MULTILEVEL MONTE CARLO METHODS

ADITYA KARTHIK SARAVANAKUMAR

Abstract. Multilevel Monte Carlo methods are hierarchical variance reduction models widely used to reduce the computational cost of the standard Monte Carlo. The fundamental idea here is to borrow ideas from multigrid methods to accelerate the convergence of classic Monte Carlo methods using a sequence of model discretizations. The computational load is reduced by taking more samples from the cheaper model discretizations and fewer samples from the expensive accurate models. Here a diffusion equation problem with a stochastic diffusion coefficient as input is solved and used to study the variance reduction properties and computational savings of this method. An HDG finite element method scheme is used to generate solutions to the PDE and this allows us to generate model discretizations using both grid refinement and solution order as criteria.

Key words. Multilevel Monte Carlo, Hybridizable Discontinuous Galerkin methods (HDG). Elliptic PDE

1. Introduction. Monte Carlo (MC) methods are algorithms which use repeated random sampling to obtain numerical approximations of various problems in physics. Over the last few decades, these methods have received a lot of attention which has led to the development of various modifications and specializations of the standard MC. Some applications of MC include optimization, fluid problems and uncertainty quantification (UQ).

One of the major challenges with MC is the fact that the probability space for many problems typically tends to be high-dimensional and the MC cost increases rapidly with dimension. Therefore, a key area of research in Monte Carlo is variance reduction methods that aim to provide more precise estimates (smaller confidence intervals) for a given computational cost. Importance sampling, control variate estimates and multi-fidelity MC (MFMC) are examples of extensively studied variance reduction techniques. The control variate method is a classic variance reduction method that takes advantage of the correlation between the RV to be estimated and a dummy random variable to produce savings. MFMC methods apply the control variate principle to a hierarchy of model discretizations to yield computational savings. The multilevel MC method is a popular example of MFMC and it can be used to demonstrate the fundamental ideas of this class of algorithms.

The Multilevel Monte Carlo (MLMC) method was introduced in 2008 [4] and has since gained a lot of traction due to its asymptotic complexity improvement in comparison to the standard MC. The main idea of MLMC involves borrowing ideas from multigrid methods to accelerate the convergence of classic Monte Carlo methods using a sequence of model discretizations. A thorough introduction to MLMC theory and applications can be found in [5]. Later, in [1] the MLMC theory was extended to a PDE with random coefficients. More recently, MLMC has been looked at in the context of estimating the trace of a matrix [3] and in tandem with high-order FEM methods [10].

This work reviews multilevel monte carlo methods and discusses results from an elementary 1D implementation in an HDG finite element solver framework. Section 2 first introduces the MLMC algorithm and the associated theory. Then, the stochastic problem in consideration is described along with a brief overview of the finite element method employed here. A two-level MC algorithm is implemented in Section 3 along with a discussion don't the computational cost analysis. This section also contains results from two-level MC implementations based on grid refinement and solution

order, with comparisons against the standard MC. Finally, Section 4 summarizes the findings of this work and points to active directions of research on this topic.

2. Formulation. In this section, we provide a brief overview of the MLMC theory, the test problem and the finite element method used to solve the PDE.

2.1. MLMC Theory. The Multilevel Monte Carlo method can be thought of as a recursive control variate method. Therefore, let's first recall control variate. Suppose we have a random variable of interest X and are required to estimate its expected value $\mathbb{E}[X]$. Let Y be another random variable and $\mu_Y = \mathbb{E}[Y]$ be the known expected value. We can estimate the expectation of interest indirectly by estimating the expectation of the new random variable Z defined as

$$Z = X + \beta(Y - \mu_Y) \tag{2.1}$$

We can see that the expectation and variance of this new RV are

$$\mathbb{E}[Z] = \mathbb{E}[X] \qquad \text{and} \qquad \mathbb{V}[Z] = \mathbb{V}[X] + \beta^2\mathbb{V}[Y] + 2\beta\text{Cov}[X, Y] \tag{2.2}$$

Solving a quadratic minimization problem to minimize variance yields an optimal value for the parameter β and the corresponding variance as

$$\beta^* = -\frac{\text{Cov}[X, Y]}{\mathbb{V}[Y]} \quad \text{and} \quad \mathbb{V}^*[Z] = \mathbb{V}[X]\left(1 - \rho^2\right) \quad \text{where} \quad \rho = \frac{\text{Cov}[X, Y]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}} \tag{2.3}$$

Notice how the control variate approach leverages the correlation between X and Y to reduce the variance of the estimator.

In the MLMC algorithm, the expectation of X is evaluated as

$$\mathbb{E}[X]_{M_0,\ldots,M_L} = \mathbb{E}[X - X_{N_1}]_{M_0} + \sum_{l=1}^{L-1} \mathbb{E}[X_{N_l} - X_{N_{l+1}}]_{M_l} + \mathbb{E}[X_{N_L}]_{M_L} \tag{2.4}$$

where $\{X_{N_1}, X_{N_2}, \ldots, X_{N_L}\}$ are decreasingly accurate multigrid approximations of X and each term on the RHS evaluated using a standard MC approach. Also, the subscripts M_l shown next to each represent the number of samples used to evaluate each expectation and $M_0 \ll M_1 \ll \cdots \ll M_L$. This way, we use fewer samples from the accurate but expensive model discretizations and more samples from the less accurate but cheaper approximations. Since each term of the form $\mathbb{E}[X_{N_l} - X_{N_{l+1}}]_{M_l}$ is computed for the same set of stochastic realizations, the associated variance is small. Furthermore, since each model discretization is an approximation of the same random variable X, each pair is highly correlated and the corresponding control variate parameter β would be close to unity.

Different criteria can be used to generate the model discretizations $\{X_{N_1}, X_{N_2}, \ldots, X_{N_L}\}$ including grid refinement h, solution order p and the number of basis functions in a K-L expansion of X. The finite element method used in this work allows us to work with both grid refinement and solution order as criteria.

2.2. Test Problem. In order to investigate the convergence properties and computational savings of MLMC methods, we consider the problem of a Poisson equation with a stochastic diffusion coefficient.

$$-\underline{\nabla} \cdot (\kappa(\omega)\,\underline{\nabla} u) = \underline{f} \qquad 0 < x < 1 \tag{2.5}$$

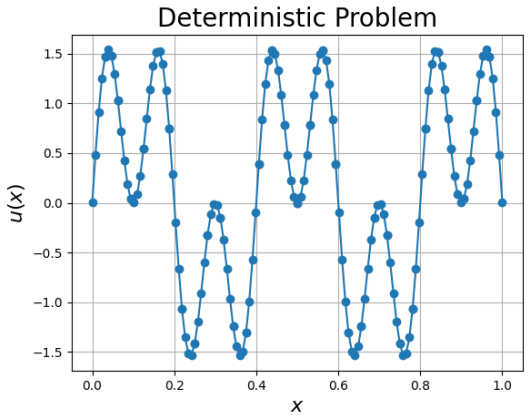

Fig. 1: Exact solution to the deterministic problem corresponding to the test problem - $\kappa(\omega) = 1$

For simplicity, we will assume homogeneous boundary conditions and a log-normally distributed diffusion coefficient that's defined as

$$\kappa(\omega) = \exp(Y(\omega)) \qquad Y \sim \mathcal{N}(0,1) \qquad \text{such that} \qquad \mathbb{E}[\kappa(\omega)] = 1 \tag{2.6}$$

We construct a right-hand side forcing $\underline{f}$ using a linear combination of functions that lies outside the polynomial space such that the numerical method cannot generate an exact solution.

$$\underline{f}(x) = 25\pi^2(\sin(5\pi x) + 9\sin(15\pi x)) \quad 0 < x < 1, \quad u(0) = u(1) = 0 \tag{2.7}$$

and the exact solution to the corresponding deterministic problem is

$$u(x) = \sin(5\pi x) + \sin(15\pi x) \tag{2.8}$$

Figure 1 depicts the solution to the corresponding deterministic problem ($\kappa = 1$) and Fig.2 shows multiple realizations of the solution to the stochastic PDE.

Various numerical methods can be used to discretize the PDE, including finite difference methods and finite volume methods, but in this work, we employ a hybridizable discontinuous Galerkin (HDG) finite element method that allows for easy *hp*-refinement.

2.3. Finite element methods. Finite element methods are robust numerical methods that have gained a lot of traction in the computational science society. These methods allow for *hp*-adaptivity over unstructured meshes that can handle complex geometries, making them useful for a variety of problems. In finite element methods, the domain is discretized into a finite set of elements, and the solution on each element is approximated using a locally defined basis function. The classic FEM schemes are the Galerkin methods, of which the continuous Galerkin (CG) and discontinuous Galerkin (DG) methods are two types. A detailed introduction to these methods can be found in [6]. The HDG methods for convection-diffusion problems were introduced in [8], and they offer computational savings over the DG method by reducing the size

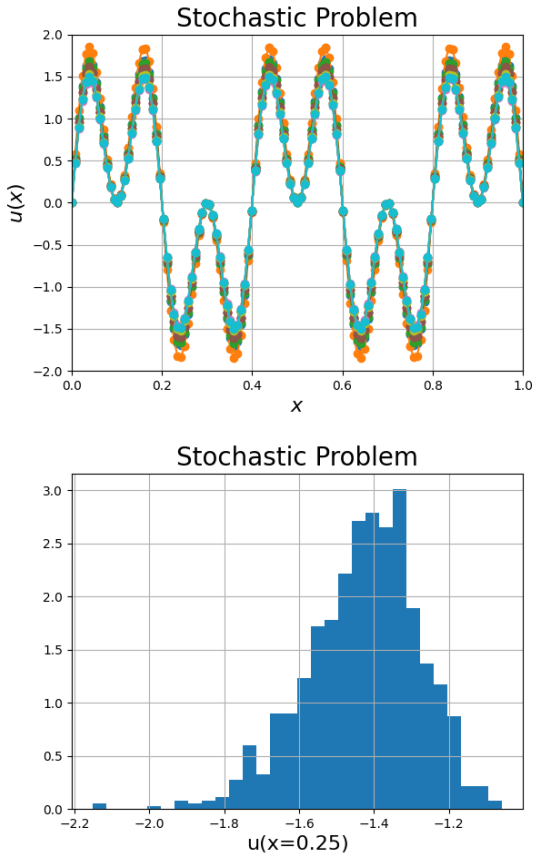

Fig. 2: Multiple realizations of the solution to the stochastic test problem and the corresponding distribution of the solution at $x = 0.25$

of the global linear that needs to be solved. The discretization and flux choices in HDG methods are set up such that the solution across the domain can be computed by solving a global system for the unknowns along the skeleton of the grid ($\underline{\Lambda}$) (referred to as the numerical traces),

$$\underline{\underline{K}}_g \, \underline{\Lambda} = \underline{F}_g \tag{2.9}$$

Following this, these numerical traces are used to reconstruct the local solution on each element.

3. Results. We conduct a few numerical experiments to illustrate the usefulness of MLMC methods. We implement two-level Monte Carlo simulations to estimate the solution to the problem described earlier at the location $x = 0.25$ (without loss of generality) and compare the results to standard MC simulations. The corresponding model discretizations are generated using grid refinement h and solution order p.

For simplicity, we look at the most trivial of MLMC algorithms, the two-level MC algorithm. Here, we estimate the expected value of the solution at our chosen point $u(x = 0.25)$ as

$$\mathbb{E}[X]_{M_1,M_2} = \mathbb{E}[X_1 - X_2]_{M_1} + \mathbb{E}[X_2]_{M_2} \tag{3.1}$$

where X_1 and X_2 correspond to the accurate and expensive estimate, and less-accurate and cheaper estimate, respectively. Note that for the first term on the right-hand side, both X_1 and X_2 are evaluated using the same stochastic realization and we expect the variance associated with this term to be small. The remaining term on the right-hand side should be relatively cheap to compute. More explicitly,

$$\mathbb{E}[X_1 - X_2]_{M_1} = \frac{1}{M_1}\sum_{m=1}^{M_1}\left(X_1(\omega_m^1) - X_2(\omega_m^1)\right) \; ; \; \mathbb{E}[X_2]_{M_2} = \frac{1}{M_2}\sum_{m=1}^{M_2} X_2(\omega_m^2) \tag{3.2}$$

We can extend this analysis to show that the variance of the two-level MC estimate is given by

$$\mathbb{V}_{M_1,M_2}[X] = \frac{\mathbb{V}[X_1 - X_2]}{M_1} + \frac{\mathbb{V}[X_2]}{M_2} = \mathbb{E}[\xi_1 - \xi_2]_{M_1} + \mathbb{E}[\xi_2]_{M_2} \tag{3.3}$$

where $\xi_1 = (X_1 - \mathbb{E}[X]_{M_1,M_2})^2$ and $\xi_2 = (X_2 - \mathbb{E}[X]_{M_1,M_2})^2$. The expectations are computed according to the previous equation.

In order to make a fair comparison between the two-level and standard MC estimators, we need to understand the associated computational costs. Let C_1 and C_2 represent the cost associated with computing a single realization of X_1 and X_2, respectively. Note that the first term on the right-hand side of Eq.3.1 requires the computation of both X_1 and X_2. Therefore, the total cost Λ of associated with a standard N-sample MC estimate and a two-level $\{M_1, M_2\}$-sample estimate would be

$$\Lambda_{standard} = NC_1 \qquad \text{and} \qquad \Lambda_{two-level} = M_1(C_1 + C_2) + M_2C_2 \tag{3.4}$$

It has been shown that to obtain optimal variance reduction [5], the number of samples for each term in the two-level algorithm (M_1, M_2) needs to be chosen according to

$$\frac{M_1}{M_2} = \frac{\sqrt{V_1/(C_1 + C_2)}}{\sqrt{V_2/C_2}} \tag{3.5}$$

where V_1 and V_2 represent the variance associated with first and second terms on the right-hand side of Eq.3.1.

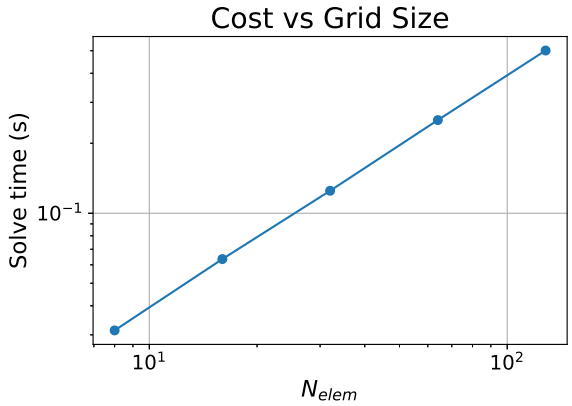

Fig. 3: Computational cost per realization as a function of grid size for a fixed solution order $p = 3$

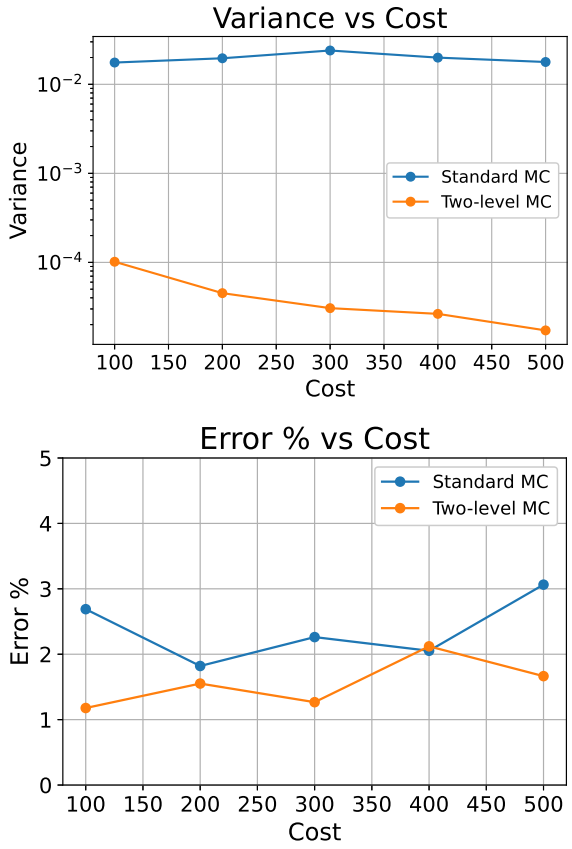

Fig. 4: Comparison of variance associated with the standard MC and two-level MC estimators for different total costs N - Grid Refinement

N	M_1	M_2	Error
100	23	208	1.18%
200	45	420	1.55%
300	67	632	1.27%
400	89	844	2.12%
500	112	1052	1.66%

Table 1: Optimal values for M_1 and M_2 for different computational budgets N and the corresponding errors - Grid Refinement

3.1. Two-level MC - Grid Refinement. To construct our two model discretizations X_1 and X_2, we consider two grids with 32 and 16 elements ($h = 0.03125$ and $h = 0.0625$) and use a constant solution order of $p = 3$. We first estimate the cost associated with generating one realization of the solution as a function of the grid size h and the results are shown in Fig.3. We notice a nearly-linear trend in the computational cost with respect to the grid size $C_1/C_2 \approx 2$. We also execute numerical experiments to empirically estimate the variance ratio to be $V_1/V_2 \approx 225$. The high value of this ratio is consistent with our prediction that the variance of $X_1 - X_2$ would be small. Using these values, we can compute and use the optimal ratio $(M_1/M_2)^*$ for a given fixed computational budget N (Table 1). In Fig.4-a, we compare the variance produced by the two-level MC estimate with that produced by a standard MC estimator for different fixed computational budgets. We see that the two-level estimator does indeed produce a significantly reduced variance for the same cost when compared to the standard MC estimator. Furthermore, we also notice that the errors made by the two estimators are similar in magnitude (Fig.4-b). We then also run a few simulations with different values for $M_1/M_2 = \lambda(M_1/M_2)^*$ (fixed cost - $N = 300$) to verify the optimality of this choice and the results are shown in Fig.5

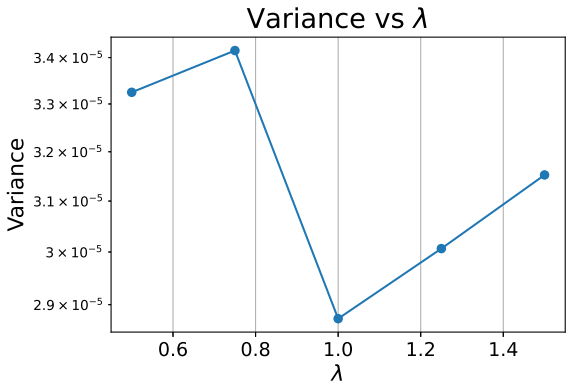

Fig. 5: Comparison of variance produced by the two-level MC for different $M_1/M_2 = \lambda(M_1/M_2)^*$ ratios - Grid Refinement

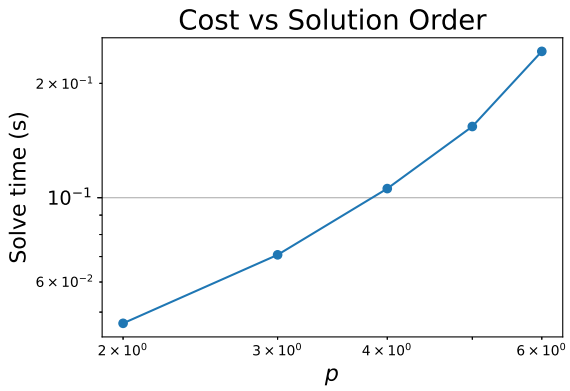

Fig. 6: Computational cost per realization as a function of solution order for a fixed grid size $N_{elem} = 16$

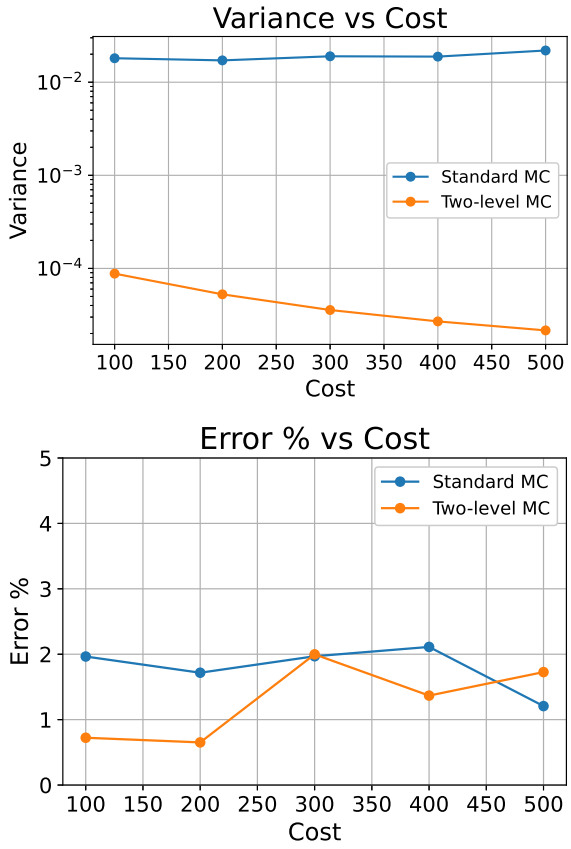

Fig. 7: Comparison of variance of standard MC and two-level MC for different total costs N - Solution Order

N	M_1	M_2	Error
100	6	229	0.72%
200	12	458	0.65%
300	17	691	1.99%
400	23	920	1.37%
500	28	1152	1.72%

Table 2: Optimal values for M_1 and M_2 for different computational budgets N and the corresponding errors - Solution Order

3.2. Two-level MC - Solution Order. We can also generate our model discretizations using the solution order as the criterion. Here we generate solutions to our stochastic PDE on a 16-element grid using solution orders $p = 3$ (X_1) and $p = 2$ (X_2). Once again for comparison purposes, we first estimate the cost associated with generating one realization of the solution as a function of the grid size p and the results are shown in Fig.6. In contrast to the grid refinement case, we notice a cost ratio of approximately $C_1/C_2 \approx 1.5$. Numerical experiments allow us to empirically estimate the variance ratio to be $V_1/V_2 \approx 500$. Once again, the variance of the difference term $X_1 - X_2$ is low as we expect. The optimal values for M_1 and M_2 are then computed and used for each given fixed computational budget N (Table 2). In Fig.7-a, we compare the variance produced by the two-level MC estimate with that produced by a standard MC estimator for different fixed computational budgets. We find that the variance reduction produced by the solution order-based two-level MC is quite similar to that produced by the grid-refinement-based two-level MC. We also check to ensure that the errors associated with the two estimators are similar in magnitude (Fig.7-b). Finally, we once again verify the optimality of the $(M_1/M_2)^*$ choice with a few simulations varying M_1 and M_2 as $M_1/M_2 = \lambda(M_1/M_2)^*$ (fixed cost - $N = 300$) and the results are shown in Fig.8.

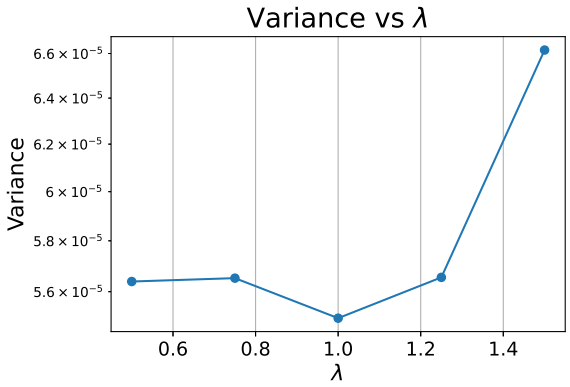

Fig. 8: Comparison of variance produced by the two-level MC for different $M_1/M_2 = \lambda(M_1/M_2)^*$ ratios - Solution Order

4. Conclusion. Multilevel Monte Carlo is a prototypical example of multi-fidelity Monte Carlo methods and serves as a demonstrative tool to understand the underlying ideas. The numerical experiments conducted in this work empirically verify the variance reduction produced by MLMC and the optimality of the sample ratio choice $M_1/M_2 = \sqrt{V_1/(C_1+C_2)}/\sqrt{V_2/C_2}$. We have only considered the two-level MC algorithm here, but this can be easily extended to more sophisticated multi-level MC implementations that could produce even more precise estimates. Another exciting extension could be to look at how one can optimally combine both grid refinement and solution order to form the model discretizations.

REFERENCES

[1] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), p. 3, https://doi.org/10.1007/s00791-011-0160-x, https://doi.org/10.1007/s00791-011-0160-x.

[2] N. Frikha, *Multi-level stochastic approximation algorithms*, The Annals of Applied Probability, 26 (2016), pp. 933–985, https://doi.org/10.1214/15-AAP1109, https://projecteuclid.org/journals/annals-of-applied-probability/volume-26/issue-2/Multi-level-stochastic-approximation-algorithms/10.1214/15-AAP1109.full. Publisher: Institute of Mathematical Statistics.

[3] A. Frommer, M. N. Khalil, and G. Ramirez-Hidalgo, *A Multilevel Approach to Variance Reduction in the Stochastic Estimation of the Trace of a Matrix*, Aug. 2021, https://doi.org/10.48550/arXiv.2108.11281, http://arxiv.org/abs/2108.11281. arXiv:2108.11281 [hep-lat].

[4] M. B. Giles, *Multilevel Monte Carlo Path Simulation*, Operations Research, 56 (2008), pp. 607–617, https://doi.org/10.1287/opre.1070.0496, https://pubsonline.informs.org/doi/10.1287/opre.1070.0496. Publisher: INFORMS.

[5] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numerica, 24 (2015), pp. 259–328, https://doi.org/10.1017/S096249291500001X, https://www.cambridge.org/core/journals/acta-numerica/article/multilevel-monte-carlo-methods/C5AF9A57ED8FF8FDF08074C1071C5511. Publisher: Cambridge University Press.

[6] J. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, vol. 54, Jan. 2007.

[7] M. Jornet, J. Calatayud, O. P. L. MaÃ®tre, and J. C. CortÃ©s, *Variance reduction methods and multilevel monte carlo strategy for estimating densities of solutions to random second order linear differential equations*, International Journal for Uncertainty Quantification, 10 (2020), https://doi.org/10.1615/Int.J.UncertaintyQuantification.2020032659, https://www.dl.begellhouse.com/journals/52034eb04b657aea,4c71a7d5794d3085,39d5c2b15f4155ff.html. Publisher: Begel House Inc.

[8] N. C. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for linear convectionâdiffusion equations*, Journal of Computational Physics, 228 (2009), pp. 3232–3254, https://doi.org/10.1016/j.jcp.2009.01.030, https://www.sciencedirect.com/science/article/pii/S0021999109000308.

[9] D. Schaden and E. Ullmann, *On multilevel best linear unbiased estimators*, SIAM/ASA Journal on Uncertainty Quantification, 8 (2020), pp. 601–635, https://doi.org/10.1137/19M1263534, https://doi.org/10.1137/19M1263534, https://arxiv.org/abs/https://doi.org/10.1137/19M1263534.

[10] F. Vidal-Codina, N. C. Nguyen, M. B. Giles, and J. Peraire, *A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations*, Journal of Computational Physics, 297 (2015), pp. 700–720, https://doi.org/10.1016/j.jcp.2015.05.041, https://www.sciencedirect.com/science/article/pii/S0021999115003757.