International Journal for Uncertainty Quantification, 7(1): 57–81 (2017)

A MULTIGRID MULTILEVEL MONTE CARLO METHOD USING HIGH-ORDER FINITE-VOLUME SCHEME FOR LOGNORMAL DIFFUSION PROBLEMS

Prashant Kumar,[1,*] Cornelis W. Oosterlee,[1,2] & Richard P. Dwight[3]

[1]*CWI—Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*
[2]*Delft University of Technology, Delft Institute of Applied Mathematics, Delft, The Netherlands*
[3]*Delft University of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands*

*Address all correspondence to: Prashant Kumar, CWI—Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, E-mail: pkumar@cwi.nl

Original Manuscript Submitted: 9/7/2016; Final Draft Received: 11/9/2016

The aim of this paper is to show that a high-order discretization can be used to improve the convergence of a multilevel Monte Carlo method for elliptic partial differential equations with lognormal random coefficients in combination with the multigrid solution method. To demonstrate this, we consider a fourth-order accurate finite-volume discretization. With the help of the Matérn family of covariance functions, we simulate the coefficient field with different degrees of smoothness. The idea behind using a fourth-order scheme is to capture the additional regularity in the solution introduced due to higher smoothness of the random field. Second-order schemes previously utilized for these types of problems are not able to fully exploit this additional regularity. We also propose a practical way of combining a full multigrid solver with the multilevel Monte Carlo estimator constructed on the same mesh hierarchy. Through this integration, one full multigrid solve at any level provides a valid sample for all the preceding Monte Carlo levels. The numerical results show that the fourth-order multilevel estimator consistently outperforms the second-order variant. In addition, we observe an asymptotic gain for the standard Monte Carlo estimator.

KEY WORDS: *stochastic partial differential equations, groundwater flow, random fields, fourth-order discretization, full multigrid, multilevel Monte Carlo*

1. INTRODUCTION

In this paper we develop and analyze a multilevel Monte Carlo method based on a high-order finite-volume (FV) approximation of a class of elliptic partial differential equations (PDEs) with lognormal random coefficients. Such PDEs find their application for example in hydrogeology and are stated as

$$-\nabla \cdot (a(\vec{x},\omega)\nabla u(\vec{x},\omega)) = f(\vec{x}), \qquad \text{for} \quad \vec{x} \in \mathcal{D} \subset \mathbb{R}^d, \qquad d = 1,2,3. \tag{1.1}$$

We denote by ω an event in the complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space with σ-algebra $\mathcal{F}$ and probability measure $\mathbb{P}$. Quantity u represents the fluid pressure, $a : \Omega \times \bar{\mathcal{D}} \to \mathbb{R}$ is the random permeability field and f is the known source term. The PDE (1.1) describes a steady-state single-phase flow in a porous medium and is the result of the coupling between the Darcy flux $\vec{q} := -a\nabla u$ and the incompressiblity condition $\nabla \cdot \vec{q} = f$. We consider deterministic mixed Dirichlet-Neumann boundary conditions

$$u(\vec{x},\cdot) = g_D(\vec{x}) \qquad \text{for} \quad \vec{x} \in \Gamma^D, \qquad \text{and} \qquad a(\vec{x},\omega)\frac{\partial u}{\partial \mathbf{n}}(\vec{x},\cdot) = g_N(\vec{x}) \qquad \text{for} \quad \vec{x} \in \Gamma^N, \tag{1.2}$$

where Γ^D and Γ^N represent the boundaries for Dirichlet and Neumann boundary conditions, respectively, and $\mathbf{n}$ denotes the outward normal to Γ^N. In the context of fluid flow problems, Dirichlet boundary conditions represent the

pressure values at inflow and outflow heads, whereas the Neumann boundary conditions define the pressure gradient at the boundaries perpendicular to the main flow direction.

In stochastic subsurface flow modeling, it is well recognized that a lognormal random field may accurately represent the permeability of a naturally occurring heterogeneous porous medium [1–3]. The logarithm of the permeability field, $\log a = Z$, where $Z : \Omega \times \bar{\mathcal{D}} \to \mathbb{R}$ is a zero-mean Gaussian random field. Therefore

$$\mathbb{E}[Z(\vec{x},\cdot)] = 0 \quad \text{and} \quad \mathbf{cov}(Z(\vec{x},\cdot), Z(\vec{y},\cdot)) = \mathbb{E}[Z(\vec{x},\cdot)Z(\vec{y},\cdot)], \qquad \vec{x},\vec{y} \in \mathcal{D}. \tag{1.3}$$

The lognormal property also ensures a positive permeability throughout the domain. For further simplification, we consider a $homogeneous$ covariance function $R : \mathbb{R} \to \mathbb{R}$ such that

$$\mathbf{cov}(Z(\vec{x},\cdot), Z(\vec{y},\cdot)) = R(r) \qquad \text{with} \qquad r = ||\vec{x} - \vec{y}||_2. \tag{1.4}$$

The homogeneity condition guarantees an *isotropic* and *stationary* Gaussian process. For any stationary process the covariance function is invariant under any spatial translation. In this paper we use the Matérn covariance function [4] characterized by the parameter set $\Phi = \{\nu, \lambda_c, \sigma^2\}$, i.e.,

$$R_\Phi(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(2\sqrt{\nu}\frac{r}{\lambda_c}\right)^\nu K_\nu \left(2\sqrt{\nu}\frac{r}{\lambda_c}\right), \tag{1.5}$$

where Γ is the gamma function and K_ν is the modified Bessel function of the second kind. The parameter ν defines the smoothness, σ^2 is the variance, and λ_c is the correlation length of the covariance function. The Matérn model has great flexibility in modeling of spatial processes because of its smoothness parameter ν, which governs the differentiabilty of the random field. For $\nu = 1/2$ the Matérn function corresponds to an exponential model and with $\nu \to \infty$ to a Gaussian model. The parameters λ_c and σ_c^2 dictate the number of peaks and the amplitude of the random field, respectively. In general, realizations of the random field are "almost surely" Hölder continuous, $Z, a \in C^\eta(\bar{\mathcal{D}})$ with $0 < \eta < \nu$ (see, e.g., [5,6]).

We refer to Gilbarg and Trudinger [7], for standard results on the regularity for second-order elliptic PDEs. In particular, we shall operate under the assumptions that $f \in H^k(\mathcal{D})$ and $a \in C^{k+1}(\bar{\mathcal{D}})$ for some $k \in \mathbb{N}$ which ensures that $u \in H^{k+2}(\mathcal{D})$. In the recent years, (1.1) has been extensively studied and the regularity results have been further refined; taking into account the regularity of the lognormal coefficient field [6,8,9].

We are particularly interested in computing expected values of different linear functionals of the solution u denoted by $Q := Q(u)$. Due to high-dimensional nature of the random permeability field, it becomes very challenging to obtain reliable estimates of these hydrogeological quantities. The choice for appropriate uncertainty quantification tools boils down to the computational efficiency. In the present context, Monte Carlo (MC)-type methods are sometimes favored [10] due to their dimension independent convergence property and simplicity of implementation. For standard MC method, the root-mean-square error (RMSE) converges as $(\mathbb{V}[Q]/N)^{1/2}$, where N is the number of samples and $\mathbb{V}[Q]$ is the sample variance. This slow convergence with respect to N is, however, the main drawback of the method. To remedy this, various sampling and variance reduction techniques have been applied, for example, the authors in [11] have applied a quasi-Monte Carlo method to improve the convergence rate for this problem. More recently, multilevel Monte Carlo (MLMC) methods have been formulated for this problem (see, e.g., [8,12,13]), which lead to a dramatic reduction in computational cost compared to the classical MC approach. The idea of the MLMC method was introduced by Heinrich [14] to speed up the computation of high-dimensional integrals. This multilevel idea was further developed by Giles [15] to reduce the order of complexity of Monte Carlo path simulations for stochastic differential equations (SDEs). The improved efficiency of the MLMC method comes from building the estimate for Q, on a hierarchy of grids or levels, by exploiting the linearity of the expectation operator, i.e.,

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}], \tag{1.6}$$

for all $\ell \in \{0, 1, ..., L\}$. On the coarsest grid for $\ell = 0$, expectations are inexpensive to compute accurately and for large values of ℓ, where the numerical solution is comparatively expensive, only a few realizations are required as the

variance of the correction term $\mathbb{V}[Q_\ell - Q_{\ell-1}]$ is significantly smaller compared to the variance of Q, i.e., $\mathbb{V}[Q_\ell]$. While offering large savings over the standard MC method, MLMC retains all the important properties of MC methods like parallelization and combination with other complementary variance reduction techniques (see, e.g., [6,16–18]).

There are two objectives of this paper. First is to show that a high-order discretization scheme can be used to reduce the computational costs of the MLMC estimator for problems that exhibit high spatial smoothness. In previous work, Giles [19] has shown an improvement from $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon)^2)$ to $\mathcal{O}(\varepsilon^{-2})$ to achieve an RMSE of $\mathcal{O}(\varepsilon)$ using a higher-order Milstein discretization compared to Euler path discretization for SDEs and certain financial payoffs with Lipschitz bound. This gain was achieved due to improvement in the strong convergence order of the schemes that is central to the efficiency of the multilevel method. In this paper we choose to use a fourth-order accurate finite volume (FV) method on a uniform mesh. The key ingredient for achieving this high-order accuracy is using a fourth-order accurate quadrature rule to approximate the boundary fluxes for each control volume. So far, the MLMC literature on Darcy flow problems relies on a second-order accurate finite-element (FE) or FV discretization scheme. We show that for certain linear functionals of u the overall asymptotic cost of the MLMC method can be improved under some smoothness assumptions. Even for the case when no asymptotic gain is possible, we show a reduction in terms of the number of MLMC levels and samples.

The second objective is to define a structure in which the multilevel estimator is integrated into a multilevel solver. To demonstrate this we use a full multigrid (FMG) solver [20] based on the same grid hierarchy as the MLMC estimator for numerical PDE solution. We will describe in detail the modifications required to transform this solver for the MLMC method so that the telescopic identity (1.6) is not violated. This method is particularly effective for less-smooth quantities of interest. Further, we also provide an efficient and scalable multigrid solver for the fourth-order linear system in 2D obtained by combining a cell-centered multigrid for the second-order discretization with the defect correction strategy.

The outline of this paper is as follows. We begin by reviewing the standard MC and MLMC schemes in Section 2. In Section 3 we describe in detail the second- and fourth-order discretization schemes in two dimensions. Section 4 explains the mechanics of coupling of a FMG solver and the multilevel estimator. Section 5 is devoted to the technical details required to construct the FMG solver for both discretization schemes. In Section 6 we provide some numerical results in two spatial dimensions for different quantities of interest and compare the cost of different MC estimators.

2. MULTILEVEL MONTE CARLO METHOD

The PDE discretization introduces a bias in MC estimators. Therefore, the computational grid should be fine enough to keep this bias below the required tolerance. A high-order scheme can achieve a higher accuracy (lower bias) on a relatively coarse grid and can be utilized to reduce the cost of MC estimators. In this section, we first describe the single-level MC estimator where all the MC samples are based on same discretization mesh. Then we extend this framework to the multilevel Monte Carlo estimator.

2.1 Single-Level Monte Carlo Estimator

To explain these methods we follow the descriptions outlined in [15,21,22]. In a single-level MC scheme we approximate $\mathbb{E}[Q]$ using the standard MC estimator which employs N realizations of the solution on a grid with mesh width h, i.e.,

$$\mathbb{E}[Q] \approx \mathbb{E}[Q_h] \approx \mathcal{Q}_{h,N}^{MC} = \frac{1}{N}\sum_{i=1}^{N} Q_h(\omega_i), \tag{2.1}$$

where $\omega_i \in \Omega$. The above estimator is biased due to spatial discretization. The mean-square-error (MSE) is quantified as

$$\mathcal{E}(\mathcal{Q}_{h,N}^{MC})^2 = \mathbb{E}[(\mathcal{Q}_{h,N}^{MC} - \mathbb{E}[Q])^2] = \frac{\mathbb{V}[Q_h]}{N} + (\mathbb{E}[Q_h - Q])^2, \tag{2.2}$$

where $\mathbb{V}[Q_h]$ is the sample variance. The first error component in (2.2) is the variance of the estimator, accounting for the sampling error, which decays with the number of MC samples N. The second term corresponds to the square of the mean FV discretization error, which converges with respect to the mesh width as $\mathcal{O}(h^\alpha)$:

$$|\mathbb{E}[Q_h - Q]| = c_1 h^{\alpha}, \qquad \alpha > 0, \tag{2.3}$$

where c_1 is a FV error constant. The sufficient condition for the MSE to be bounded by ε^2 ($\varepsilon > 0$) is that both sources of error are bounded by $\varepsilon^2/2$. This is achieved by choosing a mesh size $h = \mathcal{O}(\varepsilon^{1/\alpha})$ and the number of samples $N = \mathcal{O}(\varepsilon^{-2})$, assuming $\mathbb{V}[Q_h]$ is constant, independent of the grid size. With this, if we express the cost to compute one sample of Q_h by $\mathcal{C}_h = \mathcal{O}(h^{-\gamma})$, where $\gamma \geq 1$, we obtain a total cost of

$$\mathcal{C}_\varepsilon^{MC} = N\mathcal{C}_h = \mathcal{O}\left(\varepsilon^{-2-\gamma/\alpha}\right), \tag{2.4}$$

to achieve a RMSE of ε. The complexity of the standard MC method can be improved using an optimal solver, e.g., by an FMG solver with $\gamma \approx d$ (where d is the number of spatial dimensions). Another way to reduce the costs is by using a higher-order discretization, i.e., with an increased value for α. A larger value of α can give the same accuracy on a coarser grid and can therefore make the MC simulation significantly cheaper. For example, in a deterministic setting, discretization accuracy of $\mathcal{O}(h^2)$ on a 256×256 grid is essentially the same as that of the $\mathcal{O}(h^4)$ discretization on a 16×16 grid (assuming a small constant of proportionality for the fourth-order convergence). However, solving the 16×16 problem with fourth-order accuracy is typically cheaper than solving the 256×256 problem with second-order discretization. The fourth-order MC scheme may still be cheaper if the underlying solver is slightly sub-optimal compared to the second-order solver. Later we will show that our FMG solvers for both second- and fourth-order discretization are asymptotically optimal.

2.2 Multilevel Monte Carlo Estimators

In the following, we briefly discuss the key elements of the MLMC scheme as explained in Cliffe et al. [21]. We begin by defining a hierarchy of *cell-centered grids* $\{\mathcal{D}_{h_\ell}\}_{\ell=0}^{L}$ as a family of nodes given by

$$\mathcal{D}_{h_\ell} = \left\{(x_i, y_j) : x_i = \left(i - \frac{1}{2}\right) h_\ell, y_j = \left(j - \frac{1}{2}\right) h_\ell; i, j = 1, 2, ..., m_\ell; m_\ell = \frac{1}{h_\ell}\right\}, \tag{2.5}$$

where h_ℓ is the mesh width. A control volume with center (x_i, y_j) is denoted by $\mathcal{D}_{h_\ell}^{(i,j)}$. We assume that the mesh sizes are strictly decreasing such that $h_\ell = s^{-1} h_{\ell-1}$ with the grid scaling factor $s \in \mathbb{N}\backslash\{1\}$ for all $\ell \in \{0, 1, ..., L\}$ and $h_{-1} = 0$. In the following, we only consider uniform coarsening with $s = 2$, such that the number of cells doubles in each of the coordinate directions with increasing level.

The individual expectations in the telescopic sum (1.6) are computed independently using a standard MC estimator (2.1). The multilevel estimator reads

$$\mathbb{E}[Q_{h_L}] \approx \mathcal{Q}_{h_L}^{ML} = \frac{1}{N_0} \sum_{i=1}^{N_0} Q_{h_0}(\omega_i) + \sum_{\ell=1}^{L} \left(\frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_{h_\ell}(\omega_i) - Q_{h_{\ell-1}}(\omega_i)) \right), \tag{2.6}$$

where the number of MLMC samples $N_\ell \in \mathbb{N}$ form a decreasing sequence for increasing ℓ. As each of the expectations in the above estimator is computed independently, the variance of the multilevel estimator is the sum of variances of individual estimators, i.e.,

$$\mathbb{V}[\mathcal{Q}_{h_L}^{ML}] = \sum_{\ell=0}^{L} N_\ell^{-1} \mathcal{V}_\ell, \tag{2.7}$$

where $\mathcal{V}_\ell = \mathbb{V}[(Q_{h_\ell} - Q_{h_{\ell-1}})]$ and $Q_{h_{-1}} = 0$. Using a similar argument as in (2.2), the MSE of the multilevel estimator is given by

$$\mathcal{E}(\mathcal{Q}_{h_L}^{ML})^2 = \mathbb{E}[(\mathcal{Q}_{h_L}^{ML} - \mathbb{E}[Q])^2] = \sum_{\ell=0}^{L} \frac{\mathcal{V}_\ell}{N_\ell} + (\mathbb{E}[Q_{h_L} - Q])^2. \tag{2.8}$$

Note that the FV bias in this multilevel estimator is the same as for the MC estimator. In order to ensure that the MSE (2.8) is smaller than the prescribed tolerance ε^2, we require both the error components to be less than $\varepsilon^2/2$. Therefore, the number of levels is derived using (2.3), as

$$(\mathbb{E}[Q_{h_L} - Q])^2 = (c_1 h_L^{\alpha})^2 < \frac{\varepsilon^2}{2}. \tag{2.9}$$

Taking $h_L = s^{-L} h_0$, we get the number of MLMC levels as

$$L = \lceil \alpha^{-1} \log_s(\varepsilon^{-1}) + c_2 \rceil, \tag{2.10}$$

where c_2 is some constant depending on (c_1, s, h_0, α) but independent of ε. The above relation requires *a priori* knowledge of the constants c_1 and α, which may not be available for the numerical scheme used. Practical implementations of the MLMC method however do not require a value of L in advance. Usually, we start with a small L value and increase the number of levels until the following criterion is met,

$$|\mathbb{E}[Q_{h_L} - Q_{h_{L-1}}]| < (s^{\alpha} + 1)\frac{\varepsilon}{\sqrt{2}}, \tag{2.11}$$

which is obtained using the triangular inequality and (2.3). This numerically confirms that the bias of the MLMC estimator is less than $\varepsilon/\sqrt{2}$. The value of α is empirically determined on-the-fly using the sample averages from available levels.

Assuming that the cost of one sample on level ℓ is $\mathcal{C}_\ell = \mathcal{O}(h_\ell^{-\gamma})$, then the total cost of the MLMC estimator adds up to $\mathcal{C}^{ML} = \sum_{\ell=0}^{L} N_\ell \mathcal{C}_\ell$. The number of samples at different levels can be derived by minimizing the total cost for a fixed variance of $\varepsilon^2/2$. As suggested in [15], by treating N_ℓ as a continuous variable and using a Lagrange multiplier ζ, we can get the constraint equation

$$\frac{\partial}{\partial N_\ell}\left(\sum_{k=0}^{L} N_k \mathcal{C}_k + \zeta^2 N_k^{-1} \mathcal{V}_k\right) = 0, \tag{2.12}$$

which yields

$$N_\ell = \zeta\sqrt{\mathcal{V}_\ell/\mathcal{C}_\ell}, \tag{2.13}$$

By substituting $\mathbb{V}[\mathcal{Q}_{h_L}^{ML}] = \varepsilon^2/2$ in (2.7), we obtain

$$\zeta = 2\varepsilon^{-2}\sum_{\ell=0}^{L}\sqrt{\mathcal{V}_\ell \mathcal{C}_\ell}, \tag{2.14}$$

and hence the total cost to obtain a RMSE of ε is given by

$$\mathcal{C}_\varepsilon^{ML} = \sum_{\ell=0}^{L} N_\ell \mathcal{C}_\ell = 2\varepsilon^{-2}\left(\sum_{\ell=0}^{L}\sqrt{\mathcal{V}_\ell \mathcal{C}_\ell}\right)^2. \tag{2.15}$$

For completeness, we now state the generalized MLMC complexity theorem (for proof see [21]).

Theorem 1. *Let there be positive constants* $\alpha, \beta, \gamma > 0$ *such that* $\alpha \geq 1/2 min(\beta, \gamma)$ *and*

1. $|\mathbb{E}[Q_h - Q]| = \mathcal{O}(h^{\alpha})$;
2. $\mathbb{V}[Q_{h_\ell} - Q_{h_{\ell-1}}] = \mathcal{O}(h_\ell^{\beta})$;
3. $C_\ell = \mathcal{O}(h_\ell^{-\gamma})$.

Then, for any $\varepsilon < exp(-1)$, there exists a value L and a decreasing sequence $\{N_\ell\}_{\ell=0}^{L}$, such that $\mathcal{E}(\mathcal{Q}_{h_L}^{ML}) < \varepsilon$ and

$$\mathcal{C}_\varepsilon^{ML} = \begin{cases} \mathcal{O}\left(\varepsilon^{-2}\right), & when \quad \beta > \gamma, \\ \mathcal{O}\left(\varepsilon^{-2}(\log \varepsilon)^2\right), & when \quad \beta = \gamma, \\ \mathcal{O}\left(\varepsilon^{-2-((\gamma-\beta)/\alpha)}\right), & when \quad \beta < \gamma. \end{cases}$$

The parameters α and β depend on the numerical method used to approximate the PDE and can be increased by high-order discretizations. Also, these parameters do not change with the spatial dimension. The rate γ increases with dimension and when using an optimal solver it is approximately equal to the spatial dimension d. This implies that for some quantities of interest the asymptotic cost might change when dealing with higher-dimensional problems. Therefore, a high-order MLMC scheme may help in countering this effect. Even for the case where we do not have the scope of asymptotic improvement, we can reduce the number of MLMC levels by increasing α as observed in (2.10) and the number of samples by improving the parameter β from (2.13).

3. FV DISCRETIZATION

When modeling the subsurface flow process, FV-type methods are usually preferred over FE methods due to the local conservation property [23]. In this section we derive a FV approximation of the problem (1.1) using second- and fourth-order discretization schemes on a 2D cell-centered grid. Our method will follow ideas from [24–27].

We consider the grid $\mathcal{D}_h$ as defined in (2.5) with square blocks of size $h \times h$ denoted by $\mathcal{D}_h^{(i,j)}$ and its boundary as $\partial\mathcal{D}_h^{(i,j)} = \bigcup_{\kappa=1}^{4} \partial\mathcal{D}_h^{(i,j,\kappa)}$ with κ denoting the four cell faces. In each of the finite volumes $\mathcal{D}_h^{(i,j)}$, the integral formulation of (1.1) takes the form

$$\int_{\mathcal{D}_h^{(i,j)}} -\nabla.(a\nabla u)d\mathcal{D} = \int_{\mathcal{D}_h^{(i,j)}} f d\mathcal{D}. \tag{3.1}$$

Using the Gauss divergence theorem, the left-hand side of the above integral is reformulated as a boundary integral for boundary $\partial\mathcal{D}_h^{(i,j,\kappa)}$ and $\mathbf{n}_\kappa$ is the unit normal vector to that face

$$\int_{\mathcal{D}_h^{(i,j)}} -\nabla.(a\nabla u)d\mathcal{D} = -\sum_{\kappa=1}^{4} \int_{\partial\mathcal{D}_h^{(i,j,\kappa)}} a\nabla u.\mathbf{n}_\kappa dS_\kappa. \tag{3.2}$$

For cell ABCD in Fig. 1, the boundary integrals are given by

$$-\sum_{\kappa=1}^{4} \int_{\partial\mathcal{D}_h^{(i,j,\kappa)}} a\nabla u.\mathbf{n}_\kappa dS_\kappa = \int_A^B a\frac{\partial u}{\partial y}dx - \int_B^C a\frac{\partial u}{\partial x}dy - \int_C^D a\frac{\partial u}{\partial y}dx + \int_D^A a\frac{\partial u}{\partial x}dy. \tag{3.3}$$

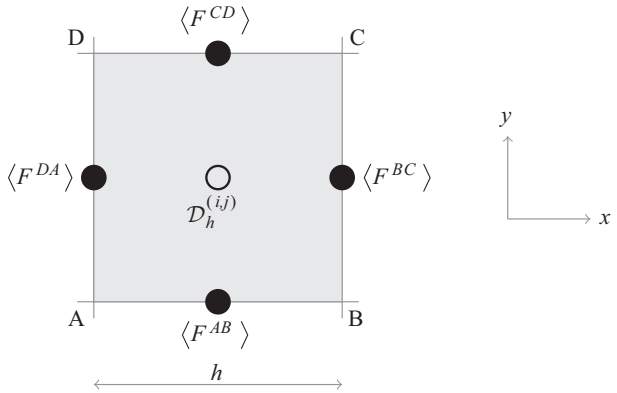

FIG. 1: A volume $\mathcal{D}_h^{(i,j)}$ and face-averaged normal fluxes at face centers

We define the normal flux as $F^\kappa = a\nabla u.\mathbf{n}_\kappa$ and the so-called face-averaged normal flux as $\langle F^\kappa \rangle = 1/h \int_{\partial \mathcal{D}_h^{(i,j,\kappa)}} F^\kappa dS_\kappa$. Now, we can rewrite the integral form of the PDE in terms of face-averaged fluxes at face centers of cell ABCD,

$$h\left[\left\langle F^{AB}\right\rangle - \left\langle F^{BC}\right\rangle - \left\langle F^{CD}\right\rangle + \left\langle F^{DA}\right\rangle\right] = \int_{\mathcal{D}_h^{(i,j)}} f d\mathcal{D}. \tag{3.4}$$

We approximate the above *flux-balance* equation to second- and fourth-order accuracy in Sections 3.1 and 3.2, respectively.

3.1 Second-Order Discretization

To approximate (3.4) to second-order accuracy, we use a central difference scheme to compute the gradient $\nabla u.\mathbf{n}_\kappa$. For the face "BC", for example, we find

$$\left\langle F^{BC}\right\rangle^{(2)} = \frac{1}{h}\int_B^C a\frac{\partial u}{\partial x}dy = \frac{1}{h}\left[a_{i+1/2,j}\left(\frac{u_{i+1,j}-u_{i,j}}{h}\right)h\right] + \mathcal{O}(h). \tag{3.5}$$

Fluxes at other faces are approximated similarly. If the coefficient field is smooth, we can approximate $a_{i+1/2,j}$ by the mean of $a_{i,j}$ and $a_{i+1,j}$. In case of discontinuous coefficients, a harmonic average of $a_{i,j}$ and $a_{i+1,j}$ works well [28]. The right-hand integral in (3.4) is approximated using the midpoint rule $\int_{\mathcal{D}_h^{(i,j)}} f d\mathcal{D} = h^2 f_{i,j} + \mathcal{O}(h^2)$. Finally, we get the discrete equation for volume $\mathcal{D}_h^{(i,j)}$ as

$$-a_{i-1/2,j}u_{i-1,j} - a_{i+1/2,j}u_{i+1,j} + \bar{a}_{i,j}u_{i,j} - a_{i,j-1/2}u_{i,j-1} - a_{i,j+1/2}u_{i,j+1} = h^2 f_{i,j}, \tag{3.6}$$

where $\bar{a}_{i,j} = a_{i-1/2,j} + a_{i+1/2,j} + a_{i,j-1/2} + a_{i,j+1/2}$. This is a standard 5-point stencil. For Dirichlet boundary conditions, i.e., $u = g_D$, we use a one-sided difference instead of central difference in (3.5). A Neumann boundary condition is applied by directly using g_N in place of the finite-difference approximation.

3.2 Fourth-Order Discretization

To explain the fourth-order discretization, we shall weaken the regularity assumptions for the coefficient field and the source term. As our approximations of fluxes and the right-hand side in (3.4) are based on Taylor's expansion, we assume f and a are at least C^4, which guarantees $u \in C^5$. Since our problem involves a Hölder continuous permeabilty field, we will numerically study the convergence of the FV error for different smoothness values ν.

We describe a fourth-order scheme for a regular 2D spatial grid. The following approach can be extended to other more complex grid systems such as mapped coordinates and locally refined grids (see [24,25]). To compute the face-averaged normal fluxes defined in (3.4), we use a fourth-order accurate quadrature rule. We will now explain in detail the computation of the face-averaged normal flux for the face "BC":

$$\left\langle F^{BC}\right\rangle^{(4)} = F^{BC}_{i+1/2,j} + \frac{h^2}{24}\left.\frac{\partial^2 F^{BC}}{\partial y^2}\right|_{i+1/2,j} + \mathcal{O}(h^4), \tag{3.7}$$

where $\left.\partial^2 F^{BC}/\partial y^2\right|_{i+1/2,j}$ is the transverse Laplacian of the flux at the center of face. The above relation can be derived using Taylor's expansion of the flux integrals in (3.3) [27]. We can reduce the above expression to

$$\left\langle F^{BC}\right\rangle^{(4)}_{i+1/2,j} = \langle a\rangle_{i+1/2,j}\left\langle\frac{\partial u}{\partial x}\right\rangle_{i+1/2,j} + \frac{h^2}{12}\frac{\partial a}{\partial y}\left.\frac{\partial^2 u}{\partial y \partial x}\right|_{i+1/2,j} + \mathcal{O}(h^4). \tag{3.8}$$

The above form has a smaller stencil size compared to (3.7). The derivation is provided in Appendix A. Each term in the above expression is computed with fourth-order accuracy.

3.2.1 Computation of $\langle \partial u/\partial x \rangle_{i+1/2,j}$

Using the same relation as (3.7), one can write

$$\left\langle \frac{\partial u}{\partial x} \right\rangle_{i+1/2,j} = \left[\frac{\partial u}{\partial x} + \frac{h^2}{24} \frac{\partial^3 u}{\partial y^2 \partial x} \right]_{i+1/2,j} + \mathcal{O}(h^4). \tag{3.9}$$

Next, using Taylor's expansion, we define

$$\rho_{i+1/2,j} = \frac{1}{24} \left[27(u_{i+1,j} - u_{i,j}) - (u_{i+2,j} - u_{i-1,j}) \right], \tag{3.10}$$

see Fig. 2 (left) for the stencil. We can now use $\rho_{i+1/2,j}$ for computing the elements in (3.9) as

$$\left. \frac{\partial u}{\partial x} \right|_{i+1/2,j} = \frac{\rho_{i+1/2,j}}{h} + \mathcal{O}(h^4), \tag{3.11}$$

$$\left. \frac{\partial^3 u}{\partial y^2 \partial x} \right|_{i+1/2,j} = \frac{1}{h^3} \left[\rho_{i+1/2,j-1} - 2\rho_{i+1/2,j} + \rho_{i+1/2,j+1} \right] + \mathcal{O}(h^2). \tag{3.12}$$

3.2.2 Computation of $\partial^2 u/\partial y \partial x\big|_{i+1/2,j}$

For this term, we use

$$\left. \frac{\partial^2 u}{\partial y \partial x} \right|_{i+1/2,j} = \frac{1}{2h^2} \left[\rho_{i+1/2,j+1} - \rho_{i+1/2,j-1} \right] + \mathcal{O}(h^2). \tag{3.13}$$

Derivations for rest of the terms are provided in Appendix B. This completes the fourth-order accurate face-averaged flux computation for face "BC." Averaged fluxes at other faces are computed analogously. Finally, we obtain a 21-point stencil in a 5×5 block which is centered at the cell on which the flux divergence is computed. In case of 1D

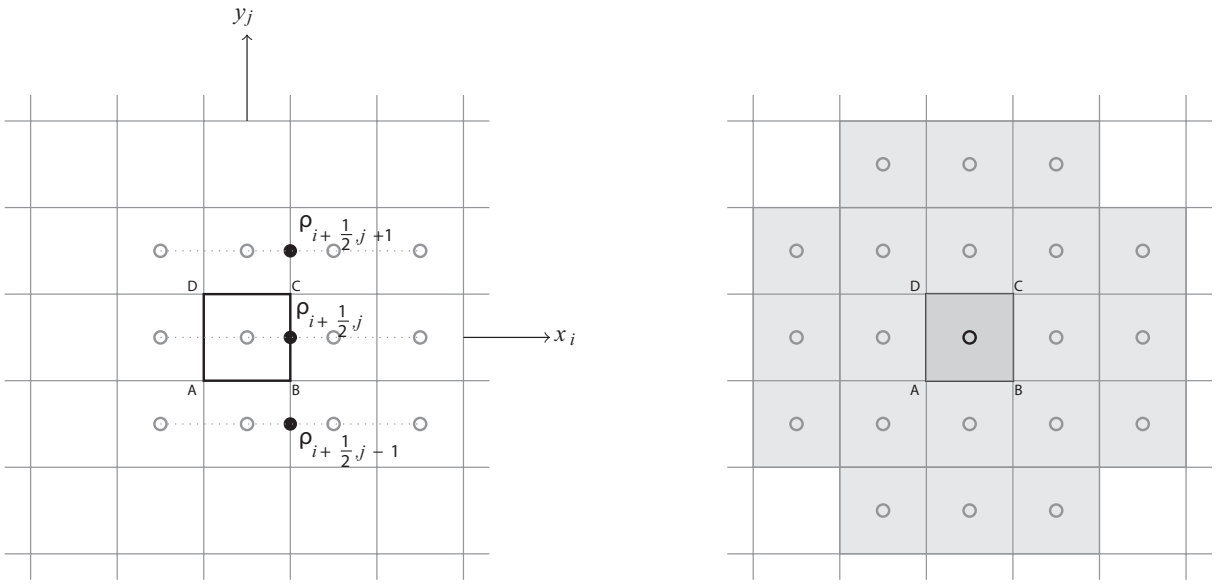

FIG. 2: (Left) Grid points at cell centers required to compute $\langle \partial u/\partial x \rangle_{i+1/2,j}$ using (3.10). (Right) A 5×5 block required to compute the flux divergence of cell ABCD.

problems, we do not have any transverse derivatives and we obtain a 5-point stencil. Regarding the right-hand side of (3.4), we use the fourth-order quadrature rule on the cell $\mathcal{D}_h^{(i,j)}$, i.e.,

$$\int_{\mathcal{D}_h^{(i,j)}} f d\mathcal{D} = h^2 \left[f + \frac{h^2}{24} \Delta f \right]_{i,j} + \mathcal{O}(h^4) \tag{3.14}$$

where Δ is the Laplacian operator computed with second-order accuracy.

3.2.3 Boundary Conditions

We only explain the discretization at the right and bottom boundaries with Dirichlet and Neumann boundary conditions, respectively. The stencil for the other boundaries can be deduced similarly. Further, we assume the same number of nodes (equal to m) along the x and y directions as per the grid definition (2.5). We prefer a ghost cell approach that greatly simplifies the discretization around the faces next to the boundaries. On the right boundary, we use a quartic polynomial extrapolation to update the ghost points $\tilde{u}$ (see Fig. 3) at cell centers. In case of Dirichlet boundary conditions, we use

$$\tilde{u}_{m+1,j} = \frac{1}{35} \left[128 g_{D\,m+1/2,j} - 140 u_{m,j} + 70 u_{m-1,j} - 28 u_{m-2,j} + 5 u_{m-3,j} \right], \tag{3.15}$$

where m denotes the index of the cell centers of the control volumes which are located on the boundary. Furthermore, to compute $\rho_{m+1/2,j}$ at the face centers of a Dirichlet boundary (labeled as $B1$ in Fig. 3), we use a cubic polynomial

$$\rho_{m+1/2,j} = \frac{1}{60} \left[184 g_{D_{m+1/2,j}} - 225 u_{m,j} + 50 u_{m-1,j} - 9 u_{m-2,j} \right]. \tag{3.16}$$

We consider the Neumann boundary conditions at the bottom boundary. The ghost points are updated using the quartic polynomial given by

$$\tilde{u}_{i,m+1} = \frac{1}{22} \left[-24 h g_{N\,i,m+1/2} + 17 u_{i,m} + 9 u_{i,m-1} - 5 u_{i,m-2} + u_{i,m-3} \right]. \tag{3.17}$$

At corners (e.g., point $C1$ in Fig. 3), the ghost points are extrapolated by averaging the values obtained from a quartic polynomial which uses 5 adjacent ghost point values along the two coordinates. The contribution from the x coordinate reads

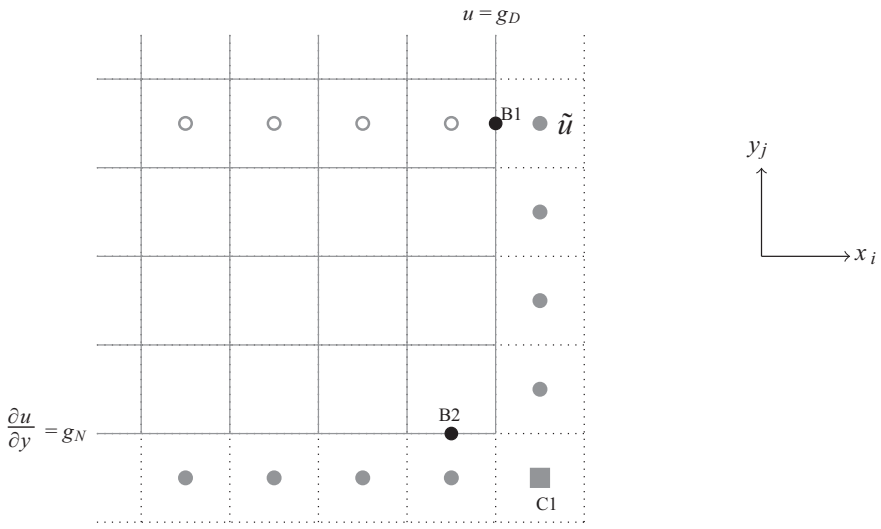

FIG. 3: Points near or at the boundary which require stencil modification to incorporate boundary conditions and ghost points (denoted by the gray circles and square)

$$\tilde{u}_{m+1,0} = 5\tilde{u}_{m,0} - 10\tilde{u}_{m-1,0} + 10\tilde{u}_{m-2,0} - 5\tilde{u}_{m-3,0} + \tilde{u}_{m-4,0}. \tag{3.18}$$

Quantity $\rho_{i,m+1/2} = hg_{N\,i,m+1/2}$ at the face centers of a Neumann boundary. For the corner faces on the boundaries (e.g., point labeled $B2$ in Fig. 3), we approximate the transverse Laplacian $\partial^2 F^{B2}/\partial x^2\big|_{m,1/2}$ and $\partial^2 u/\partial y \partial x\big|_{i+1/2,j}$ using one-sided finite differences.

4. COUPLING OF MLMC WITH FMG (FMG-MLMC)

The most expensive part in a MLMC method is solving the linear system of equations obtained from the discretzation of the PDE (1.1). Due to flexibility and ease of use an optimal "black-box" multigrid solver e.g. AMG1R5 [29] is preferred for this purpose. One of the disadvantages of such solver is that we do not have access to the solutions on the coarser grid levels. For the MLMC estimator that requires samples at different grid hierarchies one can benefit from a geometric full multigrid (FMG) solver based on the MLMC grid hierarchy. The efficiency of a FMG solver comes from using an inexpensive-to-compute solution on a coarser grid as a good initial approximation for the solution on the next grid level. There are twofold benefits of using an FMG solver for a MLMC method. First, these kinds of solvers are asymptotically optimal and can solve the linear system in $\mathcal{O}(h_\ell^{-d})$ operations where d is the spatial dimension. Second, the solution at each FMG level can be utilized to compute samples of the quantity of interest.

Here we benefit from the fact that the permeability field generated using the same random sample $\omega \in \Omega$ on a sequence of mesh $\{\mathcal{D}_{h_k}\}_{k=0}^{\ell}$, $\ell \in \{0, 1, ..., L\}$ essentially represents the same field, but it is sampled at more nodes as we move to the finer grids. Therefore, the FV approximation u_{h_k} on the level k can be utilized to accelerate the solve on the next finer level $k+1$ and at the same time can be used to compute Q_{h_k}.

We use the notation FMG$^\ell$ to denote a FMG solver with grid levels $\{h_k\}_{k=0}^{\ell}$. We now demonstrate how to compute a sample of the quantity of interest at all MLMC levels using the FMG^L solver. First the permeability field is generated at all levels using the same random sample ω_i, $i = 1, 2, ..., N_L$. Using this field, we formulate the linear system at each grid level of FMG^L, i.e.,

$$\mathcal{L}_{h_\ell}(\omega_i) u_{h_\ell} = f_{h_\ell} \quad \text{for} \quad \ell \in \{0, 1, ..., L\}. \tag{4.1}$$

We begin by solving the coarsest grid problem, $\mathcal{L}_{h_0}(\omega_i) u_{h_0} = f_{h_0}$ up to the discretization accuracy and computing the sample $Q_{h_0}(\omega_i)$. Then the prolonged version of this solution $\Pi_{2h}^{h} u_{h_0}$, where Π_{2h}^{h} is the FMG-interpolation operator, is utilized as an initial approximation for the problem on the next level $\ell = 1$. Again, after a few multigrid cycles we obtain the solution u_{h_1} and subsequently compute $Q_{h_1}(\omega_i)$. This is done recursively till the finest level L. This way, based on one FMG solve, we get one sample of the quantity of interest at each grid level, i.e., $Q_{h_0}(\omega_i), (Q_{h_1}(\omega_i) - Q_{h_0}(\omega_i)), ..., (Q_{h_\ell}(\omega_i) - Q_{h_{\ell-1}}(\omega_i)), ..., (Q_{h_L}(\omega_i) - Q_{h_{L-1}}(\omega_i))$. Repeating this process for N_L independent realizations of the random field, we get N_L samples *at all MLMC levels*. Now for the next coarser MLMC level $L-1$, we just need to compute $\tilde{N}_{L-1} = (N_{L-1} - N_L)$ extra samples using the solver FMG^{L-1}. These samples are computed in similar fashion as above.

We note that, for any level $0 \leq \ell < L$, we already have $N_{\ell+1}$ samples from higher levels $\ell+1, ..., L$. Therefore, the remaining samples $\tilde{N}_\ell$ to be computed at level ℓ are given by

$$\tilde{N}_\ell = N_\ell - N_{\ell+1}. \tag{4.2}$$

Note that on the finest level MLMC level L we have $\tilde{N}_L = N_L$. A straightforward computation using (2.13) shows that for levels $0 < \ell < L$, the number of samples grows as:

$$N_\ell = N_{\ell+1} s^{(\beta+\gamma)/2}. \tag{4.3}$$

where s is the grid scaling factor defined earlier. From (4.2) and (4.3), we get the reduction in the number of samples as

$$\frac{\tilde{N}_\ell}{N_\ell} = 1 - s^{-(\beta+\gamma)/2}, \tag{4.4}$$

This shows that the reduction in the number of samples is more pronounced when the rates β and γ are small. In other words, the "recycling" of the coarse grid samples is more effective when N_ℓ decays slowly with level. Figure 4 represents a 4-level FMG-MLMC method using a simplified FMG cycle. The computational cost of the FMG-MLMC estimator is calculated as $\sum_{\ell=0}^{L} \tilde{N}_\ell \mathcal{C}_\ell$. For comparison, the cost of the standard MLMC estimator based on the "black-box" approach will be $N_0\mathcal{C}_0 + \sum_{\ell=1}^{L} N_\ell(\mathcal{C}_\ell + \mathcal{C}_{\ell-1})$. This shows that the FMG approach will always result in some computational gain.

In practice, the FMG-MLMC algorithm can be implemented by slightly modifying the MLMC algorithm presented in [15,21] (see Algorithm 1).

Algorithm 1.

1. Start with $L' = 1$ and run some initial number of FMG solves at level L'.
2. Estimate the sample variance $\mathcal{V}_\ell, \ell = 0, 1, ..., L'$ by using the available number of samples.
3. Using (2.13) compute the optimal number of samples, N_ℓ for all $\ell = 0, 1, ..., L'$ and then using (4.2) compute $\tilde{N}_\ell$.
4. Evaluate the extra $\tilde{N}_\ell$ samples by using the FMG$^\ell$ solver.
5. Test for convergence using the criterion (2.11); if converged, set $L = L'$.
6. If not converged, set $L' = L' + 1$ and go back to step 2.

When using this procedure it is important to ensure that the telescopic identity (1.6) is not violated. For this, we need to confirm that the definitions of Q_{h_ℓ} when computing $\mathbb{E}[Q_{h_\ell} - Q_{h_{\ell-1}}]$ and $\mathbb{E}[Q_{h_{\ell+1}} - Q_{h_\ell}]$ have the same expectation, i.e.,

$$\mathbb{E}[Q_{h_\ell}]^{(coarse)} = \mathbb{E}[Q_{h_\ell}]^{(fine)} \quad \text{for} \quad \ell \in \{0, 1, ..., L-1\}. \tag{4.5}$$

The standard FMG approach where the coarse grid problem is formulated using Galerkin coarsening or a direct discretization of the problem with an restricted coefficient field from fine grid may lead to some additional bias which may decay at a slower rate than the FV error itself. Therefore, to avoid this we recommend generating the coefficient field on coarse grids using the same random vector ω_i. We will present one way of doing this in Section 5.4.

The coarsest level in the FMG solver is decided on the basis of certain stability criteria. In general, the coarsest level should be able to provide a minimum level of resolution to the problem such that it serves as a meaningful initial

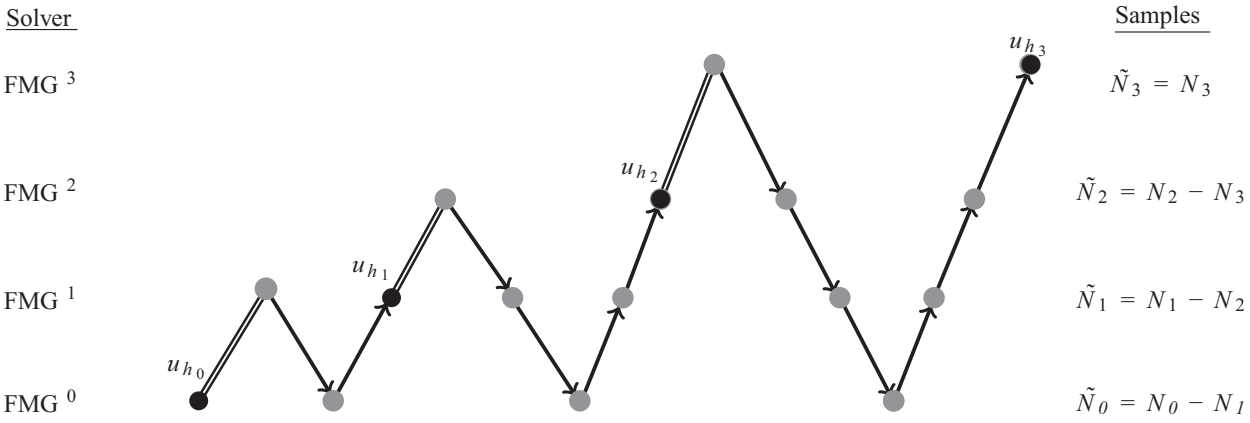

FIG. 4: Schematic representation of a 4-level FMG-MLMC method. The black dots represent the converged solution points and the double lines represent the FMG-interpolation Π_{2h}^{h}. FMG solvers are listed on left and the number of samples to be computed from these solvers is listed on the right.

guess for the next level in the FMG hierarchy. In particular, for the groundwater flow problem the coarsest mesh size should be of order of the correlation length, i.e., $h_0 = \mathcal{O}(\lambda_c)$.

5. FULL MULTIGRID ALGORITHM

Now we describe in detail the construction of a geometric FMG solver for the FV discretization developed in the previous section. For the sake of brevity, we will denote the linear system obtained from second- and fourth-order FV discretization by

$$\mathcal{L}_h^{(2)} u_h = f_h^{(2)}, \tag{5.1}$$

$$\mathcal{L}_h^{(4)} u_h = f_h^{(4)}, \tag{5.2}$$

respectively. The elements of $\mathcal{L}_h^{(4)}$ are provided in Appendix C. The linear systems (5.1) and (5.2) are solved by multigrid methods MG2 and MG4, respectively. Further, the full multigrid variants of MG2 and MG4 are denoted by FMG2 and FMG4, respectively.

Typically, a multigrid solver for elliptic problems is easy to construct. A large number of efficient multigrid algorithms can be found in the literature. For a similar problem, the authors in [28,30] have shown that a multigrid method based on a cell-centered grid using fixed transfer operators that do not depend on matrix coefficients can provide a decent convergence speed even for highly discontinuous coefficients. They use a second-order FV discretization scheme. Currently, there are no multigrid algorithms known for the fourth-order FV discretization on a cell-centered grid with variable coefficients. In this section we describe a robust and scalable fourth-order accurate multigrid solver.

There are different ways of constructing a multigrid solver for a fourth-order discretization. One way is to follow the standard multigrid approach of using an appropriate smoother and transfer operators. Due to the large size of the fourth-order operator $\mathcal{L}_h^{(4)}$, it becomes difficult to find efficient smoothing schemes compared to the second-order operator $\mathcal{L}_h^{(2)}$. Also, transfer operators will be much more complex, especially if we are aiming for independence of coefficient magnitude. A second alternative is to solve the linear system in (5.2) via a *High-Order Defect Correction* (HODC) scheme. This defect correction scheme employs lower-order schemes to obtain higher-order accuracy. Also, from a programmer's point of view it is more convenient to implement the HODC scheme. In our case, we use a second-order multigrid algorithm (MG2) and HODC to construct a fourth-order multigrid solver (MG4). We devote the next section discussing the intricacies in the implementation.

5.1 MG2 Cycle

Our MG2 method resembles the cell-centered multigrid (CCMG) algorithm proposed in [28]. Components required to construct a two-grid cycle are discussed below.

Pre- and post-smoothing: There are different possible choices available for smoothers. In CCMG the ILU smoother works very well but it is quite involved compared to basic iterative smoothers, like Jacobi or Gauss-Seidel smoothers. We will use a Gauss-Seidel Red-Black (GS-RB) smoother in our algorithm.

Defect computation and restriction: The defect $r_h^k = f_h^{(2)} - \mathcal{L}_h^{(2)} u_h^k$ is computed and is restricted to grid $\mathcal{D}_{2h}$ using a bilinear restriction operator $[I_h^{2h}]$ to obtain r_{2h}^k. The stencil form for this restriction reads

$$[I_h^{2h}] = \frac{1}{16} \begin{bmatrix} 1 & 1 & & 0 & 0 \\ 1 & 3 & & 2 & 0 \\ & & * & & \\ 0 & 2 & & 3 & 1 \\ 0 & 0 & & 1 & 1 \end{bmatrix}_h^{2h}.$$

Coarse grid correction: We apply a direct method, e.g., Gaussian elimination, to solve the coarsest grid problem, $\mathcal{L}_{2h}^{(2)} e_{2h}^k = r_{2h}^k$ in a two-grid setting. The prolongation of e_{2h}^k is done using a piece-wise constant operator $[I_{2h}^h]$ to obtain e_h^k which is added to u_h. The stencil reads

$$[I_{2h}^{h}] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}^{h}_{2h}.$$

This process can be easily extended to many grid levels. The transfer operator needs to be modified appropriately at the boundaries. The authors in [30] have done a two-grid *Local Fourier Analysis* (LFA) of this multigrid algorithm for a constant coefficient field. As it is nontrivial to prove convergence for the problem with a random coefficient field, we will here numerically show the convergence of the MG2 cycle.

Remark 2. *For any multigrid cycle, the coarse grid operators can be obtained algebraically via the Galerkin coarse grid method where $\mathcal{L}_{2h}^{(2)} = I_h^{2h}\mathcal{L}_h^{(2)}I_{2h}^h$. The other way to obtain this operator is by direct discretization on the coarse grid using the fine-grid coefficient-field approximated on the coarse grid. We prefer the latter for our multigrid cycle.*

5.2 MG4 Cycle via Defect Correction

In many scenarios, constructing an iterative solver for lower-order discretizations is comparatively easy. A defect correction strategy [20,31,32] can be used to formulate an "outer iteration" where only the right-hand side of the lower-order linear system is modified on the finest grid using the higher-order linear system for the same problem. As the left-hand side of the linear system remains the same, a lower-order multigrid cycle can be utilized to perform an "inner iteration." In this case, we consider the second- and fourth-order linear systems defined in (5.1) and (5.2) and the inner iteration uses the MG2 cycle described in the previous subsection. The kth defect correction iteration is given by

$$\mathcal{L}_h^{(2)}u_h^k = \hat{f}_h, \qquad \text{with} \qquad \hat{f}_h = f_h^{(4)} - \mathcal{L}_h^{(4)}u_h^{k-1} + \mathcal{L}_h^{(2)}u_h^{k-1}. \tag{5.3}$$

We regard this iteration as the MG4 cycle. The new approximation u_h^k is obtained by rearranging

$$u_h^k = (I - (\mathcal{L}_h^{(2)})^{-1}\mathcal{L}_h^{(4)})u_h^{k-1} + (\mathcal{L}_h^{(2)})^{-1}f_h^{(4)}. \tag{5.4}$$

Usually, an initial approximation u_h^0 is provided by a solution obtained from a nested FMG solution of second order. This scheme converges to the solution of fourth-order discrete problem if the spectral radius of iteration matrix is strictly less than one, i.e. $\varrho(I - (\mathcal{L}_h^{(2)})^{-1}\mathcal{L}_h^{(4)}) < 1$. This criterion was verified using the Fourier analysis described in [20] for frozen coefficients (for Poisson's equation).

Remark 3. *At present, we are unable to prove the convergence rate of high-order defect-correction scheme for an oscillatory permeability field. We rely on the numerical results to demonstrate the robustness of the solver with respect to covariance parameters λ_c and σ^2.*

5.3 FMG Structure

Both MG2 and MG4 cycles can be easily incorporated into the FMG hierarchy to obtain FMG2 and FMG4 solvers, respectively.

Typically, the FMG algorithm can be classified as either "fixed" or "accommodative" [33]. In the former approach, the number of cycles is fixed on all FMG levels beforehand whereas in the latter approach the number of cycles is adaptively decided based on some criterion. The accommodative algorithm can be highly effective in optimizing the number of cycles over the FMG levels, especially for problems involving random coefficient fields. In practice, the error cannot be computed but the optimal switching criteria can be based on residual reduction. The switch from grid $2h \to h$ is made when a norm of the residual on grid $2h$ drops below some tolerance η. For example, we can use the constraint

$$\frac{||r_{2h}^k||_2}{||r_{2h}^0||_2} < \eta, \tag{5.5}$$

where $||r_{2h}^0||_2$ and $||r_{2h}^k||_2$ are residuals in 2-norm computed from the initial solution and after k multigrid cycles (MG2 or MG4) at the $2h$ grid level, respectively. A reasonable choice is $\eta = 2^{-p-d}$, where p is the discretization order and d is the spatial dimension. All accommodative algorithms have the disadvantage that they require some

extra work of computing residual norms. Accommodative algorithms are robust. For a coefficient field with large σ^2 and small λ_c, linear systems of equations, (5.1) and (5.2) are highly ill-conditioned. In such cases, the residual norm computations are inexpensive relative to other calculations. In the next section, we perform some experiments to study the dependence of the number of cycles required on different grid levels on the regularity of the PDE solution.

Although the accommodative criterion does not always guarantee a reduction of error to the truncation level, it typically can optimize the work on each level. For coupling of the FMG solver with the MLMC estimator, we need to solve the linear systems on a sequence of grids as defined in (4.1). Therefore, the switch to the next grid level is made only after the solution at the current level has reached the discretization accuracy. Currently, there are no theoretical proofs to compute *a priori* the number of iterations required at FMG levels for problems with random coefficients. Therefore, we use a conservative stopping criterion ($\eta < 10^{-6}$) for residuals on these grids in order to ensure that the solution has reached the level of truncation error.

Another key component of a FMG algorithm is the FMG interpolation. Usually, any high-order scheme gives an accurate solution even on a very coarse grid. Thus, it is important to translate this accuracy to the next finer grid level using an appropriate interpolation scheme for the coarse grid solution. In our algorithm, we use a fourth-order accurate bicubic interpolation (5.6). Due to non-nested nodes in the cell-centered grid hierarchy we get a relatively large interpolation stencil. This interpolation can be seen as a two-step procedure described in Fig. 5. First, we generate auxiliary points at the corners and face centers (denoted by the gray circles) of coarse cells using stencil $[\Pi_{2h}^h]^{(1)}$ and $[\Pi_{2h}^h]^{(2)}$, respectively. In step two, we use the auxiliary and coarse grid nodes to generate points that are the fine grid nodes using the same stencil $[\Pi_{2h}^h]^{(1)}$. Near boundaries and corners, appropriate modifications can be used. In [34], a local Fourier analysis framework is proposed to analyze different components of the FMG algorithm and their effects on the final accuracy.

$$[\Pi_{2h}^h]^{(1)} = \frac{1}{32}\begin{bmatrix} -1 & 0 & & 0 & -1 \\ 0 & 9 & & 9 & 0 \\ & & * & & \\ 0 & 9 & & 9 & 0 \\ -1 & 0 & & 0 & -1 \end{bmatrix} \quad \text{and} \quad [\Pi_{2h}^h]^{(2)} = \frac{1}{16}\begin{bmatrix} -1 & 9 & * & 9 & -1 \end{bmatrix}. \tag{5.6}$$

5.4 Sampling of the Coefficient Field

Several techniques exist to generate samples of the lognormal random field a, including the truncated Karhunen-Loéve (KL) expansion [35] or the circulant embedding method [36]. The main disadvantage of the KL expansion method is that it introduces an additional bias in the approximation of the random field and has to be carefully accounted for in the error model (2.8). The circulant embedding method, however, gives an exact representation of the isotropic

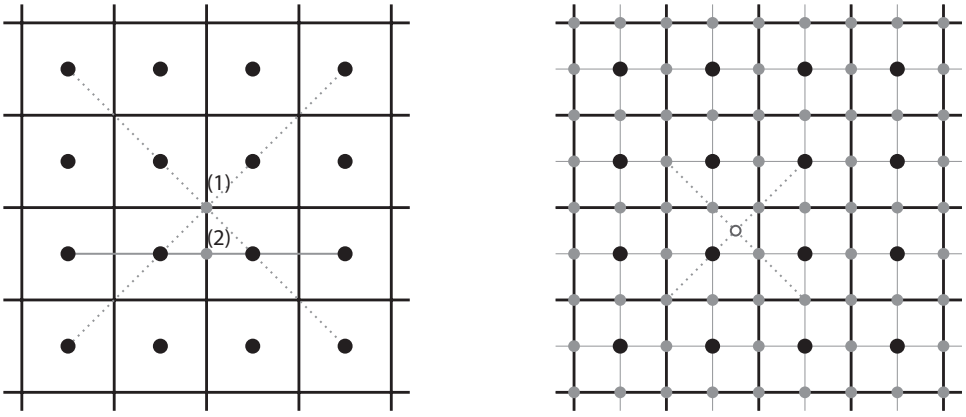

FIG. 5: Schematic representation of FMG-interpolation procedure. (Left) Generation of auxillary points at corners and face centers of coarse cells. (Right) Interpolation at cell centers of fine grid using coarse grid and auxillary points.

random field on a uniform sampling mesh; therefore, we prefer this method for our numerical experiments. This method uses the fast Fourier transform (FFT) for the covariance matrix factorization, thus requires $\mathcal{O}(h_\ell^{-d}\log h_\ell^{-d})$ operations to generate one sample of the random field. For conciseness, we do not give the full details of the circulant embedding technique. Rather, we refer to [11,36,37] for a detailed description.

Here we recommend sampling the field on a grid *twice finer than the finest grid* of the FMG$^\ell$ solver (see Section 4), i.e.,

$$\mathcal{D}^*_{h_\ell/2} = \left\{(x_i, y_j) : x_i = i\frac{h_\ell}{2}, y_j = j\frac{h_\ell}{2}; i, j = 0, 1, ..., m_\ell; m_\ell = \frac{2}{h_\ell}\right\}. \tag{5.7}$$

The above definition describes a vertex-centered grid with a mesh size of $h_\ell/2$. This choice of sampling grid is based on two observations. First, in the FMG-MLMC method we require coefficient fields at all grid levels from the same random sample and all these fields are nested in the grid $\mathcal{D}^*_{h_\ell/2}$. This way we can compute all the discrete operators $\{\mathcal{L}_{h_k}\}_{k=0}^{\ell}$ in the FMG$^\ell$ hierarchy. Second, this also simplifies the FV discretization, as the values of the coefficients are now also available at the face centers.

Alternatively, we can generate coefficients at all grid levels separately, which is slightly cheaper but less convenient than the aforementioned method.

6. NUMERICAL RESULTS

In this section we examine in detail the performance of various components described in the preceding sections. We consider PDE (1.1) on domain $\mathcal{D} \in (0,1)^2$ with a mixed Dirichlet-Neumann boundary condition,

$$g_D(0, y) = 1, \quad g_D(1, y) = 0, \quad \text{and} \tag{6.1}$$

$$g_N(x, 0) = 0, \quad g_N(x, 1) = 0, \quad \text{for} \quad x, y \in (0, 1), \tag{6.2}$$

respectively. For all tests we consider $f = 0$ such that the regularity of u only depends on the Matérn parameters. The samples of the permeability field are generated using the circulant embedding technique (see Section 5.4) on a grid which is twice finer than the finest FMG grid such that we get a nested sequence of the field for all MLMC levels. All numerical schemes are implemented in MATLAB and results are generated on a common workstation.

6.1 Multigrid Convergence

Our first task is to illustrate the convergence rates of the 2D multigrid solver for the second- and fourth-order FV discretizations as described in the preceding sections. A theoretical convergence analysis for the exponential covariance ($\nu = 1/2$) has been conducted in [38] for two-level multigrid along with different choices of smoothers. Here, we consider challenging cases with higher variances and small correlation lengths. For this we choose six different combinations of the Matérn parameters (cf. Table 1) with increasing order of complexity in terms of solvability of the linear system.

For a fixed Φ and h we generate 100 samples of the random field. Then for each sample we run the multigrid as well as the FMG cycles. The reduction factor of a multigrid cycle for the ith realization of the random field is defined as

$$q_i := \left\{\frac{||r_h^k||_2}{||r_h^0||_2}\right\}^{1/k}, \qquad \text{for} \qquad i = 1, 2, ..., 100, \tag{6.3}$$

with $||r_h^0||_2$ as the 2-norm of residual from zero initial guess and $||r_h^k||_2$ be the residual after k multigrid iterations required to achieve the reduction, $||r_h^k||_2/||r_h^0||_2 < 10^{-6}$. Finally, the average reduction factor is computed as

TABLE 1: Different combinations of the Matérn parameters $\Phi = (\nu, \lambda_c, \sigma^2)$ with increasing complexity from left to right

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6
$(2.5, 0.3, 1)$	(1.5,0.3,1)	(0.5,0.3,1)	(2.5,0.1,3)	(1.5,0.1,3)	(0.5,0.1,3)

$$\langle q \rangle = \frac{1}{100} \sum_{i=1}^{100} q_i. \tag{6.4}$$

In all the test cases, we only consider the multigrid V(1,1)-cycle. Further, we fix the coarsest grid based on the correlation length with $h_0 = 1/16$ for $\lambda_c = 0.3, \sigma^2 = 1$ and $h_0 = 1/32$ for $\lambda_c = 0.1, \sigma^2 = 3$.

First we consider the multigrid cycle (MG2) for second-order accurate solution and the corresponding full multigrid cycle (FMG2). The reduction rate of a multigrid V-cycle is governed by the number of multigrid levels as well as by how well the random field is resolved at the multigrid levels. Figure 6(left) presents the average reduction rates, $\langle q \rangle$ with different mesh widths. Additionally, Table 2 provides the numerical value along with the observed standard deviation, σ_q, for the MG2 algorithm. For each parameter set, the average reduction factor is roughly the same for different grid sizes with slight improvement as we move to a finer grid. We also see that the deterioration in the average reduction factor is more pronounced with increasing variance compared to the decreasing smoothness and correlation length. In order to measure the efficiency of the FMG2 cycle, we run a single cycle and check for the residual reduction. If not converged then extra V-cycles are run which are denoted by k^{FMG2}. Table 3 provides the bound for the average numbers of MG2 [FMG2] iterations, $\lceil \langle k \rangle \rceil [\lceil \langle k^{FMG2} \rangle \rceil]$, required to reduce the residual by a factor of 10^{-6}. Also note that the FMG2 method improves as we move to the finer grids for all cases.

Similarly, we investigate the reduction rates of the multigrid cycle (MG4) for fourth-order accurate solution along with its full multigrid variant (FMG4), based on the defect-correction strategy. We follow a similar procedure as above to measure the performance of these solvers. Also, we use the fourth-order solver only for the cases with $\nu > 1$ as we do not expect any improvement for the rough cases (Φ_3, Φ_6) over the second-order discretization. Table 4 reports the average reduction rates to achieve the stopping criteria along with the standard deviations. From Fig. 6(right), we observe that the average reduction rate for MG4 cycle stabilizes around 0.45. The better performance for cases Φ_4, Φ_5 can be attributed to the use of $h_0 = 1/32$ compared to Φ_1, Φ_2 which use $h_0 = 1/16$. Table 5 provides the number of

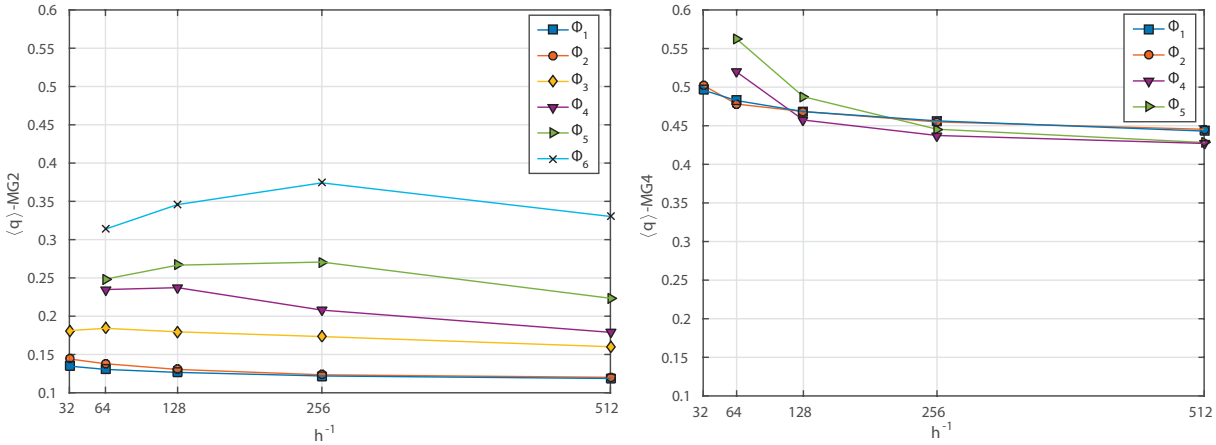

FIG. 6: Average reduction factor for MG2- and MG4-cycles for different h.

TABLE 2: Average reduction factor [standard deviation], $\langle q \rangle [\sigma_q]$, for different grid sizes for MG2 cycles

Φ	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
Φ_1	0.134[0.013]	0.130[0.010]	0.126[0.008]	0.122[0.006]	0.118[0.003]
Φ_2	0.144[0.031]	0.137[0.017]	0.130[0.010]	0.123[0.005]	0.120[0.007]
Φ_3	0.181[0.025]	0.184[0.025]	0.179[0.028]	0.173[0.022]	0.160[0.017]
Φ_4	—	0.234[0.078]	0.237[0.130]	0.208[0.106]	0.178[0.075]
Φ_5	—	0.248[0.087]	0.266[0.125]	0.270[0.169]	0.223[0.123]
Φ_6	—	0.313[0.061]	0.345[0.097]	0.374[0.149]	0.330[0.086]

TABLE 3: Average number of iterations for MG2 [FMG2], $\lceil\langle k\rangle\rceil[\lceil\langle k^{FMG2}\rangle\rceil]$, for different grid sizes

Φ	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
Φ_1	8[3]	8[2]	8[2]	8[1]	7[1]
Φ_2	8[4]	8[3]	8[2]	7[2]	8[1]
Φ_3	9[5]	9[4]	9[4]	9[3]	8[2]
Φ_4	—	11[5]	12[5]	10[2]	9[2]
Φ_5	—	11[5]	13[5]	14[5]	11[3]
Φ_6	—	13[7]	15[7]	18[8]	14[4]

TABLE 4: Average reduction factor [standard deviation], $\langle q\rangle[\sigma_q]$, for different grid sizes for MG4 cycles

Φ	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
Φ_1	0.496[0.038]	0.483[0.026]	0.468[0.021]	0.456[0.017]	0.443[0.019]
Φ_2	0.502[0.041]	0.478[0.030]	0.468[0.025]	0.455[0.021]	0.445[0.021]
Φ_4	—	0.520[0.041]	0.457[0.030]	0.437[0.025]	0.427[0.021]
Φ_5	—	0.562[0.082]	0.488[0.070]	0.446[0.047]	0.428[0.041]

TABLE 5: Average number of iterations for MG4 [FMG4], $\lceil\langle k\rangle\rceil[\lceil\langle k^{FMG4}\rangle\rceil]$, for different grid sizes

Φ	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$	$h = 1/512$
Φ_1	21[2]	20[2]	19[1]	19[1]	18[1]
Φ_2	21[3]	20[2]	19[1]	19[1]	18[1]
Φ_4	—	23[3]	19[2]	18[1]	18[2]
Φ_5	—	26[5]	21[2]	18[2]	17[2]

defect correction steps $\lceil\langle k\rangle\rceil$ for the MG4 method. Using an initial guess from a FMG2 cycle instead of using a zero initial guess can reduce the number of iterations. Next, the quantity $\lceil\langle k^{FMG4}\rangle\rceil$ shows the dependence of the defect correction steps combined with the FMG4 solver on the regularity of the solution. As expected, more iterations are required when the regularity decreases. Interestingly, the FMG4 solver is very efficient as it converges to the stopping criterion in one cycle on finer grids, which is a huge improvement when compared to the number of MG4 iterations that falls around 20. Note that for finer grids such as $1/512$, the residual reduction by a factor of 10^{-6} may not be enough to reach the discretization accuracy and should be further lowered.

A few remarks are in order. Improvement in the average reduction rate was observed with the F(1,1) and W(1,1) cycles but V(1,1) cycle was the fastest to reach the stopping criteria. The combination $\sigma^2 = 3$ and $\lambda_c = 0.1$, using $h_0 = 1/16$, resulted in a very slow convergence and in divergence for some cases. Further, on coarser grids (till 64 $\times$ 64), the performance of the sparse direct solver in MATLAB is superior in terms of CPU times compared to the MG4/FMG4 solvers. This is due to the dominating setup cost compared to the MG4 iterations itself.

6.2 Convergence of the FMG-MLMC Method

In this section, we test the convergence of the FMG-MLMC methods outlined in the previous sections. We denote by FMG2-MLMC and FMG4-MLMC the multilevel estimators obtained from second- and fourth-order discretization, respectively. Further, we denote by MC2 and MC4 the single-level Monte Carlo estimators using the second- and fourth-order discretizations, respectively.

Figure 7 shows the average CPU time required to solve the second- and fourth-order linear systems in 2D using the full multigrid solver for one random sample. The time required to generate one random sample by the circulant embedding technique is also provided. We note that the cost of solving the linear system dominates for the given range of h_ℓ. Therefore, in all our numerical results we will assume that the cost to compute one sample grows with the rate $\gamma \approx d = 2$. Since our code is not optimized, we will use a standardized cost model where we set the cost to compute one sample, $\mathcal{C}_\ell = h_\ell^{-d}$, in order to compare the costs of different estimators rather than using the CPU times.

We will now compute the expected values of different quantities of interest using the FMG-MLMC estimators. We refer readers to [8,13,39] for the convergence proofs for different linear functionals of the solution u of the Darcy flow equation. The first output quantity of interest is the horizontal flux at the center of domain $\vec{x}^* = (1/2, 1/2)^T$, i.e.,

$$Q(u) = -a(\vec{x}, \omega) \left. \frac{\partial u(\vec{x}, \omega)}{\partial x} \right|_{(\vec{x}^*)}, \tag{6.5}$$

where the partial derivative $\partial u(\vec{x}, \omega)/\partial x$ is computed using Taylor's expansion with $\mathcal{O}(h^2)$ accuracy.

In Fig. 8, we show the convergence rate of the FMG2-MLMC method with $\lambda_c = 0.1$, $\sigma^2 = 1$ and smoothness parameter $\nu \in \{0.5, 1, 1.5, 2.5\}$. For each ν, we compute the expectation $|\mathbb{E}[Q_{h_\ell} - Q_{h^*}]|$ and the variance $\mathbb{V}[Q_{h_\ell} - Q_{h^*}]$ from 10,000 independent realizations of the random field using the FMG2 solver with the finest grid $h^* = 1/256$. The purpose here is to show that the approximation at each FMG level has reached the discretization accuracy and can be used as valid samples for the MLMC estimator. Also, we see that the rate of convergence α (left) of the expectation, $|\mathbb{E}[Q_{h_\ell} - Q_{h^*}]|$, improves with increasing smoothness but stalls after $\nu > 1$. The convergence rate β (right) of the variance, $\mathbb{V}[Q_{h_\ell} - Q_{h^*}]$, does however improve with smoothness until β reaches the value 4, which follows from the inequality $\mathbb{V}[Q] \leq \mathbb{E}[Q^2]$. This verifies the inability of second-order schemes to capture additional regularity in the solution obtained when $\nu > 1$.

In Fig. 9, we use the FMG4-MLMC method for the same quantity of interest and compare with FMG2-MLMC method for $\nu = 1.5$ (top) and $\nu = 2.5$ (bottom). Again, the partial derivative $\partial u(\vec{x}, \omega)/\partial x$ is computed using Taylor's expansion with $\mathcal{O}(h^4)$ accuracy. To differentiate between the two methods we denote the convergence rates by $\alpha^{(2)}$,

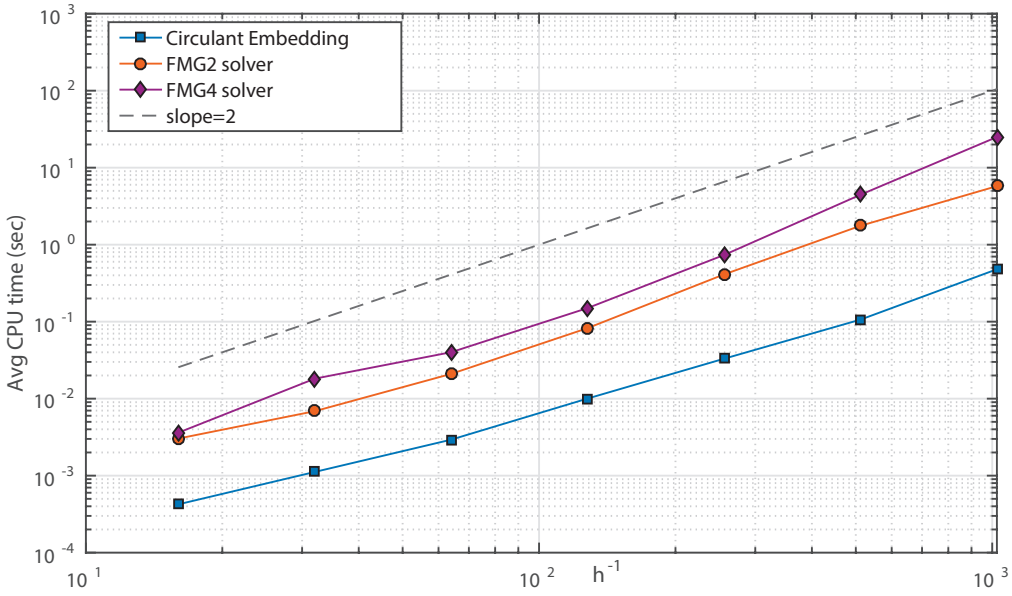

FIG. 7: Average CPU times (in seconds) for the solution of second- and fourth-order discretization along with the time to generate one sample of random field via the circulant embedding method for 2D problem with $\Phi = (1.5, 0.1, 1)$

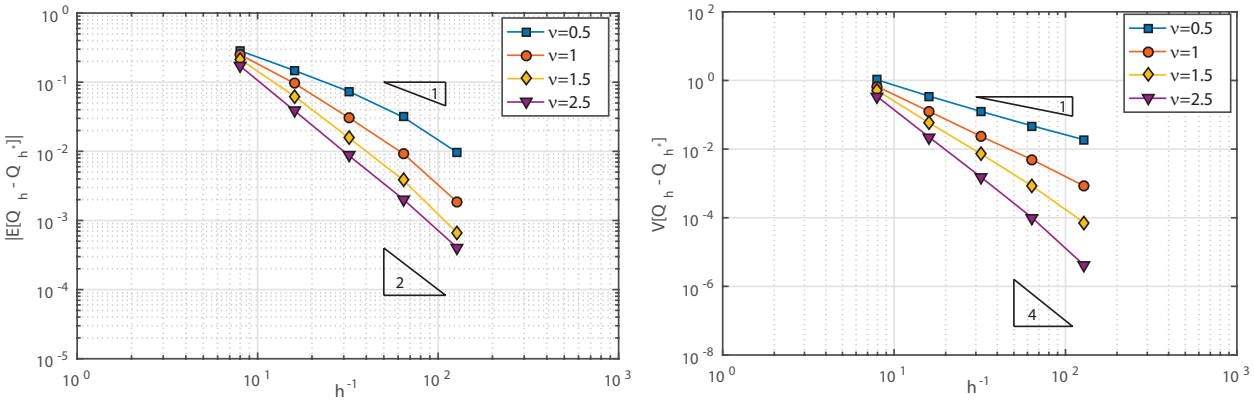

FIG. 8: Parameter α (left) and β (right) observed for horizontal flux $-a\ (\partial u/\partial x)|_{(\vec{x}^*)}$ where $\vec{x}^* = (1/2, 1/2)^T$ from FMG2-MLMC scheme with $\lambda_c = 0.1$, $\sigma^2 = 1$ for different smoothness parameter ν with the reference grid $h^* = 1/256$. The integer in triangles illustrate the convergence rates with respect to to h

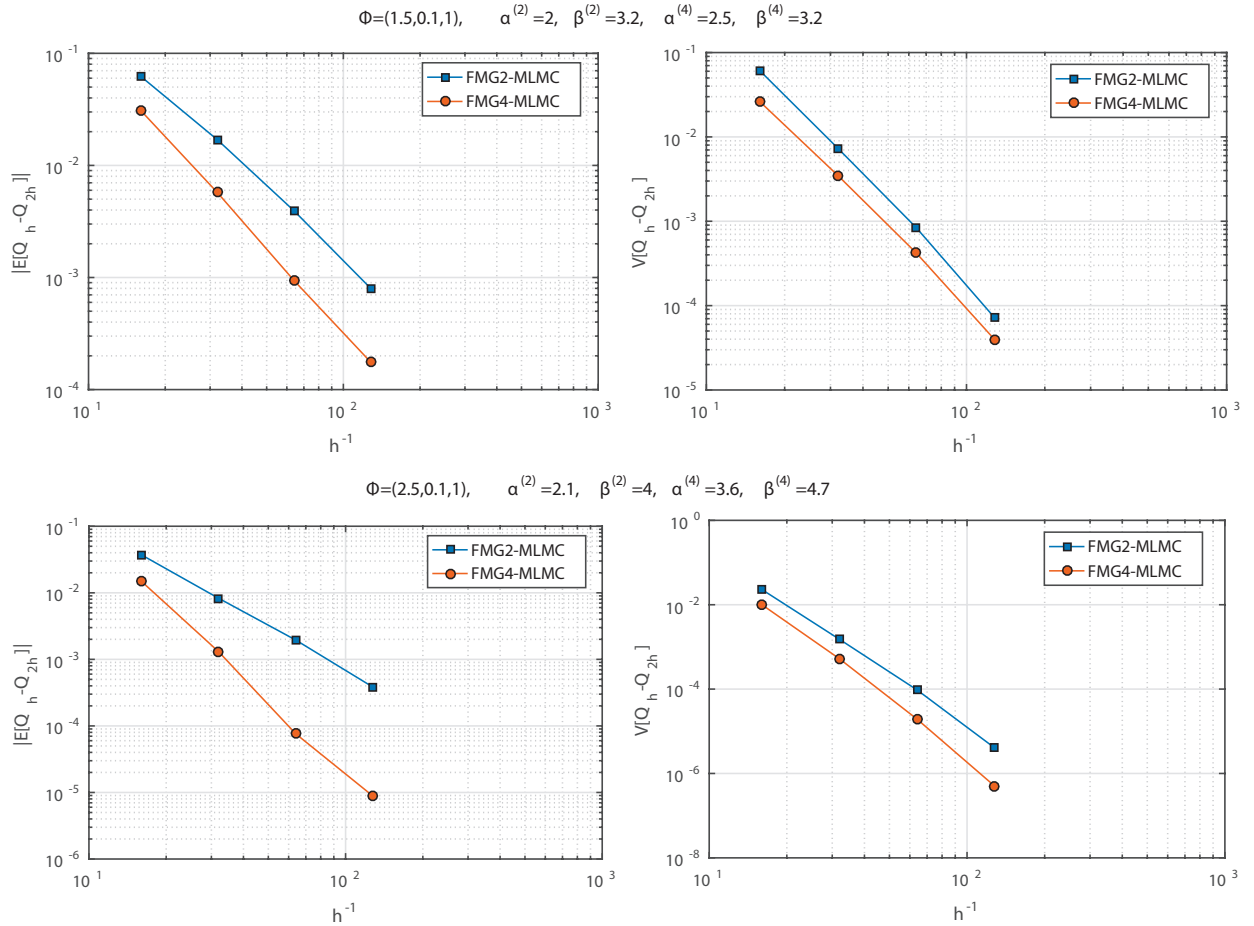

FIG. 9: Comparison of parameter α (left) and β (right) observed for horizontal flux $-a\ (\partial u/\partial x)|_{(\vec{x}^*)}$ where $\vec{x}^* = (1/2, 1/2)^T$ from FMG2-MLMC/FMG4-MLMC scheme.

$\beta^{(2)}$ and $\alpha^{(4)}$, $\beta^{(4)}$ for second- and fourth-order, respectively. The empirical values of these parameters obtained from regression are listed on the top of each figure. We observe that $\alpha^{(4)} > \alpha^{(2)}$ and $\beta^{(4)} > \beta^{(2)}$ for both smoothness values and the difference is more pronounced with increasing smoothness of the random field. Recall that a higher value of α implies fewer MLMC levels and a higher β corresponds to fewer MLMC samples on those levels. Using Theorem 1, the asymptotic cost of the MLMC estimator can be predicted if the parameters α and β are known *a priori*.

In Fig. 10, we compare the standardized cost to compute the expected values of the horizontal flux at the center with different tolerances ε. We use the coarsest grid $h_0 = 1/8$ for both FMG2-MLMC and FMG4-MLMC estimators. These results are produced using Algorithm 1 provided in Section 4. The estimators FMG2-MLMC and FMG4-MLMC converge with the rate $\mathcal{O}(\varepsilon^{-2})$ consistent with the MLMC theorem ($\beta > \gamma = 2$). The cost of the FMG4-MLMC estimator is however significantly lower than other estimators, which can be attributed to higher α and β. The MC2 method converges with the expected rate $\mathcal{O}(\varepsilon^{-3})$, whereas MC4 converges to $\mathcal{O}(\varepsilon^{-2.8})$ (left) and improves slightly with $\mathcal{O}(\varepsilon^{-2.55})$ (right) for the smoother test case. Next, we consider the mean of the solution u in $\mathcal{D}$,

$$Q(u) = \frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} u(\vec{x},\omega)d\mathcal{D}. \tag{6.6}$$

The above integral is approximated using a fourth-order quadrature rule. In Fig. 11, we compare the convergence rates of the two methods for smoothness parameter $\nu = \{1.5, 2.5\}$. The top two plots show the convergence rates for relatively "easier" Matérn parameter $\Phi = (2.5, 0.3, 1)$ with $h_0 = 1/4$ whereas the middle and the bottom two plots present more challenging parameters $\Phi = (1.5, 0.3, 3)$ and $\Phi = (2.5, 0.3, 3)$, respectively, both with $h_0 = 1/8$. For all these test cases, we observe similar improvements in the MLMC parameters as the previous quantity of interest from using the fourth-order discretization. Again, these parameters can be used to predict the asymptotic cost of different estimators.The discretization error from the second-order discretization is already very small ($<10^{-3}$) on the coarser grids. For applications which do not require a very small tolerance, an estimator based on a fourth-order discretization may be more expensive.

In the presence of strong gradients, the constant term in the discretization error can be large. This is also observed in the top left plot of Fig. 11. In these situations, using high-order schemes can also result in an expensive MLMC estimator. The issue can be tackled using a hybrid MLMC estimator which utilizes the second-order scheme on coarser levels and the fourth-order scheme on the finer levels, the cutoff level being the coarsest grid where the fourth-order solution becomes more accurate compared to the second-order solution. This approach is easy to implement and does not lead to any violation of the telescopic sum.

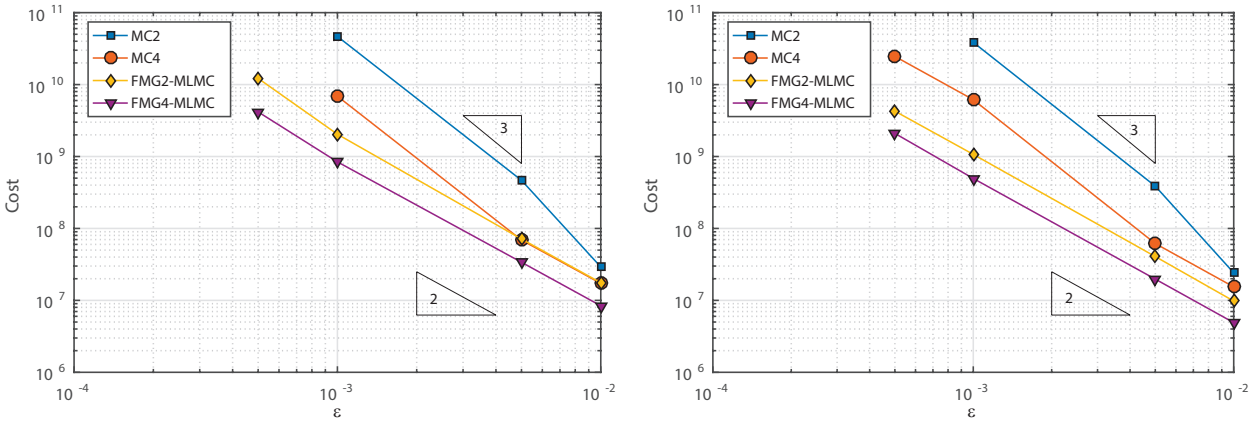

FIG. 10: Cost to obtain RMSE less than ε in the horizontal flux for the 2D problem using different estimators. The covariance parameters are $\Phi = (1.5, 0.1, 1)$ (left) and $\Phi = (2.5, 0.1, 1)$ (right).

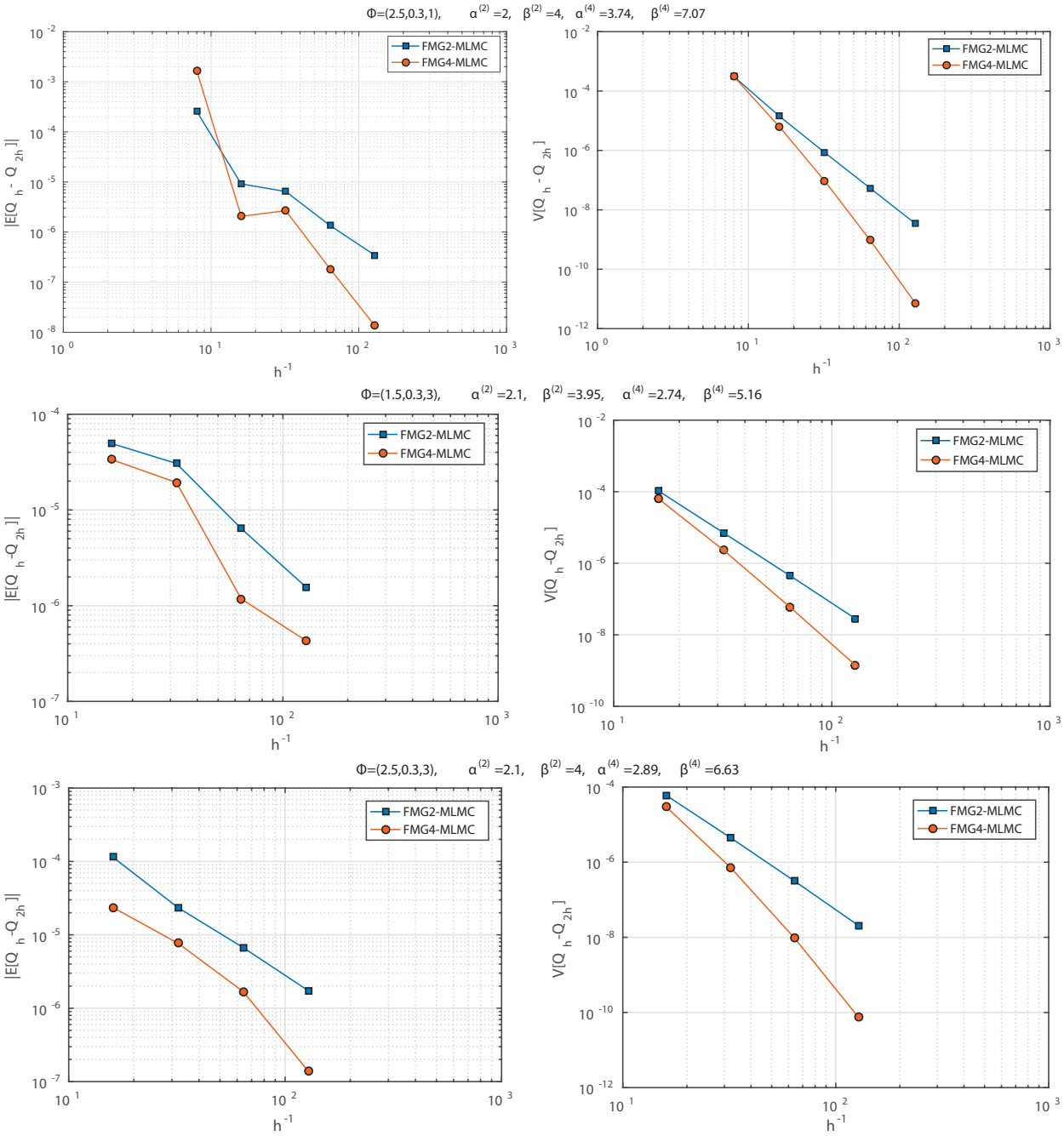

FIG. 11: Comparison of parameter α (left) and β (right) observed for the mean of the solution $1/|\mathcal{D}| \int_{\mathcal{D}} u d\mathcal{D}$ from FMG2-MLMC/FMG4-MLMC scheme.

7. CONCLUSIONS

We have presented a multilevel estimator based on a fourth-order accurate discretization of the stochastic Darcy flow problem with smooth coefficient fields. Our goal was to exploit the additional regularity in the numerical solution of the PDE to achieve a faster MLMC convergence. We utilized a fourth-order FV discretization scheme to approximate

the PDE solution. Additionally, we provided an efficient multigrid algorithm to solve the linear system arising from this discretization. A useful feature of this multigrid algorithm is that it uses a simple second-order multigrid solver combined with a defect correction strategy to obtain fourth-order accurate solution. Further, we showed that this solver is able to handle coefficient fields with large variability and small correlation length scales. Numerical experiments clearly show the benefits of using a high-order discretizations in terms of improved MLMC parameters α and β which dictate the number of MLMC levels and samples, respectively, for smoothness $\nu > 1$. The fourth-order MLMC estimator reached the required tolerance for a much lower cost compared to the estimator based on the second-order discretization. Also, we showed that the fourth-order method always leads to an asymptotic gain in case of the single-level MC estimator.

In this work, we confined ourselves to 2D problems and observed that $\beta > \gamma$ for all quantities of interest leading to an asymptotic cost $\mathcal{O}(\varepsilon^{-2})$. However, for 3D and unsteady problems for which γ is typically large, using a second-order method might lead to situation where $\beta < \gamma$ with the associated MLMC complexity of $\mathcal{O}(\varepsilon^{-2-(\gamma-\beta)/\alpha)})$. In such scenarios a higher α and β from the fourth-order scheme will improve this complexity.

Clearly, one of the issues with high-order schemes is that the cost of assembling and solving the linear system is always more expensive compared to lower-order schemes (for same grid size) and therefore, the computational gains become evident only when the applications requires a relatively small tolerance.

We have also proposed a novel approach to integrate a full multigrid solver with a multilevel estimator with the same mesh hierarchy. The algorithm is described in detail and all modifications required in the FMG solver are discussed. This combination results in computational saving which, however, depends on the rate of decay of the MLMC samples with the levels. Although we confined ourselves to a simplified version of the multigrid solver based on fixed transfer operators, this framework is easily extendable to more sophisticated solvers. Furthermore, the extension of this approach is straightforward to other uncertainty quantification problems in physics and engineering where a multigrid solver is used to solve the sparse linear system and this is the subject of a future publication.

ACKNOWLEDGMENT

This research is funded by the Shell-NWO/FOM program “Computational Sciences for Energy Research” (CSER) under the research Grant no. 14CSER004.

REFERENCES

1. Delhomme, J.P., Spatial variability and uncertainty in groundwater flow parameters: A geostatistical approach, *Water Resour. Res.*, 15(2):269–280, 1979.
2. Freeze, R.A., A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media, *Water Resour. Res.*, 11(5):725–741, 1975.
3. Hoeksema, R.J. and Kitanidis, P.K., Analysis of the spatial structure of properties of selected aquifers, *Water Resour. Res.*, 21(4):563–572, 1985.
4. Handcock, M. and Wallis, J., An approach to statistical spatial-temporal modeling of meteorological fields (with discussion), *J. Am. Stat. Assoc.*, 89:368–390, 1994.
5. Adler, R., *The Geometry of Random Fields*, SIAM Classics ed., Philadelphia: SIAM, 2010.
6. Nobile, F. and Tesei, F., A multilevel Monte Carlo method with control variate for elliptic PDEs with log-normal coefficients, *Stochastic Partial Differential Eq. Anal. Comput.*, 3(3):398–444, 2015.
7. Gilbarg, D. and Trudinger, N., *Elliptic Partial Differential Equations of Second Order*, Grundlehren Math. Wiss. 224, Berlin: Springer, 1983.
8. Charrier, J., Scheichl, R., and Teckentrup, A.L., Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods, *SIAM J. Num. Anal.*, 51(1):322–352, 2013.
9. Galvis, J. and Sarkis, M., Approximating infinity-dimensional stochastic Darcy’s equations without uniform ellipticity, *SIAM J. Num. Anal.*, 47(5):3624–3651, 2009.
10. Sposito, G., Jury, W.A., and Gupta, V.K., Fundamental problems in the stochastic convection-dispersion model of solute transport in aquifers and field soils, *Water Resour. Res.*, 22(1):77–88, 1986.

11. Graham, I., Kuo, F., Nuyens, D., Scheichl, R., and Sloan, I., Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications, *J. Comput. Phys.*, 230(10):3668–3694, 2011.
12. Barth, A., Schwab, C., and Zollinger, N., Multilevel Monte Carlo finite element method for elliptic PDEs with stochastic coefficients, *Num. Math.*, 119(1):123–161, 2011.
13. Teckentrup, A.L., Scheichl, R., Giles, M.B., and Ullmann, E., Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients, *Num. Math.*, 125(3):569–600, 2013.
14. Heinrich, S., Multilevel Monte Carlo methods, Large-Scale Scientific Computing, in *3rd International Conference LSSC 2001, Lecture Notes in Computer Science, 2170, Springer, Sozopol, Bulgaria*, pp. 58–67, 2001.
15. Giles, M., Multilevel Monte Carlo path simulation, *Operations Res.*, 256:981–986, 2008.
16. Giles, M., Multilevel quasi-Monte Carlo path simulation, *Comput. Appl. Math.*, 8:1–18, 2009.
17. Kuo, F.Y., Schwab, C., and Sloan, I.H., Multilevel quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients, *Found. Comput. Math.*, 15(2):411–449, 2015.
18. Kuo, F.Y., Schwab, C., Sloan, I.H., and Ullmann, E., Multilevel quasi-Monte Carlo methods for lognormal diffusion problem, Preprint arXiv:1507.01090v1, Cornell University, 2015.
19. Giles, M., Improved multilevel Monte Carlo convergence using Milstein scheme, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, Berlin: Springer, pp. 343–358, 2008.
20. Trottenberg, U., Oosterlee, C., and Schuller, A., *Multigrid*, San Diego, CA: Elsevier Academic Press, 2000.
21. Cliffe, K., Giles, M., Scheichl, R., and Teckentrup, A., Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients, *Comput. Vis. Sci.*, 14:3–15, 2011.
22. Giles, M.B., Multilevel Monte Carlo methods, *Acta Num.*, 24:259–328, 2015.
23. Dagan, G., *Flow and Transport in Porous Formations*, Berlin: Springer, 1989.
24. Colella, P., Dorr, M., Hittinger, J., and Martin, D., High-order, finite-volume methods in mapped coordinates, *J. Comput. Phys.*, 230:2952–2976, 2011.
25. McCorquodale, P. and Colella, P., A high-order finite-volume method for hyperbolic conservation laws on locally-refined grids, *Commun. Appl. Math. Comput. Sci.*, 61:1–25, 2011.
26. Vinokur, M., An analysis of finite-difference and finite-volume formulations of conservation laws, *J. Comput. Phys.*, 81(1):1–52, 1989.
27. Zhang, Q., Johansen, H., and Colella, P., A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the Advection-diffusion equation, *SIAM J. Sci. Comput.*, 34(2):B179–B201, 2012.
28. Khalil, M. and Wesseling, P., Vertex-centered and cell-centered multigrid for interface problems, *J. Comput. Phys.*, 98:1–10, 1992.
29. Ruge, J. and Stben, K., *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, D.J. Paddon, H. Holstein (Eds.), in *Multigrid Methods for Integral and Differential Equations*, IMA Conference Series, Clarendon Press, Oxford, pp. 169–212, 1985.
30. Mohr, M. and Wienands, R., Cell-centred multigrid revisited, *Comput. Visual. Sci.*, 7:129–140, 2004.
31. Auzinger, W. and Stetter, H., Defect corrections and multigrid iterations, in *Multigrid Methods, Lecture Notes in Mathematics 960*, W. Hackbusch and U. Trottenberg, Eds., Berlin: Springer, pp. 327–351, 1982.
32. Hemker, P., Mixed defect correction iteration for the accurate solution of convection diffusion equation, in *Multigrid Methods, Lecture Notes in Mathematics 960*, W. Hackbusch and U. Trottenberg, Eds., Berlin: Springer, pp. 485–501, 1982.
33. Brandt, A., Guide to multigrid development., in *Multigrid Methods, Lecture Notes in Mathematics 960*, W. Hackbusch and U. Trottenberg, Eds., Berlin: Springer, pp. 220–312, 1982.
34. Rodrigo, C., Gaspar, F.J., Oosterlee, C.W., and Yavneh, I., Accuracy measures and fourier analysis for the full multigrid algorithm, *SIAM J. Sci. Comput.*, 32(5):3108–3129, 2010.
35. Ghanem, M. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach.*, New York: Springer, 1991.
36. Dietrich, C. and Newsam, G., Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix, *SIAM J. Sci. Comput.*, 18:1088–1107, 1997.
37. Wood, A. and Chan, G., Simulation of stationary Gaussian processes in $[0,1]^d$, *J. Comput. Graph. Stat.*, 3:409–432, 1994.

38. Seynaevea, B., Rosseela, E., Nicolab, B., and Vandewallea, S., Fourier mode analysis of multigrid methods for partial differential equations with random coefficients, *J. Comput. Phys.*, 224(1):132–149, 2007.

39. Graham, I.G., Kuo, F.Y., Nichols, J.A., Scheichl, R., Schwab, C., and Sloan, I.H., Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients, *Num. Math.*, 131(2):329–368, 2015.

APPENDIX A. COMPUTATION OF $\left\langle F^{BC}\right\rangle^{(4)}_{i+1/2,j}$

We now derive relations between face-averaged values and face-center point values. We can rewrite (3.7) as

$$\left\langle F^{BC}\right\rangle^{(4)} = a\,\frac{\partial u}{\partial x}\bigg|_{i+1/2,j} + \frac{h^2}{24}\left[\frac{\partial^2 a}{\partial y^2}\frac{\partial u}{\partial x} + 2\frac{\partial a}{\partial y}\frac{\partial^2 u}{\partial y \partial x} + a\frac{\partial^3 u}{\partial y^2 \partial x}\right]_{i+1/2,j} + \mathcal{O}(h^4). \tag{A.1}$$

Thus, the face-averaged flux is expressed in terms of a point-wise value of a and $\partial u/\partial x$ and their derivatives at the center of the face. As the expansion in (3.7), one can also write

$$\langle a\rangle_{i+1/2,j} = a_{i+1/2,j} + \frac{h^2}{24}\,\frac{\partial^2 a}{\partial y^2}\bigg|_{i+1/2,j} + \mathcal{O}(h^4), \tag{A.2}$$

$$\left\langle \frac{\partial u}{\partial x}\right\rangle_{i+1/2,j} = \frac{\partial u}{\partial x}\bigg|_{i+1/2,j} + \frac{h^2}{24}\,\frac{\partial^3 u}{\partial y^2 \partial x}\bigg|_{i+1/2,j} + \mathcal{O}(h^4). \tag{A.3}$$

Hence, if we combine (A.2) and (A.3), we can reduce (A.1) to

$$\left\langle F^{BC}\right\rangle^{(4)}_{i+1/2,j} = \langle a\rangle_{i+1/2,j}\left\langle \frac{\partial u}{\partial x}\right\rangle_{i+1/2,j} + \frac{h^2}{12}\frac{\partial a}{\partial y}\,\frac{\partial^2 u}{\partial y \partial x}\bigg|_{i+1/2,j} + \mathcal{O}(h^4). \tag{A.4}$$

APPENDIX B. COMPUTATION OF $\langle a\rangle_{i+1/2,j}$ AND $\partial a/\partial y|_{i+1/2,j}$

Assuming that the coefficient field is smooth and is sampled at cell-centered locations, we first interpolate the coefficient values at face-centered locations using a centered fourth-order finite difference scheme, i.e.,

$$a_{i+1/2,j} = \frac{1}{16}\left[9(a_{i,j} + a_{i+1,j}) - (a_{i-1,j} + a_{i+2,j})\right] + \mathcal{O}(h^4), \tag{B.1}$$

$$\frac{\partial^2 a}{\partial y^2}\bigg|_{i+1/2,j} = \frac{1}{h^2}\left[a_{i+1/2,j-1} - 2a_{i+1/2,j} + a_{i+1/2,j+1}\right] + \mathcal{O}(h^2). \tag{B.2}$$

We can compute $\langle a\rangle_{i+1/2,j}$ using (A.2). The term $\partial a/\partial y|_{i+1/2,j}$ from (A.4) only needs to be computed with $\mathcal{O}(h^2)$ accuracy as it is multiplied with h^2. Therefore,

$$\frac{\partial a}{\partial y}\bigg|_{i+1/2,j} = \frac{1}{2h}[a_{i+1/2,j+1} - a_{i+1/2,j-1}] + \mathcal{O}(h^2). \tag{B.3}$$

APPENDIX C. ELEMENTS OF $\mathcal{L}^{(4)}_h$

In order to simplify the notation, we use e, w, n, and s to denote the four faces with centers $\left(x_{i+1/2}, y_j\right)$, $\left(x_{i-1/2}, y_j\right)$, $\left(x_i, y_{j+1/2}\right)$, and $\left(x_i, y_{j-1/2}\right)$, respectively. The fourth-order discrete operator away from the boundary can be represented as

$$\mathcal{L}_h^{(4)} = \frac{1}{1152h^2}\begin{bmatrix} 0 & l_{2,-1} & l_{2,0} & l_{2,1} & 0 \\ l_{1,-2} & l_{1,-1} & l_{1,0} & l_{1,1} & l_{1,2} \\ l_{0,-2} & l_{0,-1} & l_{0,0} & l_{0,1} & l_{0,2} \\ l_{-1,-2} & l_{-1,-1} & l_{-1,0} & l_{-1,1} & l_{-1,2} \\ 0 & l_{-2,-1} & l_{-2,0} & l_{-2,1} & 0 \end{bmatrix},$$

with

$$\begin{aligned}
l_{0,0} =& 1188 \sum_{d=e,w,n,s} \langle a \rangle_d , \\
l_{0,-1} =& -44 \langle a \rangle_e - 1188 \langle a \rangle_w + 27 \sum_{d=n,s} (2 \langle a \rangle_d - \nabla^d a_d), \\
l_{0,1} =& -44 \langle a \rangle_w - 1188 \langle a \rangle_e + 27 \sum_{d=n,s} (2 \langle a \rangle_d + \nabla^d a_d), \\
l_{-1,0} =& -44 \langle a \rangle_n - 1188 \langle a \rangle_s + 27 \sum_{d=e,w} (2 \langle a \rangle_d - \nabla^d a_d), \\
l_{1,0} =& -44 \langle a \rangle_s - 1188 \langle a \rangle_n + 27 \sum_{d=e,w} (2 \langle a \rangle_d + \nabla^d a_d), \\
l_{-1,-1} =& \sum_{d=e,n} (2 \langle a \rangle_d - \nabla^d a_d) - 27 \sum_{d=w,s} (2 \langle a \rangle_d - \nabla^d a_d), \\
l_{1,1} =& \sum_{d=w,s} (2 \langle a \rangle_d + \nabla^d a_d) - 27 \sum_{d=e,n} (2 \langle a \rangle_d + \nabla^d a_d), \\
l_{1,-1} =& -27(2 \langle a \rangle_w + \nabla^w a_w) - (2 \langle a \rangle_s - \nabla^s a_s) - (2 \langle a \rangle_e + \nabla^e a_e) - 27(2 \langle a \rangle_n - \nabla^n a_n), \\
l_{-1,1} =& -(2 \langle a \rangle_w - \nabla^w a_w) - 27(2 \langle a \rangle_s + \nabla^s a_s) - 27(2 \langle a \rangle_e - \nabla^e a_e) - (2 \langle a \rangle_n + \nabla^n a_n), \\
l_{0,-2} =& 44 \langle a \rangle_w , \qquad l_{0,2} = 44 \langle a \rangle_e , \qquad l_{-2,0} = 44 \langle a \rangle_s , \qquad l_{2,0} = 44 \langle a \rangle_n , \\
l_{-2,-1} =& 2 \langle a \rangle_s - \nabla^s a_s, \quad l_{-2,1} = 2 \langle a \rangle_s + \nabla^s a_s, \quad l_{2,-1} = 2 \langle a \rangle_n - \nabla^n a_n, \quad l_{2,1} = 2 \langle a \rangle_n + \nabla^n a_n, \\
l_{-1,-2} =& 2 \langle a \rangle_w - \nabla^w a_w, \quad l_{-1,2} = 2 \langle a \rangle_e - \nabla^e a_e, \quad l_{1,-2} = 2 \langle a \rangle_w + \nabla^w a_w, \quad l_{1,2} = 2 \langle a \rangle_e + \nabla^e a_e.
\end{aligned}$$

where $\nabla^d a_d$ denote the gradient of a along the face d and is computed using Eq. (B.3).