*13th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP13*
*Seoul, South Korea, May 26-30, 2019*

# Multilevel sampling with Monte Carlo and Quasi-Monte Carlo methods for uncertainty quantification in structural engineering

Philippe Blondeel
*Dept. of Computer Science, KU Leuven, Leuven, Belgium*

Pieterjan Robbe
*Dept. of Computer Science, KU Leuven, Leuven, Belgium*

Cédric Van hoorickx
*Dept. of Civil Engineering, KU Leuven, Leuven, Belgium*

Geert Lombaert
*Dept. of Civil Engineering, KU Leuven, Leuven, Belgium*

Stefan Vandewalle
*Dept. of Computer Science, KU Leuven, Leuven, Belgium*

ABSTRACT: Practical structural engineering applications tend to exhibit a certain degree of uncertainty in their material parameters, loading forces and so forth. As such, the accurate quantification of the effect of those uncertainties is of capital importance. The standard Monte Carlo method is one of the most common sampling methods used to compute this effect. In this paper we compare two extensions of the standard Monte Carlo method: the Multilevel Monte Carlo (MLMC) and the Multilevel Quasi-Monte Carlo (MLQMC) method. These two methods are tested on a structural engineering problem: a cantilever steel beam clamped at both sides and loaded in the middle, with an uncertain Young's modulus. A Gamma random field is used to model the uncertainty. For the response of the beam we consider its spatial displacement in the elasto-plastic domain. Our aim is to demonstrate the effectiveness and versatility of both MLMC and MLQMC by coupling them with this Finite Element code. We show that MLQMC has a lower computational cost than MLMC for a desired tolerance on the root mean square error. Furthermore both methods are significantly faster than a standard Monte Carlo method.

## INTRODUCTION

Models of practical structural engineering applications often exhibit a certain degree of uncertainty. This uncertainty can be located in the dimension of the considered model, the magnitude of the loading forces, material parameters and so forth. In order to address this uncertainty and its propagation through the model, two main types of methods exist: non-sampling methods and sampling methods. The category of non-sampling methods encompasses, among others, the Stochastic Galerkin Finite Element method, see Ghanem and Spanos (2003). This method, whilst being highly effective, is also highly intrusive and memory demanding. It transforms the uncertain coefficient partial differential equation into a large system of coupled deterministic PDEs by means of a Galerkin projection technique. Sampling methods, on the other hand, are generally non intrusive. A well known example of such a sampling method is the Monte Carlo (MC) method. A major drawback however, is the slow

convergence in function of the number of samples. This can be alleviated by variance reduction techniques, such as Multilevel Monte Carlo, or by improved sampling methods, such as Latin Hypercube sampling, see Loh (1996), and Quasi-Monte Carlo, see Caflisch (1998); Niederreiter (2004). Multilevel Monte Carlo was first proposed by Heinrich (2001) and successfully applied in a finance context for option pricing by Giles (2008).

It is in this framework that we present our work. In work published in conference proceedings by Blondeel et al. (2018) and subsequently extended in the paper by Blondeel et al. (2018), we compared standard Monte Carlo with Multilevel Monte Carlo for a structural engineering application. We observed a speedup of a factor ten in favor of Multilevel Monte Carlo. In the present paper we now combine a non-linear MATLAB Finite Element solver with a JULIA Multilevel Monte Carlo and Multilevel Quasi-Monte Carlo implementation by Robbe et al. (2017, 2018). By doing so, we demonstrate the performance of both methods and compare them in terms of simulation time. The engineering problem we consider, is a steel beam, clamped at both sides and loaded in the middle. The Young's modulus is considered uncertain and modeled by means of a Gamma random field.

## CASE PRESENTATION

This part introduces the beam characteristics, the uncertainty modeling and the Finite Element solver.

### *Beam Characteristics*

We consider a steel beam clamped on both sides subjected to a static force applied at the center, see Figure 1. The beam has the following dimensions and material parameters: a length of 2.5 m, a height of 0.25 m and a thickness of $10^{-3}$ m, with a yield strength of 240 MPa, a Poisson ratio of 0.25 and a Young's modulus with a mean value of 200 GPa.

### *Uncertainty Modeling*

The uncertainty resides in the beam's Young's modulus. In order to model this uncertainty, we opt for a heterogeneous representation: the uncertainty will be represented by a spatially varying Gamma Random Field. This field is constructed by means of a Karhunen-Loève (KL) expansion, see Loève (1977), followed by a memoryless transformation, see Grigoriu (1998).

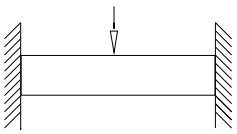

*Figure 1: Cantilever beam clamped at both sides loaded in the middle.*

We consider a Gaussian random field $Z(\mathbf{x},\omega)$, with an exponential covariance kernel

$$C(\mathbf{x},\mathbf{y}) := \sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_p}{\lambda}\right), \qquad (1)$$

where the correlation length $\lambda = 0.3$, the standard deviation $\sigma = 1.0$, and the 1-norm, i.e., $p = 1$ are selected. The KL expansion can be formulated as

$$Z(\mathbf{x},\omega) = \overline{Z}(\mathbf{x},.) + \sum_{n=1}^{\infty} \sqrt{\theta_n}\xi_n(\omega)b_n(\mathbf{x}). \qquad (2)$$

The obtained Gaussian random field, $Z(\mathbf{x},\omega)$ with a mean value of $\overline{Z}(\mathbf{x},.)$, depends on respectively the eigenvalues, $\theta_n$, and the eigenfunctions, $b_n(\mathbf{x})$, of the covariance kernel Eq. (1). $\xi_n(\omega)$ denotes the i.i.d standard normal random variables. For a one dimensional domain [0,1], the eigenvalues and eigenfunctions can be computed analytically according to

$$\begin{aligned} \theta_n^{1D} &= \frac{2\lambda}{\lambda^2 w_n^2 + 1} \quad \text{and} \\ b_n^{1D}(x) &= A_n\left(\sin(w_n x) + \lambda w_n \cos(w_n x)\right), \end{aligned} \qquad (3)$$

see, e.g., Cliffe et al. (2011). The normalizing constants $A_n$ are chosen such that $\|b_n\|_2 = 1$. The constants $w_n$ represent the real solutions, in increasing order, of the transcendental equation

$$\tan(w) = \frac{2\lambda w}{\lambda^2 w^2 - 1}. \qquad (4)$$

For the two-dimensional case, the eigenvalues and functions are obtained in a tensorproduct way,

$$\begin{aligned} \theta_n^{2D} &= \theta_{i_n}^{1D}\theta_{j_n}^{1D} \quad \text{and} \\ b_n^{2D}(\mathbf{x}) &= b_{i_n}^{1D}(x_1)b_{j_n}^{1D}(x_2) \;\text{ with }\; n = (i_n, j_n). \end{aligned} \qquad (5)$$

In practice, the infinite sum in Eq. (2) is truncated to a finite value. Here, we chose to truncate the sum after 101 terms, effectively capturing 90 % of the variance of the random field.

This obtained Gaussian Random field is then transformed to a Gamma Random field by means of a point wise memoryless transformation, see Grigoriu (1998). The characteristics of the Gamma distribution are as follows: a shape parameter $\alpha = 934.2$ and a scale parameter $\beta = 0.214 \times 10^9$, see Hess et al. (2002), leading to a mean value of 200 GPa and a standard deviation of 6.543 GPa .

*Finite Element Modeling*

In order to compute the spatial displacement of the beam in the elasto-plastic region, a MATLAB plane stress Finite Element code is used. The beam is discretized by means of an equidistant regular rectangular mesh consisting of four node isoparametric quadrilateral elements. A short overview of the solution method is presented below.

In the elasto-plastic domain, the stress-strain relationship is non-linear. An incremental load approach is used, in which the plastic region is governed by the von Mises yield criterion with isotropic linear hardening. Starting with a force of 0 N, this force is increased in steps of 135 N until the total force of 13.5 kN is reached. For each of these force steps, the spatial displacement of the beam is computed using a return mapping algorithm, see Perez-Foguet et al. (2001). Details about the implementation of this algorithm can be found in Chapter 2 §4 and Chapter 7 §3 and §4 of de Borst et al. (2012) and in the paper by Blondeel et al. (2018).

## METHODOLOGY

This section gives a short overview of the Multilevel Monte Carlo and Multilevel Quasi-Monte Carlo methods. A more detailed discussion can be found in Giles (2008); Giles and Waterhouse (2009)

*Multilevel Monte Carlo*

The Multilevel Monte Carlo (MLMC) method is an extension of the standard Monte Carlo (MC) method, which employs a hierarchy of levels in order to achieve a variance reduction. By employing such a hierarchy and exploiting the telescoping sum identity given below, see Eq. (7), the cost of the estimation is significantly reduced. Here, the levels are defined as a geometric hierarchy of nested finite element meshes, see Figure 2. The method relies on taking many computationally cheap and low resolution samples on coarser meshes and few computationally expensive but high resolution samples on finer meshes.

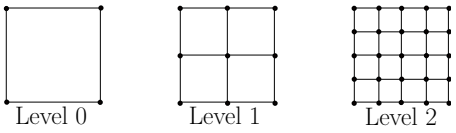


*Figure 2: Illustrative example of a hierarchy used in the MLMC and MLQMC method.*

The standard MC estimator $Q_{\mathrm{L}}^{\mathrm{MC}}$ for the expected value $\mathbb{E}[P_L]$ of a particular quantity of interest $P$ using $N_L$ samples is

$$Q_L^{\mathrm{MC}} = \frac{1}{N_L} \sum_{n=1}^{N_L} P_L(\omega^n), \tag{6}$$

where $\omega^n$ represents the n-th random sample. In MLMC, the expected value, $\mathbb{E}[P_L]$, can be rewritten as

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^{L} \mathbb{E}[P_\ell - P_{\ell-1}], \tag{7}$$

which consequently allows us to write the MLMC estimator as

$$\begin{aligned} Q_L^{\mathrm{MLMC}} &= \frac{1}{N_0} \sum_{n=1}^{N_0} P_0(\omega^n) + \\ &\sum_{\ell=1}^{L} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(P_\ell(\omega^n) - P_{\ell-1}(\omega^n)\right) \right\}. \end{aligned} \tag{8}$$

Each term on the right hand side of Eq. (8) is estimated separately by a standard MC estimator with $N_\ell$ samples. Compactly, the MLMC estimator can be written as a sum of $L+1$ estimators for the expected value of the difference on each level, i.e,

$$\begin{aligned} Q_L^{\mathrm{MLMC}} &= \sum_{\ell=0}^{L} Y_\ell, \\ \text{where} \quad Y_\ell &= \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(P_\ell(\omega^n) - P_{\ell-1}(\omega^n)\right), \end{aligned} \tag{9}$$

with the condition that $P_{-1} := 0$.

The mean square error (MSE) of the MLMC estimator is

$$\mathrm{MSE}(Q_L^{\mathrm{MLMC}}) := \mathbb{V}\left[Q_L^{\mathrm{MLMC}}\right] + \left(\mathbb{E}\left[Q_L^{\mathrm{MLMC}}\right] - \mathbb{E}\left[P\right]\right)^2. \tag{10}$$

It encompasses a part belonging to the variance of the estimator $\mathbb{V}\left[Q_L^{\mathrm{MLMC}}\right]$ and a part belonging to the squared bias $\left(\mathbb{E}\left[Q_L^{\mathrm{MLMC}}\right] - \mathbb{E}\left[P\right]\right)^2$. The variance of the estimator, computed as

$$\mathbb{V}[Q_L^{\mathrm{MLMC}}] = \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell}, \tag{11}$$

with $V_\ell$ denoting the variance of the difference $P_\ell - P_{\ell-1}$, will dictate the number of samples $N_\ell$ needed on each level $\ell$ according to

$$N_\ell = \frac{2}{\varepsilon^2}\sqrt{\frac{V_\ell}{C_\ell}}\sum_{\ell=0}^{L}\sqrt{V_\ell C_\ell}, \tag{12}$$

where $\varepsilon$ is the desired tolerance and $C_\ell$ denotes the cost needed to resolve one sample on level $\ell$. This number of samples is the solution of an optimization problem, see Giles (2008). MLMC is level-adaptive, in the sense that the decision of adding additional levels lies with the condition on the squared bias, written as $(\mathbb{E}[P_\ell - P])^2$. The bias can be rewritten as

$$|\mathbb{E}[P_L - P]| = \left|\sum_{\ell=L+1}^{\infty}\mathbb{E}[P_\ell - P_{\ell-1}]\right| \approx \frac{|\mathbb{E}[P_L - P_{L-1}]|}{2^\alpha - 1}, \tag{13}$$

where $\alpha$ is a simulation parameter that will be defined later on. In order for the MSE to be below a given tolerance $\varepsilon^2$, the squared bias and the variance of the estimator must both be below $\frac{\varepsilon^2}{2}$. Convergence is reached when $|\mathbb{E}[P_L - P_{L-1}]|/(2^\alpha - 1) \leq \varepsilon/\sqrt{2}$ and $\sum_{\ell=0}^{L}\frac{V_\ell}{N_\ell} \leq \frac{\varepsilon^2}{2}$.

In the context of this paper we follow the approach used in the paper by Blondeel et al. (2018). We fix the maximal level for MLMC and MLQMC. This level is defined as L = 3. The number of elements per level is proportional to $40 \cdot 2^{2\cdot\ell}$, resulting in 40 elements on level 0 and 2560 on level 3.

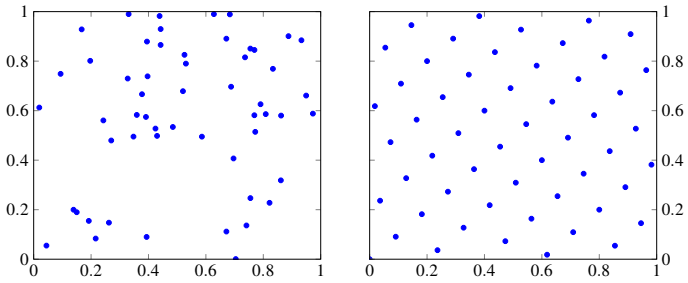

*Figure 3: Example of points sampled for MLMC (left) and MLQMC (right)*

*Multilevel Quasi-Monte Carlo*

One of the major differences with MLMC is that for Multilevel Quasi-Monte Carlo (MLQMC), the individual sample points are not chosen at random but according to some deterministic rule, see Figure 3. For our application in this paper, we use rank-1 lattice rules as in Giles and Waterhouse (2009). These points have the following representation,

$$\mathbf{x}_n = \mathrm{frac}\left(\frac{n}{N}\mathbf{z}\right), \tag{14}$$

where $\mathrm{frac}(x) = x - \lfloor x \rfloor, x > 0$. Vector $\mathbf{z}$ is a $s$-dimensional vector of positive integers, and $N$ is the number of points.

Due to the deterministic nature of the MLQMC points, a shift has to be introduced in order to obtain unbiased estimates of the quantities of interest, as discussed in section 2.9 of Dick et al. (2013). Eq. (14) is rewritten as

$$\mathbf{x}_n = \mathrm{frac}\left(\frac{n}{N}\mathbf{z} + \Delta\right), \tag{15}$$

where $\Delta$ is a shift or offset, $\Delta \in [0,1]^s$. In practice, a number of different random shifts must be chosen, $\Delta_1, \Delta_2, \ldots, \Delta_R$, in order to allow for the computation of the variance, and hence the accuracy of the approximation. The MLQMC estimator is then written as

$$Q_L^{\mathrm{MLQMC}} = \frac{1}{R_0}\sum_{i=1}^{R_0}\frac{1}{N_0}\sum_{n=1}^{N_0}P_0(\mathbf{x}_{i,n}) + \sum_{\ell=1}^{L}\frac{1}{R_\ell}\sum_{i=1}^{R_\ell}\left\{\frac{1}{N_\ell}\sum_{n=1}^{N_\ell}\left(P_\ell(\mathbf{x}_{i,n}) - P_{\ell-1}(\mathbf{x}_{i,n})\right)\right\}. \tag{16}$$

We choose the number of shifts to be constant on each level, i.e., $R_\ell = R$, $\ell = 0, 1, \ldots, L$. A value

$R = 10$ will be chosen in our numerical experiments. Contrary as to MLMC, the number of samples for MLQMC is not the result of an optimization problem, as in Eq. (12). For MLQMC a doubling algorithm is used, see Giles and Waterhouse (2009). Starting with an initial number of samples, this doubling algorithm multiplies the number of samples with a constant factor until the variance of the estimator is smaller than $\frac{\varepsilon^2}{2}$. In our implementation this multiplication constant is chosen to be 1.1.

*Cost*

Having introduced both methods, we now present a complexity theorem for MLQMC, which, with minor changes is also applicable for MLMC. This is done by setting the variable $\delta$ to 1. More details can be found in Teckentrup (2013).

**Theorem 1.** *Given the positive constants* $\alpha, \beta, \gamma, c_1, c_2, c_3$ *such that* $\alpha \geq \frac{1}{2}\min(\beta, \delta^{-1}\gamma)$ *with* $\delta \in (1/2, 1]$ *and assume that the following conditions hold:*

1. $|\mathbb{E}[P_\ell - P]| \leq c_1 2^{-\alpha \ell}$,
2. $\mathbb{V}[Y_\ell] \leq c_2 2^{-\beta \ell} N_\ell^{-1/\delta}$ *and*
3. $C_\ell \leq c_3 2^{\gamma \ell}$.

*Then, there exists a positive constant* $c_4$ *such that for any* $\varepsilon < \exp(-1)$ *there exists an* $L$ *and a sequence* $\{N_\ell\}_{\ell=0}^{L}$ *for which the multilevel estimator,* $Q_L^{\text{MLQMC}}$ *has an* $\text{MSE} \leq \varepsilon^2$, *and*

$$\text{cost}(Q^{\text{MLQMC}}) \leq \begin{cases} c_4 \varepsilon^{-2\delta} & \text{if} \quad \delta\beta > \gamma, \\ c_4 \varepsilon^{-2\delta} (\log \varepsilon)^{1+\delta} & \text{if} \quad \delta\beta = \gamma, \\ c_4 \varepsilon^{-2\delta - (\gamma - \delta\beta)/\alpha} & \text{if} \quad \delta\beta < \gamma. \end{cases} \quad (17)$$

Following this theorem, the optimal cost of the MLMC estimator, is proportional to $\varepsilon^{-2}$ when the variance over the levels decreases faster than the cost per level increases, i.e., $\beta > \gamma$. Similarly, for the MLQMC estimator, the optimal cost is proportional to $\varepsilon^{-1}$. Note that this is only true in the limit, i.e., $\delta \to 1/2$.

RESULTS

In this section we will first estimate the parameters, $\alpha$, $\beta$ and $\gamma$ from Theorem 1. After this we will compare simulation run times for successive refinements on the tolerance. Thirdly, we will present the number of samples per level for both methods. Lastly, we will investigate the effect of the starting mesh on the simulation time. All the results have been computed on a workstation with 14 physical (28 logical cores) Intel Xeon E5-2697 V3, each clocked at 2.6 GHz, and a total of 128 GB RAM. The samples were computed in parallel. With the above mentioned configuration, 28 samples are computed in parallel.

*Variances and Expected Values*

In this part we estimate and present the values for the rates $\alpha$, $\beta$ and $\gamma$ in Theorem 1. These rates have been estimated during the run of the algorithm for different tolerances on the RMSE. Here, we opt to list only the rates for the finest tolerance on the RMSE, which equals $\varepsilon = 2.5E-6$, see Table 1.

Using the results in this table, it is possible to estimate the asymptotic cost of the estimators using the different regimes from Eq. (17). Since $\beta > \gamma$, we expect an optimal cost proportional to $\varepsilon^{-2}$ for MLMC and at most $\varepsilon^{-1}$ for MLQMC. This is indeed what we observe in Figure 5.

*Table 1: Parameters for the MLMC and MLQMC algorithm.*

| RMSE [/] | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|
| 2.5E-6 | 1.04 | 1.95 | 0.89 |

Figure 4 shows the behavior of the expected value and the variance of the quantity of interest $P_\ell$ and of the difference $P_\ell - P_{\ell-1}$ for the MLQMC simulation when the tolerance equals $2.5E-6$. Results for MLMC are identical. The rate $\alpha$ can be read of as the slope of the dashed lines of the left figure, while the rate $\beta$ can be read of as the slope of the dashed lines of the right figure. The rate $\gamma$ is defined by the efficiency of the solver. With our MATLAB FEM code, we obtain $\gamma = 0.89$.

Recalling Eq. (7), it becomes apparent that these *differences*, written as $\mathbb{E}[P_\ell - P_{\ell-1}]$, tend to zero for increasing level $\ell$. This term thus essentially acts as

a correction term for the expected value on level zero, $\mathbb{E}[P_0]$. Because $\mathbb{E}[P_\ell]$ is an approximation of the real quantity $\mathbb{E}[P]$ and because for increasing level $\mathbb{E}[P_\ell] \to \mathbb{E}[P]$, then $\mathbb{V}[P_\ell - P_{\ell-1}]$ tends to zero. The implication of this is that the number of samples $N_\ell$ will be a decreasing function of $\ell$. Most samples will thus be taken on the coarse mesh, where samples are cheap, and a decreasing number of more expensive samples will be taken on finer meshes.

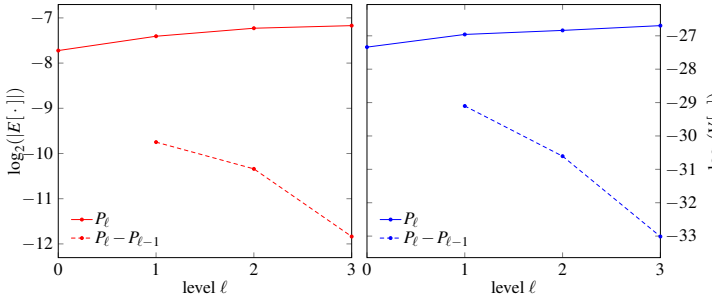


*Figure 4: Expected value of $P_\ell$ and the difference $P_\ell - P_{\ell-1}$ for all levels (left), and variance of $P_\ell$ and the difference $P_\ell - P_{\ell-1}$ for all levels (right) for MLQMC for a tolerance of 2.5E-6.*

*Runtime Comparison*

Figure 5 compares the actual simulation runtime needed to reach a given tolerance for MLMC and MLQMC. As can be seen, the cost order for MLMC is roughly proportional to $\varepsilon^{-2}$, while for MLQMC it tends to $\varepsilon^{-1}$. These are indeed the cost orders we expected for the rates we obtained.

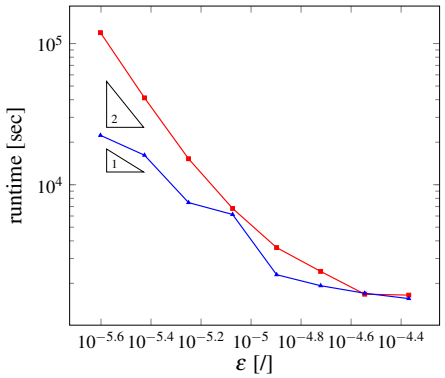


*Figure 5: Total simulation run time in function of RMSE for MLMC and for MLQMC.*

Numerical values corresponding to the runtimes in Figure 5 are shown in Table 2.

As can be observed from the values in Table 2 and Figure 5, the simulation time for MLMC and MLQMC is more or less equal up until a tolerance of 1.9E-5. At that time the speedup in favor of MLQMC become apparent. For finer tolerances, the speedup will be more apparent. This effect is linked to the sample size needed. For MLQMC, there exists a non-negligible startup time. The time needed for one MLQMC sample is tenfold that of one MLMC sample. This is due to the fact that ten shifts are taken for one MLQMC sample. Once we reach these fine tolerances, we observe a speedup of a factor five or more.

*Table 2: Actual simulation time in seconds for MLMC and MLQMC.*

| RMSE [/] | Time [sec] MLMC | MLQMC |
|---|---|---|
| 4.2E-5 | 1.65E+3 | 1.56E+3 |
| 2.8E-5 | 1.67E+3 | 1.70E+3 |
| 1.9E-5 | 2.43E+3 | 1.92E+3 |
| 1.2E-5 | 3.58E+3 | 2.30E+3 |
| 8.4E-6 | 6.79E+3 | 6.15E+3 |
| 5.6E-6 | 1.52E+4 | 7.47E+3 |
| 3.7E-6 | 4.12E+4 | 1.61E+4 |
| 2.5E-6 | 1.19E+5 | 2.23E+4 |

*Sample sizes*

The sample sizes for different tolerances on the RMSE are shown in Figure 6 for MLMC and in Figure 7 for MLQMC. As can be observed, the number of samples per level is decreasing with increasing level. This matches our earlier observation. Table 3 shows the values from Figure 6 and Figure 7. The number of samples listed for MLMC are the actual samples taken, while for MLQMC these numbers include the multiplication with the number of shifts.

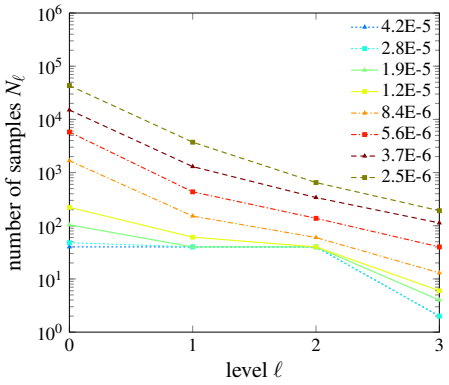


*Figure 6: Total number of samples $N_\ell$ taken on each level for MLMC.*

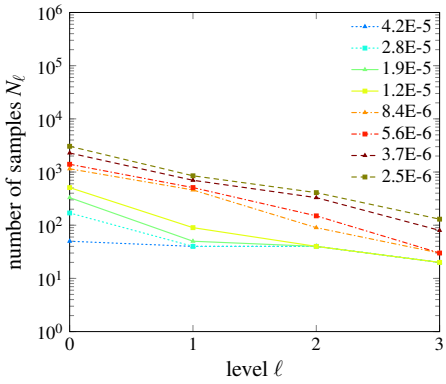


*Figure 7: Total number of samples $N_\ell$ taken on each level for MLQMC.*

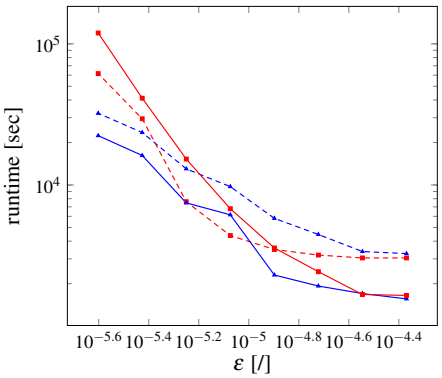


*Figure 8: Total simulation run time in function of RMSE for MLMC and for MLQMC for a starting level $\ell = 0$ and for MLMC and for MLQMC for a starting level $\ell = 1$.*

*Table 3: Number of samples for MLMC and MLQMC.*

| RMSE [/] | MLMC level | | | | MLQMC level | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 4.2E-5 | 40 | 40 | 40 | 2 | 50 | 40 | 40 | 20 |
| 2.8E-5 | 48 | 40 | 40 | 2 | 170 | 40 | 40 | 20 |
| 1.9E-5 | 104 | 40 | 40 | 4 | 330 | 50 | 40 | 20 |
| 1.2E-5 | 222 | 61 | 40 | 6 | 510 | 90 | 40 | 20 |
| 8.4E-6 | 1674 | 152 | 60 | 13 | 1150 | 460 | 90 | 30 |
| 5.6E-6 | 5790 | 434 | 138 | 40 | 1400 | 410 | 150 | 30 |
| 3.7E-6 | 15025 | 1295 | 339 | 113 | 2280 | 700 | 330 | 80 |
| 2.5E-6 | 43312 | 3726 | 649 | 192 | 3050 | 850 | 410 | 130 |

*Table 4: Number of samples for MLMC and MLQMC for a finer starting mesh than the results reported in Table 3.*

| RMSE [/] | MLMC level | | | | MLQMC level | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 4.2E-5 | - | 40 | 40 | 40 | - | 70 | 40 | 40 |
| 2.8E-5 | - | 40 | 40 | 40 | - | 100 | 40 | 40 |
| 1.9E-5 | - | 81 | 40 | 40 | - | 510 | 50 | 40 |
| 1.2E-5 | - | 189 | 40 | 40 | - | 570 | 110 | 40 |
| 8.4E-6 | - | 411 | 58 | 40 | - | 1150 | 270 | 50 |
| 5.6E-6 | - | 1432 | 136 | 40 | - | 1150 | 300 | 30 |
| 3.7E-6 | - | 6383 | 276 | 103 | - | 2510 | 570 | 130 |
| 2.5E-6 | - | 11312 | 647 | 220 | - | 4490 | 570 | 170 |

### *Influence of the starting level*

In this section we investigate the influence of the starting level on the total runtime of both methods. We chose as a starting level $\ell = 1$ instead of $\ell = 0$ as was the case in the previous sections. This implies that a finer starting mesh is used, see Figure 2.

In Figure 8, the total simulation time with a starting level $\ell = 0$, is compared to that of a starting level $\ell = 1$. As can be observed, the choice of the starting mesh influences whether MLQMC is faster than MLMC for a given tolerance. This effect can be understood by investigating the number of samples per level. These values are shown in Table 4. The number of samples per level for MLQMC is larger than for MLMC for coarser tolerances. For finer tolerance the number of samples of MLMC surpasses the number of samples for MLQMC. It is important to note that if a finer starting mesh is used, $\ell = 1$, the MLQMC speed gain will only become apparent for finer tolerances than in the case of $\ell = 0$.

Comparing the values from Table 3 with the ones from Table 4, it is clear that the number of samples for MLMC on level $\ell = 0$ is a factor four to five larger than for MLMC on level $\ell = 1$. While for MLQMC the number is roughly the same.

## CONCLUSION

In this work, we have successfully demonstrated that the Multilevel Monte Carlo and the Multilevel Quasi-Monte Carlo implementation can be coupled with a non-linear Finite Element solver for a structural engineering problem subject to uncertainty of the material properties. This proves the versatility of these two methods. To the best of our knowledge, this is the first time the MLQMC method has been successfully applied for problems in structural engineering. We also investigated the runtime in function of the desired tolerance on the RMSE. We found that for fine tolerances, the MLQMC method offers a significant speedup. For coarser tolerances,

the difference between the two approaches is less outspoken. This is due to the start-up cost (the so called "warm-up" samples), which is similar for both methods. We also notice that the choice of the starting mesh has an important effect on whether MLQMC outperforms MLMC or not for a given tolerance. Further paths of research will consist of comparing MLMC and MLQMC with Multi-index Monte Carlo and Multi-index Quasi-Monte Carlo, see Robbe et al. (2017).

ACKNOWLEDGEMENTS

This research was funded by project IWT-140068 of the IWT-SBO EUFORIA consortium. The authors gratefully acknowledge the support from the research council of KU Leuven through the funding of project C16/17/008. The authors also would like to thank the Structural Mechanics Section of the KU Leuven and Jef Wambacq, in particular, for many helpful discussions.

## REFERENCES

Blondeel, P., Robbe, P., Van hoorickx, C., Lombaert, G., and Vandewalle, S. (2018). "Multilevel Monte Carlo for uncertainty quantification in structural engineering." *arXiv:1808.10680v1*.

Blondeel, P., Robbe, P., Van hoorickx, C., Lombaert, G., and Vandewalle, S. (2018). "The Multilevel Monte Carlo method applied to structural engineering problems with uncertainty in the young's modulus." *Proceedings of the 28th edition of the Biennial ISMA conference on Noise and Vibration Engineering, ISMA 2018*, Desmet, W and Pluymers, B and Moens, D and Rottiers, W (eds), 4899–4913.

Caflisch, R. E. (1998). "Monte Carlo and quasi-Monte Carlo methods." *Acta Numerica*, 7, 1–49.

Cliffe, K. A., Giles, M. B., Scheichl, R., and Teckentrup, A. L. (2011). "Multilevel Monte Carlo methods and applications to elliptic pdes with random coefficients." *Comput. Vis. Sci.*, 14(1), 3–15.

de Borst, R., Crisfield, M. A., and Remmers, J. J. C. (2012). *Non Linear Finite Element Analysis of Solids and Structures*. Wiley, U.K.

Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). "High-dimensional integration: The quasi-monte carlo way." *Acta Numerica*, 22, 133–288.

Ghanem, R. G. and Spanos, P. D. (2003). *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, New York.

Giles, M. B. (2008). "Multilevel Monte Carlo path simulation." *Operations Research*, 56(3), 607–617.

Giles, M. B. and Waterhouse, B. J. (2009). "Multilevel quasi-Monte Carlo path simulation." *Radon Series on Computational and Applied Mathematics*, 8, 1–18.

Grigoriu, M. (1998). "Simulation of stationary non-gaussian translation processes." *J. Engrg. Mech. (ASCE)*, 124(2), 121–126.

Heinrich, S. (2001). "Multilevel Monte Carlo methods." *Large-Scale Scientific Computing*, S. Margenov, J. Waśniewski, and P. Yalamov, eds., Berlin, Heidelberg, Springer Berlin Heidelberg, 58–67.

Hess, P. E., Bruchman, D., Assakkaf, I. A., and Ayyub, B. M. (2002). "Uncertainties in material and geometric strength and load variables." *Naval Engineers Journal*, 114(2), 139–166.

Loève, M. (1977). *Probability theory*. Springer, New York.

Loh, W.-L. (1996). "On latin hypercube sampling." *Ann. Statist.*, 24(5), 2058–2080.

Niederreiter, H. (2004). *Monte Carlo and quasi-Monte Carlo methods*. Springer, Berlin.

Perez-Foguet, A., Rodriguez-Ferran, A., and Huerta, A. (2001). "Consistent tangent matrices for substepping schemes." *Comput.Methods in Appl.Mech.Eng*, 190(35), 4627–4647.

Robbe, P., Nuyens, D., and Vandewalle, S. (2017). "A multi-index quasi–Monte Carlo algorithm for lognormal diffusion problems." *SIAM J. Sci. Comput.*, 39(5), S851–S872.

Robbe, P., Nuyens, D., and Vandewalle, S. (2018). "A dimension-adaptive multi-index Monte Carlo method applied to a model of a heat exchanger." *Monte Carlo and Quasi-Monte Carlo Methods*, A. B. Owen and P. W. Glynn, eds., Cham, Springer International Publishing, 429–445.

Teckentrup, A. L. (2013). "Multilevel Monte Carlo methods and uncertainty quantification." Ph.D. thesis, University of Bath. `http://www.maths.bath.ac.uk/~masrs/Teckentrup_PhD.pdf`.