Comparing speed and cost of MLMC with MLQMC in the framework of a practical structural engineering case

Philippe Blondeel
Graduate Student, Dept. of Computer Science, NUMA Section, KU Leuven, Leuven, Belgium

Pieterjan Robbe
Graduate Student, Dept. of Computer Science, NUMA Section, KU Leuven, Leuven, Belgium

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

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

Stefan Vandewalle
Professor, Dept. of Computer Science, NUMA Section, KU Leuven, Leuven, Belgium

ABSTRACT: The abstract is written last, when the paper is complete. It contains a summary of the paper, usually starting with a few words about the objectives. Thereafter follow usually statements about the importance of the work and the methodology that is employed. The key results are then summarized followed by an overview of the conclusions that were drawn from the study.

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 to quantify it, two main types of methods exist: Non sampling methods and sampling methods. The category of 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 due to the transformation of the uncertain coefficient partial differential equation into a large system of coupled of deterministic PDEs by means of a Galerkin projection technique. Sampling methods on the other hand do generally not exhibit this intrusive character. A well known example of such a sampling method is the Monte Carlo (MC) method. For each at random selected sample, the problem is deterministically solved. A major drawback attributed to the Monte Carlo method, is the slow convergence in function of the number of samples taken. This can be however overcome by variance reduction techniques, such as Multilevel Monte Carlo or improved sampling methods, such as Latin Hypercube sampling, see Loh (1996) or Quasi-Monte Carlo, see Caflisch (1998); Niederreiter (2004). Multilevel Monte Carlo was first proposed in Heinrich (2001) and successfully applied in the finance context for option pricing in Giles (2008).

It is in this context that we present our work.

Building on the work done in Blondeel et al. (2018); Blondeel et al. (2018), where we compared the speed and cost of Monte Carlo with Multilevel Monte Carlo for a similar structural engineering application and found a speed of a factor ten in favor of Multilevel Monte Carlo, 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). Our aim will be to offer a comparison in terms of runtime and cost for a practical structural engineering problem and to demonstrate the versatility of these two methods. The engineering problem we consider, is defined as a steel beam, clamped at both sides and loaded in the middle where the Young's modulus is considered uncertain.

Case Presentation

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

Beam Characteristics

In this paper we consider a steel beam clamped on both sides subjected to a static force located in its middle, Fig. 1. The beam has the following dimensions and material parameters: 2.5 m long by 0.25 m high by 10^{-3} m tick, 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.

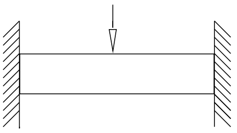

Figure 1: Cantilever beam clamped at both sides loaded in the middle.

Uncertainty Modeling

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

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

$$C(\mathbf{x},\mathbf{y}) := \sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_1}{\lambda}\right). \quad (1)$$

The 1-norm is selected together with a correlation length $\lambda = 0.3$ and a standard deviation $\sigma = 1.0$. The KL expansion can then be formulated as follows:

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

The obtained Gaussian random field, $Z(\mathbf{x},\omega)$, is dependent upon respectively the eigenvalues and eigenfunctions, θ_n and $b_n(\mathbf{x})$ of the covariance kernel (1). $\xi_n(\omega)$ denotes the i.i.d standard normal random variables. For a one dimensional domain [0,1], the eigenvalue and eigenfunctions can be computed analytically according to

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

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

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

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

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

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 $\beta = 0.215 \times 10^9$, see Hess et al. (2002).

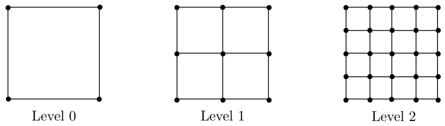

Figure 2: Illustrative example of a hierarchy used in the MLMC and MLQMC method.

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 underlying equations and solution methods is presented hereunder.

Due to the nonlinear stress-strain relationship that exists in the elasto-plastic domain, an incremental load approach is used. 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 by means of a return mapping algorithm. 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).

Elaborate a bit?

METHODOLOGY

This section gives a short overview about the theory pertaining to the Multilevel Monte Carlo method and the Multilevel Quasi-Monte Carlo method. A more detailed analysis can be found in Giles (2008, 2015)

Multilevel Monte Carlo

The Multilevel Monte Carlo (MLMC) method is a extension of the standard Monte Carlo (MC) Method, which employs a hierarchy of leveld in order to achieve a variance reduction and by doing so speed up the computation. Here these levels are defined as a hierarchy of nested finite element meshes, Fig. 2. The method thus relies on taking many computationally cheap and low resolution samples on coarser meshes and few computationally expensive but high resolution samples on finer meshes.

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

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

In the case of MLMC, the expected value, $\mathbb{E}[P_L]$, can be rewritten as follows

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

which consequently allows us to write the MLMC as follows

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

Each terms on the right hand side of (8) is estimated separately by a standard MC estimator with N_ℓ 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) can be written as follows,

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

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

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

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

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

MLMC is level adaptive, the decision of adding extra levels lies with the condition on the squared bias, written as $(\mathbb{E}[P_\ell - P])^2$. The bias can be rewritten as follows.

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

In order for the MSE to be below a given tolerance ε^2, the squared bias and the variance of the estimator must both be below $\frac{\varepsilon^2}{2}$. Convergence is then checked according to $|\mathbb{E}[P_L - P_{L-1}]|/(2^\alpha - 1) \leq \varepsilon/\sqrt{2}$ and $\sum_{\ell=0}^{L}\frac{V_\ell}{N_\ell} \leq \frac{\varepsilon^2}{2}$.

In the context of this paper and following the approach used in Blondeel et al. (2018), we opt to fix the maximal level for MLMC and MLQMC. This level is defined as level three. The number of elements per level is proportional to $40 \cdot 2^{2\cdot\ell}$.

Elaborate over interpretation

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. Examples of such rules are Sobol'sequences and digital nets. For our application in this paper, we use rank-1 lattice rules as in Giles and Waterhouse (2009). These points have the following representation,

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

where frac (x) stands for the fractional part function which is defined as frac $(x) = x - \lfloor x \rfloor, x > 0$, z is a d-dimensional vector consisting of integers, n being the index and N the number of points.

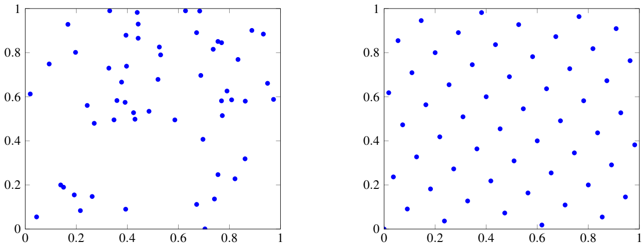

Figure 3: Example of points sampled for MLMC (left) and MLQMC (right)

Due to the deterministic nature of the MLQMC points, a shift has to be introduced in order to be able to obtain unbiased estimates of the quantities of interest. (14) is rewritten as

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

where Δ stands for shift or offset. In practice a number of different random shifts must be chosen $\Delta_1, \Delta_2, \ldots, \Delta_i$. The MLQMC estimator is then written as

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

According to good practice, the number of shifts, R_ℓ in this paper is taken to be 10 on each level. Contrary as for MLMC, the number of samples for MLQMC is not the result of an optimization problem (12). For MLQMC a doubling algorithm is used as in Kuo et al. (2017). Starting with a user defined initial number of samples and number of shifts, this doubling algorithm checks the condition on the variance of the system according to $\sum_{l=0}^{L} V_\ell > \varepsilon^2/2$. If the variance is above $\varepsilon^2/2$, the number of samples to be taken equals the number already taken times a multiplication constant. In our case 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 by setting the variable δ 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 $\alpha \geq \frac{1}{2}\min\left(\beta, \delta^{-1}\gamma\right)$ 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^{MLQMC} has an MSE $\leq \varepsilon^2$, and

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

Following this theorem, the cost of the MLMC estimator, in its most optimal case equals ε^{-2}. The most optimal cast being defined as when the variance over the levels decreases faster than the cost per level increases, $\beta > \gamma$. Similarly, for the MLQMC estimator, the cost, in the same condition as for MLMC estimator, is proportional to ε^{-1}. Note that this is only true in the limit, so the best we can expected is $\delta \approx 1/2$.

Results

This section will present a comparison in actual simulation runtime for the considered beam case computed with the MLMC and the MLQMC method. Following this comparison, the variances and the expected values on each level will be shown for each method. 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.

Variances and Expected Values

This part we validate the results obtained with MLMC and MLQMC by investigating the expected value, the variance, the difference of the expected values and the difference of the variances on each level. Fig. 4 and Fig. 5, respectively shows this for the MLMC and MLQMC simulation with a tolerance on the RMSE of 2.5E-6.

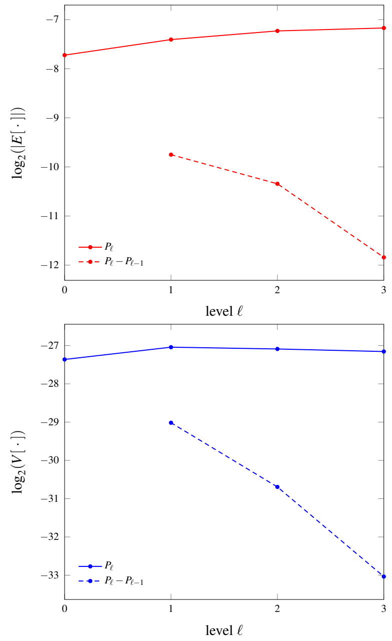

Figure 4: Expected value and variance and their differences over the levels for MLMC for a tolerance of 2.5E-6.

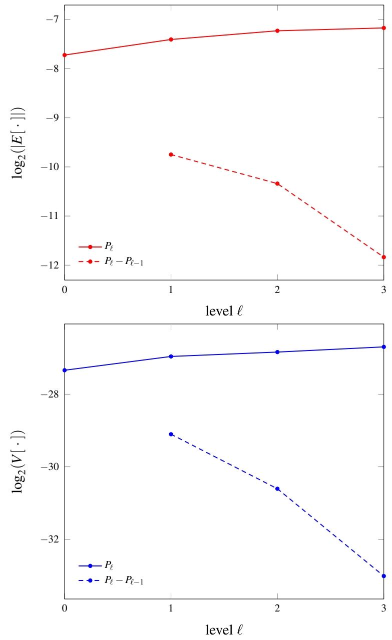

Figure 5: Expected value and variance and their differences over the levels for MLQMC for a tolerance of 2.5E-6.

Intoduce the table

Table 1: Parameters for the MLMC and MLQMC algorithm.

RMSE [/]	MLMC			MLQMC		
	α	β	γ	α	β	γ
2.5E-6	tbd	tbd	tbd	1.04	1.95	0.89

Discus why this decreases, the implications, ect..

Cost Comparison

Fig. 6 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 ε^{-2}, while for MLQMC it is proportional to ε^{-1}.

Elaborate over cost

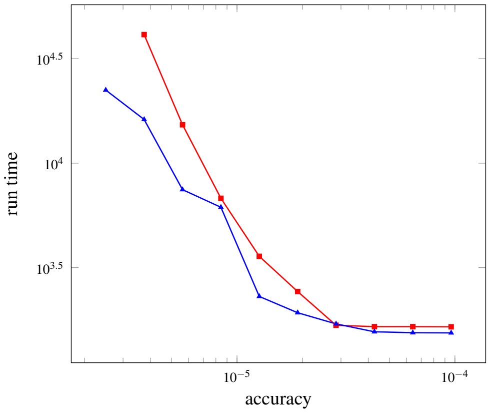

Figure 6: Total simulation run time in function of RMSE for MLMC —🟥 *and for MLQMC* —▲.

Values corresponding to the times shown in Fig. 6 are taken up in Tab. 2.

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

RMSE [/]	Time [sec] MLMC	MLQMC
9.6E-5	1.64E+3	1.54E+3
6.4E-5	1.65E+3	1.54E+3
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	TBD	2.23E+4

As can be observed from the value in Tab. 2 and Fig. 6, the simulation time for MLMC and MLQMC is more or less equal up until the tolerance of 1.9E-5. At that time the speedup in favor of MLQMC become apparent. Only for fine enough tolerances will there thus be a speedup. This effect is linked to the sample size needed, indeed for MLQMC, there exists a non-negligible startup cost. This is because the cost for 1 MLQMC sample is tenfold that of 1 MLMC sample. This is due to the fact that 10 shifts are taken for 1 MLQMC sample. The speedup for MLQMC will thus only become apparent for fine tolerances.

Sample sizes

Show decreasing sample sizes for fines level This section presents the sample sizes for the different tolerances for MLMC, Fig. 7 and for MLQMC, Fig. 8.

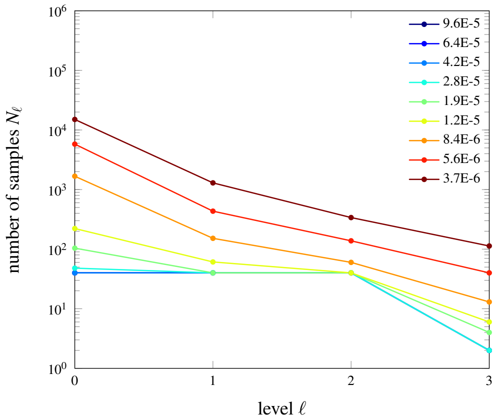

Figure 7: Total number of samples N_ℓ taken on each level for MLMC.

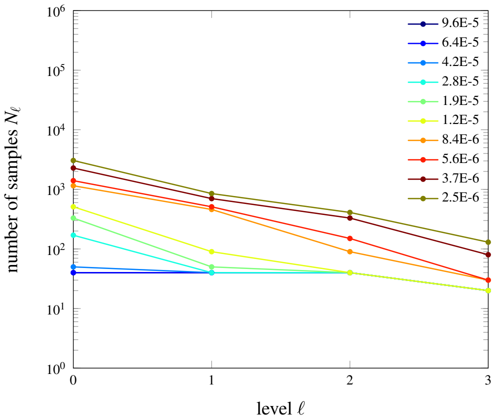

Figure 8: Total number of samples N_ℓ taken on each level for MLQMC.

Tab. 3 tables the values corresponding what is presented in Fig. 7 and Fig. 8.

Table 3: Number of samples for MLMC and MLQMC.

	MLMC				MLQMC			
	level				level			
RMSE [/]	0	1	2	3	0	1	2	3
9.6E-5	40	40	40	2	40	40	40	20
6.4E-5	40	40	40	2	40	40	40	20
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	TBD	TBD	TBD	TBD	3050	850	410	130

Conclusion

REFERENCES

Blondeel, P., Robbe, P., van hoorickx, C., Lombaert, G., and Vandewalle, S. (2018). "Multilevel Monte Carlo for uncertainty quantification in structural engineering." *ArXiv e-prints*.

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." Desmet, W, Department of mechanical engineering - KULeuven, 4899–4913.

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

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

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. (2015). "Multilevel Monte Carlo methods." *Acta Numerica*, 24, 259–328.

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

Kuo, F. Y., Scheichl, R., Schwab, C., Sloan, I. H., and Ullmann, E. (2017). "Multilevel quasi-monte carlo methods for lognormal diffusion problems." *Math. Comp.*, 39(86), 2827–2860.

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.

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`.