} & 0.999948 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 0.999872 & 3.0674 \cdot 10^{-8} \\ 3.0674 \cdot 10^{-8} & 0.999872 \end{pmatrix}$	54
	MC1	$K^* = \begin{pmatrix} 0.980719 & 1.1145 \cdot 10^{-4} \\ 8.61976 \cdot 10^{-4} & 0.980919 \end{pmatrix}$	18
	MC2	$K^* = \begin{pmatrix} 0.998296 & -1.59201 \cdot 10^{-5} \\ 8.65048 \cdot 10^{-7} & 0.998297 \end{pmatrix}$	10000

Table 30: Upscaled coefficient with $\alpha = 1$ and $\tau = 0.5$.

σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 2 \cdot 10^{-14} \\ 2 \cdot 10^{-14} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00002 & 4.09402 \cdot 10^{-7} \\ 2.60996 \cdot 10^{-7} & 2.00002 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 2.00001 & 1.39367 \cdot 10^{-6} \\ -1.12553 \cdot 10^{-7} & 2.00001 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 2 & 1.1 \cdot 10^{-12} \\ 1.1 \cdot 10^{-12} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & -6.24975 \cdot 10^{-8} \\ -6.24975 \cdot 10^{-8} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00021 & 3.21561 \cdot 10^{-6} \\ 1.60966 \cdot 10^{-6} & 2.00022 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 2.00012 & 1.16113 \cdot 10^{-5} \\ -3.52223 \cdot 10^{-6} & 2.00013 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 2 & 7.00001 \cdot 10^{-13} \\ 7.00001 \cdot 10^{-13} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 6.25494 \cdot 10^{-9} \\ 6.25494 \cdot 10^{-9} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00056 & -6.36304 \cdot 10^{-6} \\ 6.65335 \cdot 10^{-5} & 2.00051 \end{pmatrix}$	18
	MC2	$K^* = \begin{pmatrix} 2.00015 & -5.37206 \cdot 10^{-6} \\ 5.69184 \cdot 10^{-6} & 2.00015 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 1.99995 & -7.2887 \cdot 10^{-10} \\ -7.2887 \cdot 10^{-10} & 1.99995 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 1.99994 & 9.27524 \cdot 10^{-7} \\ 9.27524 \cdot 10^{-7} & 1.99994 \end{pmatrix}$	54
	MC1	$K^* = \begin{pmatrix} 1.9819 & 9.23236 \cdot 10^{-5} \\ 8.42196 \cdot 10^{-4} & 1.98166 \end{pmatrix}$	18
	MC2	$K^* = \begin{pmatrix} 1.99909 & -1.40936 \cdot 10^{-5} \\ 2.59981 \cdot 10^{-6} & 1.99908 \end{pmatrix}$	10000

Table 31: Upscaled coefficient with $\alpha = 2$ and $\tau = 0.5$.

$\Sigma \vert M$	**Ansatz**	**coefficient**	**N** in each direction
$\Sigma = 0.0001$	KL1	$K^* = \begin{pmatrix} 0.999997 & -3 \cdot 10^{-14} \\ -3 \cdot 10^{-14} & 0.999997 \end{pmatrix}$	2
$\mu_{log} = 1$	KL2	$K^* = \begin{pmatrix} 1 & 1.44426 \cdot 10^{-12} \\ 1.44426 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00003 & -4.17333 \cdot 10^{-7} \\ 1.21728 \cdot 10^{-7} & 1.00003 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00002 & -1.15326 \cdot 10^{-6} \\ -1.72008 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	1
$\Sigma = 0.001$	KL1	$K^* = \begin{pmatrix} 1 & -1.3 \cdot 10^{-12} \\ -1.3 \cdot 10^{-12} & \end{pmatrix}$	2
$\mu_{log} = 1$	KL2	$K^* = \begin{pmatrix} 1 & 2.92263 \cdot 10^{-12} \\ 2.92263 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00031 & -3.77329 \cdot 10^{-6} \\ 3.58006 \cdot 10^{-6} & 1.00031 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00023 & -1.12308 \cdot 10^{-5} \\ -2.8462 \cdot 10^{-7} & 1.00022 \end{pmatrix}$	1
$\Sigma = 0.01$	KL1	$K^* = \begin{pmatrix} 1.00005 & 3.015 \cdot 10^{-12} \\ 3.015 \cdot 10^{-12} & 1.00005 \end{pmatrix}$	2
$\mu_{log} = 1.00005$	KL2	$K^* = \begin{pmatrix} 1.00005 & 1.77067 \cdot 10^{-11} \\ 1.77067 \cdot 10^{-11} & 1.00005 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00113 & -3.48853 \cdot 10^{-6} \\ 6.51272 \cdot 10^{-5} & 1.00107 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00027 & 3.95062 \cdot 10^{-6} \\ 7.40914 \cdot 10^{-6} & 1.00027 \end{pmatrix}$	100
$\Sigma = 0.1$	KL1	$K^* = \begin{pmatrix} 1.00489 & -6.10741 \cdot 10^{-10} \\ -6.10741 \cdot 10^{-10} & 1.00489 \end{pmatrix}$	54
$\mu_{log} = 1.00494$	KL2	$K^* = \begin{pmatrix} 1.0049 & 5.3252 \cdot 10^{-11} \\ 5.3252 \cdot 10^{-11} & 1.0049 \end{pmatrix}$	54
	MC1	$K^* = \begin{pmatrix} 1.00316 & -2.79847 \cdot 10^{-4} \\ 4.48964 \cdot 10^{-4} & 1.00273 \end{pmatrix}$	14
	MC2	$K^* = \begin{pmatrix} 1.00417 & 9.27642 \cdot 10^{-6} \\ 5.49208 \cdot 10^{-7} & 1.00419 \end{pmatrix}$	10000

Table 32: Upscaled lognormal coefficient with expected value $\mu_{log} = \exp(\frac{\sigma^2}{2})$ and $\tau = 1$.

$\Sigma \vert M$	**Ansatz**	**coefficient**	**N** in each direction
$\Sigma = 0.0001$	KL1	$K^* = \begin{pmatrix} 1 & -3 \cdot 10^{-14} \\ -3 \cdot 10^{-14} & 1 \end{pmatrix}$	2
$\mu_{log} = 1$	KL2	$K^* = \begin{pmatrix} 1 & 1.59191 \cdot 10^{-12} \\ 1.59191 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00002 & 3.20373 \cdot 10^{-7} \\ 1.0951 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00001 & 1.20297 \cdot 10^{-6} \\ -4.37568 \cdot 10^{-7} & 1.00001 \end{pmatrix}$	1
$\Sigma = 0.001$	KL1	$K^* = \begin{pmatrix} 0.999999 & 1 \cdot 10^{-13} \\ 1 \cdot 10^{-13} & 0.999999 \end{pmatrix}$	2
$\mu_{log} = 1$	KL2	$K^* = \begin{pmatrix} 1 & 4.3991 \cdot 10^{-12} \\ 4.3991 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00021 & 3.20465 \cdot 10^{-6} \\ 1.62479 \cdot 10^{-6} & 1.00022 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00012 & 1.15304 \cdot 10^{-5} \\ -3.52732 \cdot 10^{-6} & 1.00013 \end{pmatrix}$	1
$\Sigma = 0.01$	KL1	$K^* = \begin{pmatrix} 1.00005 & 7.80613 \cdot 10^{-10} \\ 7.80613 \cdot 10^{-10} & 1.00005 \end{pmatrix}$	2
$\mu_{log} = 1.00005$	KL2	$K^* = \begin{pmatrix} 1.00005 & 3.2471 \cdot 10^{-11} \\ 3.2471 \cdot 10^{-11} & 1.00005 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00058 & -5.33551 \cdot 10^{-6} \\ 6.75597 \cdot 10^{-5} & 1.00053 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00019 & -5.74519 \cdot 10^{-6} \\ 5.32602 \cdot 10^{-6} & 1.0002 \end{pmatrix}$	100
$\Sigma = 0.1$	KL1	$K^* = \begin{pmatrix} 1.00484 & -1.29519 \cdot 10^{-9} \\ -1.29519 \cdot 10^{-9} & 1.00484 \end{pmatrix}$	54
$\mu_{log} = 1.00494$	KL2	$K^* = \begin{pmatrix} 1.00485 & 7.4433 \cdot 10^{-11} \\ 7.4433 \cdot 10^{-11} & 1.00485 \end{pmatrix}$	54

Table 33: Upscaled lognormal coefficient with expected value $\mu_{log} = \exp(\frac{\sigma^2}{2})$ and $\tau = 0.5$.

Σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 0.999997 & 3 \cdot 10^{-14} \\ 3 \cdot 10^{-14} & 0.999997 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 1.44426 \cdot 10^{-12} \\ 1.44426 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00003 & -4.17333 \cdot 10^{-7} \\ 1.21728 \cdot 10^{-7} & 1.00003 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00002 & -1.15326 \cdot 10^{-6} \\ -1.72008 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 1 & -1.3 \cdot 10^{-12} \\ -1.3 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 2.92263 \cdot 10^{-12} \\ 2.92263 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00031 & -3.77329 \cdot 10^{-6} \\ 3.58006 \cdot 10^{-6} & 1.00031 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00023 & -1.12308 \cdot 10^{-5} \\ -2.8462 \cdot 10^{-7} & 1.00022 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 1 & 3.075 \cdot 10^{-12} \\ 3.075 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 1.77066 \cdot 10^{-11} \\ 1.77066 \cdot 10^{-11} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00108 & -3.49242 \cdot 10^{-6} \\ 6.51238 \cdot 10^{-5} & 1.00102 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00022 & 3.94998 \cdot 10^{-6} \\ 7.40804 \cdot 10^{-6} & 1.00022 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 0.999948 & 2.65352 \cdot 10^{-9} \\ 2.65352 \cdot 10^{-9} & 0.999948 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 0.999956 & 4.8025 \cdot 10^{-11} \\ 4.8025 \cdot 10^{-11} & 0.999956 \end{pmatrix}$	54

Table 34: Upscaled coefficient with $\alpha = 1$ and $\tau = 1$ (lognormal).

Σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 1 & -3 \cdot 10^{-14} \\ -3 \cdot 10^{-14} & 1 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 1.59191 \cdot 10^{-12} \\ 1.59191 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00002 & 3.20373 \cdot 10^{-7} \\ 1.0951 \cdot 10^{-7} & 1.00002 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00001 & 1.20297 \cdot 10^{-6} \\ -4.37568 \cdot 10^{-6} & 1.00001 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 0.999999 & 1 \cdot 10^{-13} \\ 1 \cdot 10^{-13} & 0.999999 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 4.3991 \cdot 10^{-12} \\ 4.3991 \cdot 10^{-12} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00021 & 3.20465 \cdot 10^{-6} \\ 1.62479 \cdot 10^{-6} & 1.00022 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 1.00012 & 1.15304 \cdot 10^{-5} \\ -3.52732 \cdot 10^{-6} & 1.00013 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 0.999998 & 7.80573 \cdot 10^{-10} \\ 7.80573 \cdot 10^{-10} & 0.999998 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 1 & 3.2471 \cdot 10^{-11} \\ 3.2471 \cdot 10^{-11} & 1 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 1.00053 & -5.33934 \cdot 10^{-6} \\ 6.75526 \cdot 10^{-5} & 1.00048 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 1.00014 & -5.74432 \cdot 10^{-6} \\ 5.32569 \cdot 10^{-6} & 1.00015 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 0.999898 & -1.27463 \cdot 10^{-9} \\ -1.27463 \cdot 10^{-9} & 0.999898 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 0.999914 & 8.3377 \cdot 10^{-11} \\ 8.3377 \cdot 10^{-11} & 0.999914 \end{pmatrix}$	54

Table 35: Upscaled coefficient with $\alpha = 1$ and $\tau = 0.5$ (lognormal).

Σ	**Ansatz**	**coefficient**	**N** in each direction
0.0001	KL1	$K^* = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 2.87191 \cdot 10^{-12} \\ 2.87191 \cdot 10^{-12} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00002 & 4.09402 \cdot 10^{-7} \\ 2.60996 \cdot 10^{-7} & 2.00002 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 2.00001 & 1.39367 \cdot 10^{-6} \\ -1.12553 \cdot 10^{-7} & 2.00001 \end{pmatrix}$	1
0.001	KL1	$K^* = \begin{pmatrix} 2 & 1.1 \cdot 10^{-12} \\ 1.1 \cdot 10^{-12} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 5.6791 \cdot 10^{-12} \\ 5.6791 \cdot 10^{-12} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00021 & 3.21561 \cdot 10^{-6} \\ 1.60966 \cdot 10^{-6} & 2.00022 \end{pmatrix}$	2
	MC2	$K^* = \begin{pmatrix} 2.00012 & 1.16113 \cdot 10^{-5} \\ -3.52224 \cdot 10^{-6} & 2.00013 \end{pmatrix}$	1
0.01	KL1	$K^* = \begin{pmatrix} 2 & 7.00005 \cdot 10^{-13} \\ 7.00005 \cdot 10^{-13} & 2 \end{pmatrix}$	2
	KL2	$K^* = \begin{pmatrix} 2 & 3.3751 \cdot 10^{-11} \\ 3.3751 \cdot 10^{-11} & 2 \end{pmatrix}$	2
	MC1	$K^* = \begin{pmatrix} 2.00054 & -6.53769 \cdot 10^{-6} \\ 6.64053 \cdot 10^{-5} & 2.00049 \end{pmatrix}$	4
	MC2	$K^* = \begin{pmatrix} 2.00015 & -5.32668 \cdot 10^{-6} \\ 5.77404 \cdot 10^{-6} & 2.00016 \end{pmatrix}$	100
0.1	KL1	$K^* = \begin{pmatrix} 1.99995 & -1.58594 \cdot 10^{-9} \\ -1.58594 \cdot 10^{-9} & 1.99995 \end{pmatrix}$	54
	KL2	$K^* = \begin{pmatrix} 1.99996 & 1.09234 \cdot 10^{-10} \\ 1.09234 \cdot 10^{-10} & 1.99996 \end{pmatrix}$	54

Table 36: Upscaled coefficient with $\alpha = 2$ and $\tau = 0.5$ (lognormal).

m ϵ_{ACA}	5	10	20	40	80
10^{-2}	0.18	0.56	0.57	1.18	3.01
10^{-4}	0.36	0.36	1.07	2.23	2.64
10^{-6}	0.23	0.35	0.69	1.45	2.74
10^{-8}	0.36	0.65	1.15	2.22	3.91
10^{-10}	0.35	0.59	1.03	3.10	3.65
0	1.65	1.98	3.01	5.91	12.58

Table 37: Required time in seconds to calculate m eigenpairs for the weak admissibility condition and leaf size 1.

m ϵ_{ACA}	5	10	20	40	80
10^{-2}	0.05	0.20	0.28	0.48	1.66
10^{-4}	0.13	0.19	0.65	1.26	1.66
10^{-6}	0.13	0.20	0.57	0.67	1.70
10^{-8}	0.16	0.24	0.70	0.80	1.96
10^{-10}	0.31	0.30	0.61	0.99	2.27
0	1.65	1.98	3.01	5.91	12.58

Table 38: Required time in seconds to calculate m eigenpairs for the weak admissibility condition and leaf size 4.

m ϵ_{ACA}	5	10	20	40	80
10^{-2}	1.86	10.93	7.57	12.4	17.84
10^{-4}	2.01	2.65	6.18	31.42	24.78
10^{-6}	2.11	4.58	8.71	17.04	27.07
10^{-8}	2.1	2.99	5.77	11.48	18.23
10^{-10}	2.01	2.82	5.48	10.76	17.18
0	1.65	1.98	3.01	5.91	12.58

Table 39: Required time in seconds to calculate m eigenpairs for the standard admissibility condition and leaf size 1.

m ϵ_{ACA}	5	10	20	40	80
10^{-2}	0.73	3.27	3.20	6.40	9.21
10^{-4}	0.89	1.27	3.59	14.49	21.24
10^{-6}	0.75	1.08	2.08	3.32	8.33
10^{-8}	0.97	1.16	2.24	3.55	7.40
10^{-10}	1.00	1.30	2.49	3.99	8.16
0	1.65	1.98	3.01	5.91	12.58

Table 40: Required time in seconds to calculate m eigenpairs for the standard admissibility condition and leaf size 4.

References

[1] Arpack. Website. `http://www.caam.rice.edu/software/ARPACK/`.

[2] Matlab. Website. `http://www.mathworks.com/products/matlab/`.

[3] J.E. Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *SIAM MMS*, 2:421–439, 2004.

[4] J.E. Aarnes and Y. Efendiev. Mixed multiscale finite element methods for stochastic porous media flows. *SIAM J. Sci. Comput.*, 30(5):2319–2339, 2008.

[5] J.E. Aarnes, Y. Efendiev, and L. Jiang. Analysis of multiscale finite element methods using global information for two-phase flow simulations. *SIAM MMS*, 7(2):655–676, 2007.

[6] J.E. Aarnes, T. Gimse, and K.-A. Lie. An introduction to the numerics of flow in porous media using matlab. In *Geometric Modelling, Numerical Simulation, and Optimization*, pages 265–306. 2007.

[7] J.E. Aarnes, S. Krogstad, and K.-A. Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, 5(2):337–363, 2006.

[8] A. Anantharaman, R. Costaouec, C. Le Bris, F. Legoll, and F. Thomines. Introduction to numerical stochastic homogenization and the related computational challenges: some recent developments. In W. Bao and Q. Du, editors, *Multiscale Modeling and Analysis for Material Simulation*, volume 22, pages 197–272, National University of Singapore, 2011. Institute for Mathematical Sciences, Lecture Notes Series.

[9] G. Bal. Homogenization in random media and effective medium theory for high frequency waves. *Discrete and Continuous Dynamical Systems, Ser. B*, 8(2):473–492, 2007.

[10] A. Barth, C. Schwab, and N. Zollinger. Multi-level Monte Carlo Finite Element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, 199(1):123–161, 2011.

[11] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. II: Implementation and tests in DUNE. *Computing*, 82(2-3):121–138, 2008.

[12] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. I: Abstract framework. *Computing*, 82(2-3):103–119, 2008.

[13] X. Blanc, R. Costaouec, C. Le Bris, and F. Legoll. Variance reduction in stochastic homogenization using antithetic variables. *Markov Processes and Related Fields*, 18(1):31–66, 2012. (preliminary version available at `http://www.alglib.net/eigen/symmetric/symmevd.php`).

[14] M. Blatt and P. Bastian. The iterative solver template library. In B. Kagström, E. Elmroth, J. Dongarra, and J. Wasniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, number 4699 in Lecture Notes in Scientific Computing, pages 666–675. Springer, 2007.

[15] S. Börm, L. Grasedyck, and W. Hackbusch. Hlib. Website. `http://www.hlib.org`.

[16] A. Bourgeat and A. Piatnitski. Approximations of effective coefficients in stochastic homogenization. *Annales de l'Institut Henri PoincarÃ© (B) Probability and Statistics*, 40(2):152–165, 2004.

[17] R.E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. Acta Numerica 7, 1-49 (1998)., 1998.

[18] J. Charrier, R. Scheichl, and A. Teckentrup. Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. *SIAM Journal on Numerical Analysis*. in press.

[19] Y. Chen and L. Durlofsky. Ensemble level upscaling for efficient estimation of fine-scale production statistics. *SPE Journal*, (13):400–411, 2008.

[20] Y. Chen, L. Durlofsky, M. Gerritsen, and X. Wen. A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, 26:1041–1060, 2003.

[21] Z. Chen and T.Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comput.*, 72(242):541–576, 2003.

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

[23] R. Costaouec, C. Le Bris, and F. Legoll. Variance reduction in stochastic homogenization: Proof of concept, using antithetic variables. *Bol. Soc. Esp. Mat. Apl.*, 50:9–27, 2010.

[24] A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger. A generic interface for parallel and adaptive scientific computing: Abstraction principles and the DUNE-FEM module. *Computing*, 90(3–4):165–196, 2010.

[25] DUNE-Group. The distributed and unified numerics enviroment (dune). Website. `http://www.dune-project.org/`.

[26] U. (ed.) Hornung. *Homogenization and porous media.* Interdisciplinary Applied Mathematics. 6. New York, NY: Springer. xvi, 275 p., 1997.

[27] Y. Efendiev. *The multiscale finite element method (MsFEM) and its applications.* PhD thesis, California Institute of Technology, 1999.

[28] Y. Efendiev, J. Galvis, and F. Thomines. A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications. *Multiscale Model. Sim.*, (submitted), 2011.

[29] Y. Efendiev and T.Y. Hou. *Multiscale finite element methods. Theory and applications.* Springer, 2009.

[30] Y. Efendiev, O. Iliev, and C. Kronsbein. Multi-level Monte Carlo methods using ensemble level mixed MsFEM for two-phase flow and transport simulations. Submitted to Comp. Geosci., 2013.

[31] Y. Efendiev, F. Legoll, and C. Kronsbein. Multi-level Monte Carlo approaches for numerical homogenization. Submitted to SIAM MMS (preliminary version available at `http://arxiv.org/abs/1301.2798`), 2013.

[32] G.S. Fishman. *Monte Carlo: Concepts, algorithms, and applications.* Springer Series in Operations Research. Springer-Verlag, New York, 1996.

[33] P. Frauenfelder, C. Schwab, and R.A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Eng.*, 194(2-5):205–228, 2005.

[34] M. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. Keller, Alexander (ed.) et al., Monte Carlo and quasi-Monte Carlo methods 2006. Berlin: Springer. 343-358 (2008)., 2008.

[35] M. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[36] A. Gloria and F. Otto. An optimal variance estimate in stochastic homogenization of discrete elliptic equations. *Ann. Probab.*, 39(3):779–856, 2011.

[37] W. Hackbusch, L. Grasedyck, and S. Börm. An introduction to hierarchical matrices. *Math. Bohem.*, 127(2):229–241, 2002.

[38] S. Heinrich. Multilevel Monte Carlo methods. In S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-scale scientific computing*, volume 2179 of *Lecture Notes in Computer Science*, pages 58–67. Berlin: Springer, 2001.

[39] R. Helmig. *Multiphase Flow and Transport Processes in the Subsurface: A Contribution to the Modeling of Hydrosystems.* Springer-Verlag, Berlin, Heidelberg, 1997.

[40] T.Y. Hou and X.H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134(1):169–189, 1997.

[41] O. Iliev and I. Rybak. On numerical upscaling for flows in heterogeneous porous media. *Comput. Methods Appl. Math.*, 8(1):60–67, 2008.

[42] V.V. Jikov, S M. Kozlov, and O.A. Oleinik. *Homogenization of differential operators and integral functionals.* Springer-Verlag, 1994.

[43] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: Statistical and numerical approach. *International Journal of Solids and Structures*, 40(13-14):3647–3679, 2003.

[44] B.N. Khoromskij, A. Litvinenko, and H.G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.

[45] K.-A. Lie, S. Krogstad, I. Ligaarden, J. Natvig, H. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16(2):297–322, 2012.

[46] M. Loeve. *Probability theory II. 4th ed.* Graduate Texts in Mathematics. 46. New York - Heidelberg - Berlin: Springer-Verlag. XVI, 413 p. , 1978.

[47] X. Ma and N. Zabaras. A stochastic mixed finite element heterogeneous multiscale method for flow in porous media. *J. Comput. Phys.*, 230(12):4696–4722, 2011.

[48] J. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontiuous Galerkin schemes with optimal ordering of elements.

[49] H. Owhadi and L. Zhang. Metric-based upscaling. *Communications on Pure and Applied Mathematics*, 60(5):675–723, 2007.

[50] G. Papanicolaou and S. Varadhan. Boundary value problems with rapidly oscillating random coefficients. Random fields. Rigorous results in statistical mechanics and quantum field theory, Esztergom 1979, Colloq. Math. Soc. Janos Bolyai 27, 835-873 (1981)., 1981.

[51] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parameterized Partial Differential Equations*, volume 1.0. to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2006.

[52] D.V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA J. Numer. Anal.*, 26(3):423–445, 2006.

[53] X.H. Wu, Y. Efendiev, and T.Y. Hou. Analysis of upscaling absolute permeability. *Discrete Contin. Dyn. Syst., Ser. B*, 2(2):185–204, 2002.

[54] D. Xiu. Fast numerical methods for stochastic computations: A review. *Commun. Comput. Phys*, 5(2-4):242–272, 2009.

[55] D. Xiu and G.E. Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Comput. Methods Appl. Mech. Eng.*, 191(43):4927–4948, 2002.

[56] D. Xiu and G.E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.

[57] V.V. Yurinskii. Averaging of symmetric diffusion in random medium. *Sibirskii Mat. Zh.*, 27(4):167–180, 1986.

CURRICULUM VITAE

Cornelia Kronsbein

deutsch:

1988-1992	Besuch der Grundschule in Werther (Westf.)
1992-2001	Besuch des evangelischen Gymnasium Werther (EGW) Abschluss: Abitur
2001-2008	Studium der Mathematik an der Albert-Ludwigs-Universität Freiburg
September 2008	Diplom in Mathematik
2009	ProSat Studium an der TU Kaiserslautern
2009-2012	Promotion im Fachbereich Mathematik an der TU Kaiserslautern

English:

1988-1992	Elementary school in Werther (Westf.)
1992-2001	Secondary school: Evangelisches Gymnasium Werther
2001-2008	University of Freiburg, Department of Mathematics
September 2008	Diploma in Mathematics
2009	TU Kaiserslautern, ProSat student
2009-2012	TU Kaiserslautern, PhD student in the Department of Mathematics