14th World Congress on Computational Mechanics (WCCM)
ECCOMAS Congress 2020)
Virtual Congress: 11-–15 January 2021
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

BENCHMARKING THE P-MLQMC METHOD ON A GEOTECHNICAL ENGINEERING PROBLEM

Philippe Blondeel[1], Pieterjan Robbe[1], Stijn François[2], Geert Lombaert[2] and Stefan Vandewalle[1]

[1] KU Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Leuven, Belgium
{philippe.blondeel,pieterjan.robbe,stefan.vandewalle}@kuleuven.be

[2] KU Leuven, Department of Civil Engineering
Kasteelpark Arenberg 40, 3001 Leuven, Belgium
{stijn.francois,geert.lombaert}@kuleuven.be

Key words: Multilevel Quasi-Monte Carlo, p-refinement, Higher Order Finite Elements

Abstract. Problems in civil engineering are often characterized by significant uncertainty in their material parameters. Sampling methods are a straightforward manner to account for this uncertainty, which is typically modeled as a random field. A novel method developed by the authors called p-refined Multilevel Quasi-Monte Carlo (p-MLQMC), achieves a significant computational cost reduction with respect to classic Multilevel Monte Carlo (h-MLMC). p-MLQMC uses a hierarchy of p-refined Finite Element meshes, combined with a deterministic Quasi-Monte Carlo (QMC) sampling rule. A non-negligible part of modeling the stochastics in non-deterministic engineering problems consists in adequately incorporating the uncertainty in the Finite Element model. This is typically done by evaluating the random field at certain carefully chosen evaluation points, and assigning the resulting scalars to the different finite elements. For the h-MLMC method, these evaluation points consist of the centroids of the elements. For the p-MLQMC method, we distinguish two different approaches to select the evaluation points, *the Non-Nested Approach* (NNA) and *the Local Nested Approach* (LNA). We investigate how these approaches affect the variance reduction over the levels and the total computational cost. We benchmark the p-ML(Q)MC-LNA, p-ML(Q)MC-NNA and h-MLMC method on a slope stability problem where the uncertainty is located in the soil's cohesion. We show that the p-MLQMC-LNA method outperforms all other considered methods in terms of computational cost.

1 INTRODUCTION

Problems in the engineering sciences are typically subject to uncertainty. This uncertainty can be present in the material parameters, as for example, the cohesion of the soil in a slope stability problem. Purely deterministic solutions provide only limited insight into the assessement of structural safety or stability, under uncertain material conditions. There is therefore an increasing need for fast stochastic methods which can assess the uncertainty on the output of a model. A popular stochastic sampling method consists of the Multilevel Monte Carlo Method (MLMC). First developped by Giles, see [1, 2], the MLMC methods relies on a hierarchy of refined meshes in order to reduce the total computational cost by means of variance reduction. Most of the samples are taken on low resolution and computationally cheap meshes, while a decreasing

number of samples are taken on high resolution and computationally expensive meshes. The mesh hierarchy is typically constructed by selecting a coarse Finite Element mesh approximation of the considered problem, and recursively applying the h-refinement scheme. The MLMC method already considerably reduces the total computational cost with respect to standard Monte Carlo (MC). In standard MC, all the samples are taken on the highest resolution mesh, i.e., the mesh with the highest computational cost. A speedup up to a factor 10 can be achieved by means of MLMC with respect to standard MC. This was demonstrated on an academic beam bending problem where the beam deflection was considered, see [3]. Improvements such as the Multilevel Quasi-Monte Carlo method have been developped, see [4]. Here the randomly generated Monte Carlo sample points are replaced with deterministically generated Quasi-Monte Carlo (QMC) sample points. In our previous work, see [5], we have demonstrated that the MLQMC method outperforms the MLMC method in terms of computational cost. However, classic ML(Q)MC remains costly due to an almost geometrical increase in the number of degrees of freedom for each refined mesh. Therefore, we have developped a novel multilevel method, called p-refined Multilevel Quasi-Monte Carlo (p-MLQMC), see [6]. This multilevel method combines a mesh hierarchy based on a p-refinement scheme with a QMC sampling rule. This approach yields significant computational cost savings with respect to classic Multilevel (Quasi-)Monte Carlo (h-ML(Q)MC), where the mesh hierarchy is based on an h-refinement approach. However, the p-MLQMC method presents the practitioner with additional difficulties. The main difficulty consists how to adequately incorporate the uncertainty, modeled as a random field, in the Finite Element model. In case of h-ML(Q)MC, this is achieved by applying the midpoint method. Scalars obtained from the evaluation of the random field at the elements centroids are assigned to each element. However for the p-MLQMC method, the midpoint method cannot be used. This is because the refinement strategy in the p-MLQMC method does not increase the number of elements. For the p-MLQMC method, we resort to the integration point method. Here, scalars resulting from the evaluation of the random field at certain spatial locations, are accounted for during numerical integration of the element stiffness matrices. The choice of these evaluation points will influence the variance reduction over the levels, and thus the total computational runtime. We distinguish two different approaches to select these points, *the Non-Nested Approach* (NNA) and *the Local Nested Approach* (LNA).

In this work we will benchmark the p-ML(Q)MC-LNA/NNA methods and the h-MLMC method. We will show how the uncertainty is accounted for in the model, and how the evaluation points of the random field are to be selected. The methods will be compared in terms of total computational cost. The model problem on which we will benchmark the methods, consists of a slope stability problem. The slope stability problem is a geotechnical engineering problem, where the goal is to assess the stability of natural or man-made slopes. This stability can, among other, be assessed by investigating the vertical displacement of the top of the slope. The paper is structured as follows. First we introduce the theoretical background pertaining to multilevel methods. Second, we discuss the underlying Finite Element Solver, and present the model problem. Last, we present the results obtained for p-ML(Q)MC-LNA/NNA and h-MLMC.

2 THEORETICAL BACKGROUND

In this section, we present the theoretical background for the p-ML(Q)MC-LNA/NNA and h-MLMC methods. We start by giving a formulation for the MC estimator, before introducing the MLMC and MLQMC estimators. Hereafter, we discuss the mechanics pertaining to the variance reduction over the levels. In the next part we relate the number of samples to the variance reduction. In the second to last part, we present the mesh hierarchies of the discretized slope stability problem. In the final part, we present how the midpoint and integration point method are to be implemented for h-MLMC and p-ML(Q)MC-LNA/NNA. Here the

focus lies on the choice of the evaluation points of the random field.

2.1 Estimators and Variances

The MC estimator is given by

$$Q_{\mathrm{L}}^{\mathrm{MC}} := \frac{1}{N_{\mathrm{L}}} \sum_{n=1}^{N_{\mathrm{L}}} P_{\mathrm{L}}(\mathbf{u}_{\mathrm{L}}^{(n)}). \tag{1}$$

It expresses the expected value of the quantity of interest P on the finest level $\ell = \mathrm{L}$, as the sample average of the quantity of interest on the finest level P_{L}, where $\mathbf{u}_{\mathrm{L}}^{(n)}$ stands for the Monte Carlo evaluation points, i.e., all the samples are computed on the finest level.

On the other hand, the MLMC estimator is given by

$$Q_{\mathrm{L}}^{\mathrm{MLMC}} := \frac{1}{N_0} \sum_{n=1}^{N_0} P_0(\mathbf{u}_0^{(n)}) + \sum_{\ell=1}^{\mathrm{L}} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(P_\ell(\mathbf{u}_\ell^{(n)}) - P_{\ell-1}(\mathbf{u}_\ell^{(n)}) \right) \right\}. \tag{2}$$

Here the expected value of the quantity of interest on the finest level $\ell = \mathrm{L}$, is expressed as the sample average of the quantity of interest on the coarsest level $\ell = 0$, plus a series of correction terms on levels $\ell = \{1, \ldots, \mathrm{L}\}$. The variance of the MLMC estimator is given by

$$\mathbb{V}\left[Q_{\mathrm{L}}^{\mathrm{MLMC}} \right] = \sum_{\ell=0}^{\mathrm{L}} \mathbb{V}\left[\Delta Q_\ell^{\mathrm{MLMC}} \right] := \sum_{\ell=0}^{\mathrm{L}} \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \frac{\left(\Delta P_\ell^{(n)} - \mathbb{E}\left[\Delta P_\ell\right] \right)^2}{(N_\ell - 1)} = \sum_{\ell=0}^{\mathrm{L}} \frac{\mathbb{V}\left[\Delta P_\ell\right]}{N_\ell}, \tag{3}$$

where $\mathbb{E}\left[\Delta P_\ell\right] := \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \Delta P_\ell^{(n)}$, with $\Delta P_\ell^{(n)} := P_\ell(\mathbf{u}_\ell^{(n)}) - P_{\ell-1}(\mathbf{u}_\ell^{(n)})$, and $P_{-1} := 0$.

The MLQMC estimator, used in the p-MLQMC-LNA/NNA methods, and its variance are given by

$$Q_{\mathrm{L}}^{\mathrm{MLQMC}} := \frac{1}{R_0} \sum_{r=1}^{R_0} \frac{1}{N_0} \sum_{n=1}^{N_0} P_0(\mathbf{u}_0^{(r,n)}) + \sum_{\ell=1}^{\mathrm{L}} \frac{1}{R_\ell} \sum_{r=1}^{R_\ell} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(P_\ell(\mathbf{u}_\ell^{(r,n)}) - P_{\ell-1}(\mathbf{u}_\ell^{(r,n)}) \right) \right\}, \tag{4}$$

and

$$\mathbb{V}\left[Q_{\mathrm{L}}^{\mathrm{MLQMC}} \right] = \sum_{\ell=0}^{\mathrm{L}} \mathbb{V}\left[\Delta Q_\ell^{\mathrm{MLQMC}} \right] := \sum_{\ell=0}^{\mathrm{L}} \frac{1}{N_\ell R_\ell} \sum_{r=1}^{R_\ell} \frac{1}{(R_\ell - 1)} \sum_{n=1}^{N_\ell} \frac{\left(\Delta P_\ell^{(r,n)} - \mathbb{E}\left[\Delta P_\ell\right] \right)^2}{N_\ell} = \sum_{\ell=0}^{\mathrm{L}} \frac{\mathbb{V}\left[\Delta P_\ell\right]}{N_\ell R_\ell}, \tag{5}$$

where $\mathbb{E}\left[\Delta P_\ell\right] := \frac{1}{R_\ell} \sum_{r=1}^{R_\ell} \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \Delta P_\ell^{(r,n)}$, with $\Delta P_\ell^{(r,n)} := P_\ell(\mathbf{u}_\ell^{(r,n)}) - P_{\ell-1}(\mathbf{u}_\ell^{(r,n)})$, and $P_{-1} := 0$. R_ℓ stands for the number of shifts per level, see further on.

The major difference between the MLQMC and MLMC estimator resides in the choice of the sample points. The estimator for MLQMC and its variance, see Eq. (4) and Eq. (5) are based on deterministic sample points (QMC points), $\mathbf{u}_\ell^{(r,n)}$. On the other hand, the estimator for MLMC and its variance, see Eq. (2) and Eq. (3) are based on a set of random sample points (MC points), $\mathbf{u}_\ell^{(n)}$. This is illustrated in Figure 1 on a two-dimensional unit square.

However, the use of these deterministic sample points introduces an additional bias on the stochastic quantities of the computed solution. In order to recover unbiased estimates, 'randomness' needs to be reintroduced. This is achieved by supplementing the formulation of the estimator and its variance with a summation over

a number of random shifts $r = 1, 2, ..., R_\ell$ on each level ℓ, with the random shifts $\Xi_r \in [0,1]^s$. In our implementation $R_\ell = 10$ for $\ell = \{0, \ldots, \mathrm{L}\}$. The QMC sample points are obtained according to

$$\mathbf{u}^{(r,n)} := \Phi^{-1}\left(\mathrm{frac}\left(\phi_2(n)\mathbf{z} + \Xi_r\right)\right), n \in \mathbb{N}, \tag{6}$$

where Φ^{-1} is the inverse of the univariate standard normal cumulative distribution function, $\mathrm{frac}(x) := x - \lfloor x \rfloor, x > 0$, ϕ_2 is the radical inverse function in base 2 and $\mathbf{z}$ is an s-dimensional vector of positive integers. This representation is known as a *shifted rank-1 lattice rule*. The generating vector $\mathbf{z}$ was constructed with the component-by-component (CBC) algorithm with decreasing weights, $\gamma_j = 1/j^2$, see [7].

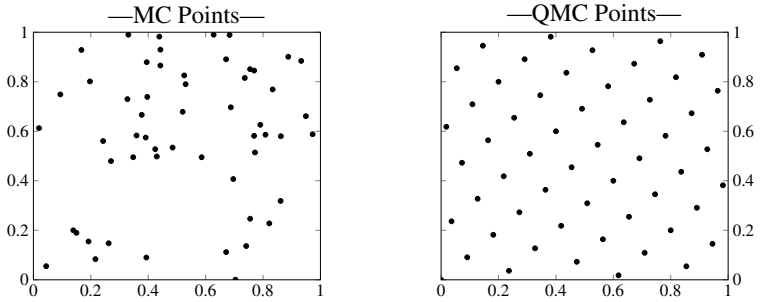

Figure 1: Example of MC and QMC sample points.

2.2 Variance Reduction

Multilevel methods rely on a variance reduction across the levels in order to achieve a computational speedup. This means that the sample variance of the difference for increasing level ℓ continuously decreases, i.e., $\mathbb{V}[\Delta P_1] > \mathbb{V}[\Delta P_2] > \cdots > \mathbb{V}[\Delta P_\mathrm{L}]$. This variance reduction is only obtained when a strong positive correlation is achieved between the results of two successive levels, i.e.,

$$\begin{aligned} \mathbb{V}[\Delta P_\ell] &= \mathbb{V}[P_\ell - P_{\ell-1}] \\ &= \mathbb{V}[P_\ell] + \mathbb{V}[P_{\ell-1}] - 2\mathrm{cov}(P_\ell, P_{\ell-1}), \end{aligned} \tag{7}$$

where $\mathrm{cov}(P_\ell, P_{\ell-1}) = \rho_{\ell,\ell-1}\sqrt{\mathbb{V}[P_\ell]\mathbb{V}[P_{\ell-1}]}$ is the covariance between P_ℓ and $P_{\ell-1}$ with $\rho_{\ell,\ell-1}$ the correlation coefficient. The value of $\mathrm{cov}(P_\ell, P_{\ell-1})$ must be larger than 0 to have a large variance reduction, and hence an efficient multilevel method.

2.3 Number of Samples per Level

The sample variance of the difference $\mathbb{V}[\Delta P_\ell]$ determines the number of samples to be taken on level $\ell = \{1, \ldots, \mathrm{L}\}$. It is only on level $\ell = 0$, that the sample variance $\mathbb{V}[P_\ell]$ is used.

In the MLMC method, the number of samples per level N_ℓ is the result of an optimization problem, see [1]. The cost of the MLMC estimator, given by

$$C\left(Q_\mathrm{L}^\mathrm{MLMC}\right) = \sum_{\ell=0}^{\mathrm{L}} N_\ell C_\ell, \tag{8}$$

with C_ℓ the computational cost to compute a single realization on level ℓ, is minimized, while subjected to the constraint that the variance of the estimator from Eq. (3) is smaller than $\frac{\varepsilon^2}{2}$. Here, ε is the user requested tolerance on the root mean square error (RMSE). Note that $C_0 < C_1 < \cdots < C_\mathrm{L}$. When treating the number of samples as a continuous variable, the number of samples per level is given by,

$$N_\ell = \frac{2}{\varepsilon^2}\sqrt{\frac{\mathbb{V}[\Delta P_\ell]}{C_\ell}}\sum_{\tau=0}^{\mathrm{L}}\sqrt{\mathbb{V}[\Delta P_\tau]C_\tau}. \tag{9}$$

For the MLQMC method, a different approach is followed. This is because an expression similar to the one in Eq. (9), is difficult to obtain. Therefore, we resort to the 'doubling' algorithm. The procedure starts by computing a number of startup samples together with a user defined number of shifts on each level. From these samples, $\mathbb{V}[\Delta P_\ell]$ and $\mathbb{V}\left[\Delta Q_\ell^{\text{MLQMC}}\right]$ are estimated on each level ℓ, see Eq. (5). The iterative step of the doubling algorithm consists of selecting the level τ on which the ratio of the contribution to the total variance of the estimator, with the sample cost is maximal, i.e., $\underset{\tau\in\text{L}}{\text{argmax}}\left(\mathbb{V}[\Delta P_\tau]/(C_\tau R_\tau N_\tau)\right)$. On this level τ, the number of samples is multiplied with a constant factor. This procedure is repeated until $\mathbb{V}\left[Q_{\text{L}}^{\text{MLQMC}}\right] < \frac{\varepsilon^2}{2}$. In our approach, this constant is chosen as 1.2.

2.4 Mesh Hierarchies

In the multilevel setting, the levels $\ell = \{0,\ldots,\text{L}\}$ refer to the meshes in the mesh hierarchy. The coarsest mesh is denoted as level 0, while subsequent refinements of the coarse mesh are denoted as level 1, level 2,... Classically, the mesh hierarchy in the ML(Q)MC method is constructed starting from a coarse Finite Element mesh, on which h-refinement is applied recursively, see [8]. However, this approach leads to an almost geometric increase in the number of degrees of freedom (DOF) per increasing level, resulting in a high computational cost. Therefore, we propose a mesh hierarchy based on a p-refinement approach, i.e., increasing the polynomial's order of the shape functions per increasing level. With such an approach, the DOFs do not increase geometrically, see Table 1. This mesh hierarchy applied to the slope stability problem, is shown in Figure 2. Here, the Finite Element nodal points are represented as black dots. In Figure 2, we also present the h-refined mesh hierarchy of the slope stability problem.

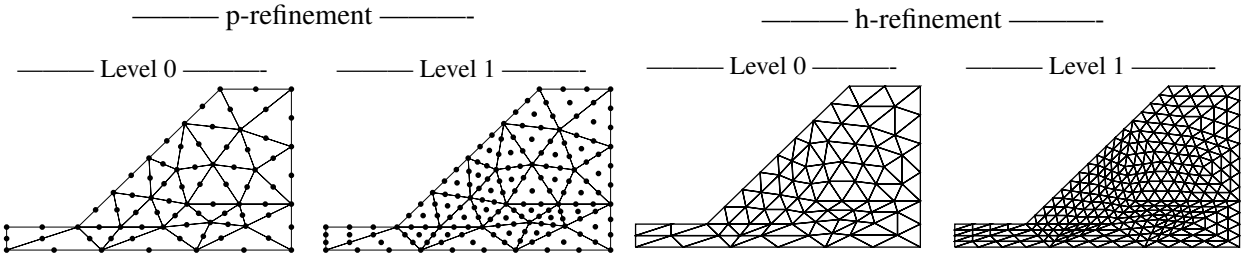

Figure 2: p- and h-refined hierarchy of meshes used for the slope stability problem.

2.5 Incorporating the Uncertainty

In the slope stability problem, the soil's cohesion is uncertain. This uncertainty is modeled by means of a random field, which is constructed by means of the Karhunen Loève (KL) expansion,

$$Z(\mathbf{x},\omega) = \overline{Z}(\mathbf{x}) + \sum_{n=1}^{s} \sqrt{\theta_n}\xi_n(\omega)b_n(\mathbf{x}). \tag{10}$$

Here, $\overline{Z}(\mathbf{x})$ is the mean of the field and $\xi_n(\omega)$ denote i.i.d. standard normal random variables. The symbols θ_n and $b_n(\mathbf{x})$ denote the eigenvalues and eigenfunctions respectively, which are the solutions of the eigenvalue problem $\int_D C(\mathbf{x},\mathbf{y})b_n(\mathbf{y})\text{d}\mathbf{y} = \theta_n b_n(\mathbf{x})$ with a given covariance kernel $C(\mathbf{x},\mathbf{y})$. Note that in order to represent the uncertainty of the soil's cohesion we do not use $Z(\mathbf{x},\omega)$ but $\exp(Z(\mathbf{x},\omega))$, see §3.

The uncertainty is represented by point evaluations of the random field resulting from the evaluation of Eq.(10) at certain evaluation points contained in the set $\mathbf{x}$. Because we consider a multilevel hierarchy, a set of evaluation points on each level is needed, i.e., $\mathbf{x}_\ell$ for $\ell = \{0,\ldots,\text{L}\}$.

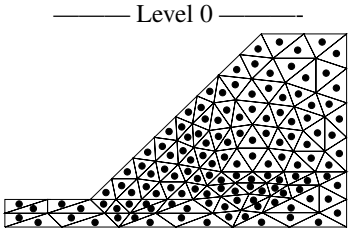

Figure 3: Locations of the random field evaluation points for the midpoint approach represented as dots.

In order to incorporate the uncertainty in the Finite Element model, we consider two different methods, the midpoint method and the integration point method. In both methods the uncertainty resides in the elastoplastic constitutive matrix $\mathbf{D}$. This matrix is used for constructing the element stiffness matrices by integrating the following expression,

$$\mathbf{K^e} = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega_e, \tag{11}$$

with the matrix $\mathbf{B}$ containing the derivatives of the element shape function. In practice, the right-hand side expression in Eq.(11) is computed as follows,

$$\mathbf{K^e} = \sum_{i=1}^{|\mathbf{q}|} \mathbf{B}_i^T \mathbf{D}_i \mathbf{B}_i \mathrm{w}_i, \tag{12}$$

with $\mathbf{B}_i$ the matrix $\mathbf{B}$ evaluated at quadrature point $\mathbf{q}^i \in \mathbf{q}$, i.e., $\mathbf{B}(\mathbf{q}^i)$, $\mathbf{D}_i$ the matrix $\mathbf{D}$ containing the uncertainty, i.e., $\mathbf{D}(\omega_i)$, and w_i the quadrature weight.

2.5.1 Midpoint Method

The midpoint method is classically used in conjunction with the h-MLMC method. The evaluation points of the random field in the set $\mathbf{x}_\ell$ on levels $\ell = \{0, \dots \mathrm{L}\}$ are selected as the centroids of the elements. This is shown in Figure 3, where the dots represent the spatial locations of the centroids of the elements. In case of the midpoint method, the uncertainty inside each element is considered constant, i.e., $\mathbf{D}_1 = \mathbf{D}_2 = \cdots = \mathbf{D}_{|\mathbf{q}|}$, see Eq.(12). Note that the resolution of the random field increases with each level, i.e., $|\mathbf{x}_0| < |\mathbf{x}_1| < \cdots |\mathbf{x}_\mathrm{L}|$.

2.5.2 Integration Point Method

The p-MLQMC method uses the integration point method in order to incorporate the uncertainty in the Finite Element model. Here the uncertainty varies inside each individual element, i.e., $\mathbf{D}_1 \neq \mathbf{D}_2 \neq \cdots \neq \mathbf{D}_{|\mathbf{q}|}$, see Eq.(12). We consider two different approaches for the integration point method. The *Non-Nested Approach* (NNA) and the *Local Nested Approach* (LNA). These approaches differ in the selection process of the points used for the point evaluation of the random field. However, for both approaches, the user first needs to define the number of quadrature points to be used per level.

The selection process for NNA is as follows. Here, the evaluation points of the random field are selected in a local triangular coordinate system, and grouped in sets $\mathbf{x}_\ell^{\mathrm{local}}$ for $\ell = \{0, \dots, \mathrm{L}\}$. The sets $\mathbf{x}_\ell^{\mathrm{local}}$ are taken equal to the sets of quadrature points used for the numerical integration in Eq. (12), i.e., $\mathbf{x}_\ell^{\mathrm{local}} = \mathbf{q}_\ell$. Because $\mathbf{q}_0 \nsubseteq \mathbf{q}_\ell \nsubseteq \cdots \nsubseteq \mathbf{q}_\mathrm{L}$, it follows that $\mathbf{x}_0^{\mathrm{local}} \nsubseteq \mathbf{x}_\ell^{\mathrm{local}} \nsubseteq \cdots \nsubseteq \mathbf{x}_\mathrm{L}^{\mathrm{local}}$. This is illustrated in Figure 4 , where $\mathbf{q}_\ell^i \in \mathbf{q}_\ell$ is represented by $\triangle_\ell$. The evaluation points of the random field contained in the set $\mathbf{x}_\ell^{\mathrm{local}}$ are represented by $\bullet_\ell$. After their selection, the points in $\mathbf{x}_\ell^{\mathrm{local}}$ are mapped to a global mesh coordinate system, resulting in $\mathbf{x}_\ell$, in order to be used in Eq. (10). For NNA, there exists only one set of points $\mathbf{x}_\ell$ per level. Note that here also the resolution of the random field increases with each level because $|\mathbf{q}_0| < |\mathbf{q}_1| < \cdots |\mathbf{q}_\mathrm{L}|$.

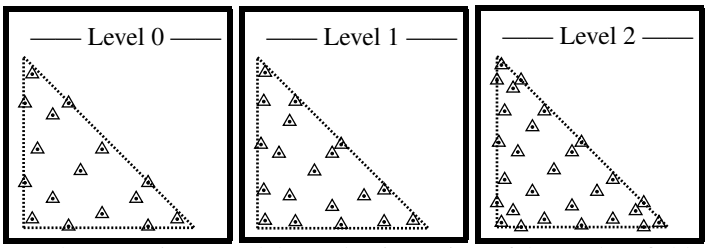

Figure 4: Locations of the quadrature points △ and the evaluation points of the random field ● on a reference triangular element for NNA.

For LNA, a different approach is used. The underlying idea consists of correlating the levels two-by-two. In practice this is done by constructing two sets of evaluation points of the random field, denote by $\mathbf{x}_{\ell,\text{coarse}}$ and $\mathbf{x}_{\ell,\text{fine}}$ for levels $\ell = \{1,\ldots,\text{L}\}$. This results in two representations of the random field per level, a coarse and a fine one. The coarse representation of the random field essentially acts as a representation of the field on level $\ell - 1$. This is because $\mathbf{q}_{\ell,\text{coarse}} = \mathbf{q}_{\ell-1,\text{fine}}$. The points in both sets are selected such that the sets are nested for given level ℓ, i.e., $\mathbf{x}_{\ell,\text{coarse}} \subseteq \mathbf{x}_{\ell,\text{fine}}$. The procedure starts on level $\ell = \{\text{L},\ldots,0\}$ by setting $\mathbf{x}^{\text{local}}_{\ell,\text{fine}} = \mathbf{q}_{\ell,\text{fine}}$. Then for level $\ell = \{\text{L},\ldots,1\}$ the points for set $\mathbf{x}^{\text{local}}_{\ell,\text{coarse}}$ are taken from the set $\mathbf{x}^{\text{local}}_{\ell,\text{fine}}$, such that each selected point is the closest neighbor to a point in the set $\mathbf{q}_{\ell,\text{coarse}}$. Note that here also the resolution of the random field increases with each level because $|\mathbf{q}_{0,\text{fine}}| < |\mathbf{q}_{1,\text{fine}}| < \cdots |\mathbf{q}_{\text{L},\text{fine}}|$, and $\mathbf{q}_{\ell,\text{coarse}} = \mathbf{q}_{\ell-1,\text{fine}}$. This is illustrated in Figure 5, where $\mathbf{q}^i_{\ell,\text{coarse}} \in \mathbf{q}_{\ell,\text{coarse}}$ is represented by $\triangle_{\ell,\text{coarse}}$, $\mathbf{q}^i_{\ell,\text{fine}} \in \mathbf{q}_{\ell,\text{fine}}$ by $\triangle_{\ell,\text{fine}}$. The points in $\mathbf{x}^{\text{local}}_{\ell,\text{coarse}}$ and $\mathbf{x}^{\text{local}}_{\ell,\text{fine}}$ are represented by $\bullet_{\ell,\text{coarse}}$ and by $\bullet_{\ell,\text{fine}}$. Following their selection, the points from $\mathbf{x}^{\text{local}}_{\ell,\text{coarse}}$ and $\mathbf{x}^{\text{local}}_{\ell,\text{fine}}$ are mapped to global coordinates, resulting in sets $\mathbf{x}_{\ell,\text{coarse}}$ and $\mathbf{x}_{\ell,\text{fine}}$.

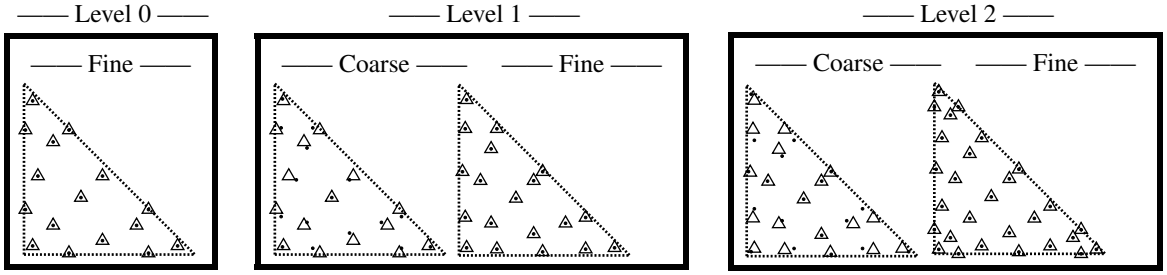

Figure 5: Locations of the quadrature points △ and the evaluation points of the random field ● on a reference triangular element for LNA.

3 MODEL PROBLEM

The model problem we consider for benchmarking the p-ML(Q)MC-LNA/NNA methods and the h-MLMC method, consists of a slope stability problem where the soil's cohesion has a spatially varying uncertainty, see [9]. In a slope stability problem, the safety of the slope can be assessed by evaluating the vertical displacement of the top of the slope when sustaining its own weight. We consider the displacement in the plastic domain, which is governed by the Drucker–Prager yield criterion. A small amount of isotropic linear hardening is taken into account for numerical stability reasons. Because of the nonlinear stress–strain relation arising in the plastic domain, a Newton–Raphson iterative solver is used. In order to compute the displacement in a slope stability problem, an incremental load approach is used, i.e., the total load resulting from the slope's weight is added in discrete load steps, starting with a force of 0 N. These loads steps are added until the downward force resulting from the slope's weight is reached. This approach results in the

following system of equations for the displacement,

$$\mathbf{K}\Delta\mathbf{u} = \mathbf{f} + \Delta\mathbf{f} - \mathbf{k}, \tag{13}$$

where $\Delta\mathbf{u}$ stands for the displacement increment and $\mathbf{K}$ the global stiffness matrix resulting from the assembly of element stiffness matrices $\mathbf{K}^{\mathbf{e}}$, see Eq. (11). The right hand side of Eq. (13) stands for the residual. Here, $\mathbf{f}$ is the sum of the external force increments applied in the previous steps, $\Delta\mathbf{f}$ is the applied load increment of the current step and $\mathbf{k}$ is the internal force resulting from the stresses. For a more thorough explanation on the methods used to solve the slope stability problem we refer to [10, Chapter 2 §4 and Chapter 7 §3 and §4].

The mesh hierarchies shown in Figure 2 are generated by using a combination of the open source mesh generator GMSH, see [11], and MATLAB, see [12]. Table 1 lists the number of elements (Nel), degrees of freedom (DOF), element order (Order), and the number of quadrature points per element (Nquad), per level for p-ML(Q)MC-LNA/NNA and h-MLMC. The number of quadrature points in the p-ML(Q)MC-LNA/NNA methods are chosen so as to increase the spatial resolution of the field per increasing level, and to ensure numerical stability of the computations of the displacement in the plastic domain. In this paper we consider two-dimensional uniform, Lagrange triangular elements.

Table 1: Characteristics of the mesh hierarchy for p-ML(Q)MC-LNA/NNA and h-MLMC.

p-ML(Q)MC-LNA/NNA				
Level	**Nel**	**DOF**	**Order**	**Nquad**
0	33	160	2	16
1	33	338	3	19
2	33	582	4	28
3	33	892	5	37
4	33	1268	6	61
5	33	1710	7	73
6	33	2218	8	126
h-MLMC				
Level	**Nel**	**DOF**	**Order**	**Nquad**
0	132	160	1	7
1	528	582	1	7
2	2112	2218	1	7
3	8448	8658	1	7
4	33792	34210	1	7
5	/	/	/	/
6	/	/	/	/

The quantity of interest (QoI) is taken as the vertical displacement in meters of the upper left node of the model. This is depicted in Figure 6 by the arrow.

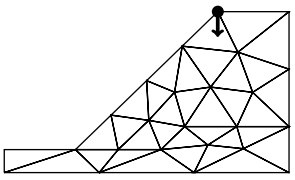

Figure 6: The QoI selected as the vertical displacement of the upper left node, indicated with an arrow.

The uncertainty of the cohesion of the soil is represented by means of a lognormal random field. This field is obtained by applying the exponential to the field obtained in Eq. (10), $Z_{\text{lognormal}}(\mathbf{x},\omega) = \exp(Z(\mathbf{x},\omega))$. For

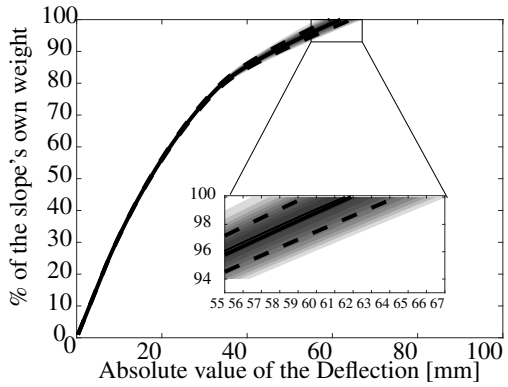

Figure 7: Uncertainty on the QoI

the covariance Kernel $C(\mathbf{x},\mathbf{y})$ of the random field, we use the Matérn covariance kernel,

$$C(\mathbf{x},\mathbf{y}) := \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{y}\|_2}{\lambda}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{\|\mathbf{x}-\mathbf{y}\|_2}{\lambda}\right), \tag{14}$$

with $\nu = 2.0$ the smoothness parameter, K_ν the modified Bessel function of the second kind, $\sigma^2 = 1$ the variance and $\lambda = 1.5$ the correlation length. The characteristics of the lognormal distribution used to represent the uncertainty of the cohesion of the soil are as follows: a mean of 8.02 kPa and a standard deviation of 400 Pa. The spatial dimensions of the slope are: a length of 20 m, a height of 14 m and a slope angle of 30°. The material characteristics are: a Young's modulus of 30 MPa, a Poisson ratio of 0.25, a density of 1330 kg/m^3 and a friction angle of 20°. Plane strain is considered for this problem. The number of stochastic dimensions considered for the generation of the Gaussian random field is $s = 400$, see Eq. (10). With a value $s = 400$, 99% of the variability of the random field is accounted for.

The stochastic part of our simulations was performed with the Julia packages **MultilevelEstimators.jl**, see [13], and **GaussianRandomFields.jl**, see [14]. The Finite Element code used, is an in-house MATLAB code developed by the Structural Mechanics Section of the KU Leuven. All the results have been computed on a workstation equipped with 2 physical cores, Intel Xeon E5-2680 v3 CPU's, each with 12 logical cores, clocked at 2.50 GHz, and a total of 128 GB RAM.

4 RESULTS

In this section we will discuss the results obtained with the p-ML(Q)MC-LNA/NNA and the h-MLMC methods.

4.1 Uncertainty on the QoI

In Figure 7, we show the uncertainty on the QoI, obtained with the p-MLQMC-LNA method. The shades of grey represent the probability density function (PDF), the black line represents the mean, and the black dotted lines represent the 1 sigma bounds.

4.2 Variance and Expected Value

In Figure 8 we show the sample variance over the levels $\mathbb{V}[P_\ell]$, the sample variance of the difference over the levels $\mathbb{V}[\Delta P_\ell]$, the expected value over the levels $\mathbb{E}[P_\ell]$ and the expected value of the difference over

the levels $\mathbb{E}[\Delta P_\ell]$.

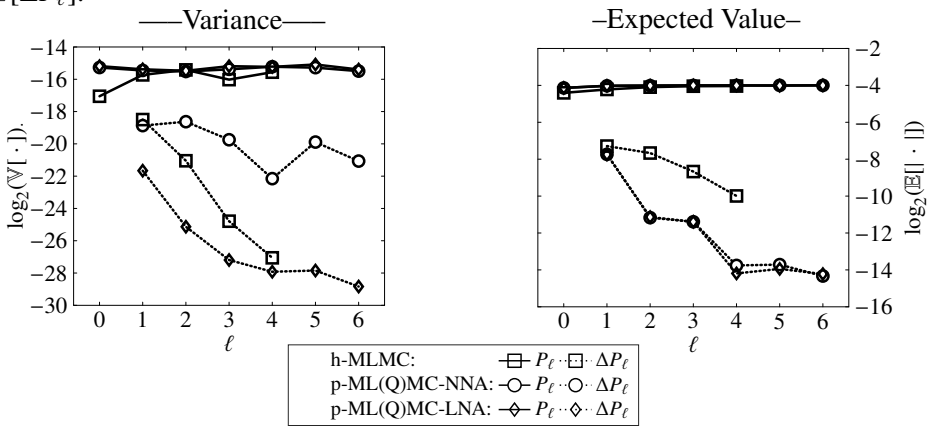

Figure 8: Variance and Expected Value over the levels.

As expected with multilevel methods, we observe that $\mathbb{E}[P_\ell]$ remains constant over the levels, while $\mathbb{E}[\Delta P_\ell]$ decreases with increasing level. As explained in §2.2, multilevel methods are based on a variance reduction by means of a hierarchical refinement of Finite Element meshes. In practice this means that the sample variance $\mathbb{V}[P_\ell]$ remains constant across the levels, while the sample variance of the difference over the levels $\mathbb{V}[\Delta P_\ell]$ decreases per increasing level. This is indeed what we observe for p-ML(Q)MC-LNA and h-MLMC. For p-ML(Q)MC-NNA we observe that $\mathbb{V}[\Delta P_\ell]$ does not decrease. From Figure 8, we can conclude that the choice of the evaluation points for the random field greatly influences the behavior of $\mathbb{V}[\Delta P_\ell]$ in the p-MLQMC method.

4.3 Samples

In this section we show the number of samples for p-ML(Q)MC-LNA/NNA and h-MLMC for a tolerance on the RMSE of $7.310\text{x}10^{-5}$, see Figure 9.

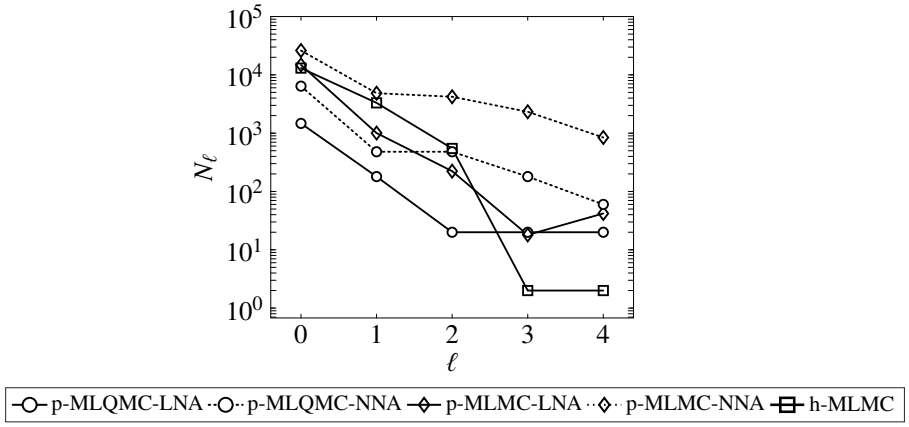

Figure 9: Number of samples for a tolerance on the RMSE of $7.310\text{x}10^{-5}$.

We observe that the number of samples for p-ML(Q)MC-LNA is lower than for p-ML(Q)MC-NNA. This is due to the 'bad' decrease of $\mathbb{V}[\Delta P_\ell]$ when NNA is combined with p-ML(Q)MC. The horizontal plateaus, e.g., the horizontal line created by connecting the number of samples of level 2, 3 and 4 for p-MLQMC-LNA, indicate that no additional samples have yet been taken in addition to the initial startup samples. For stricter tolerances, additional samples will be taken on these finer levels.

4.4 Runtimes

We show the absolute and relative runtime as a function of the user requested tolerance ε on the RMSE in Figure 10.

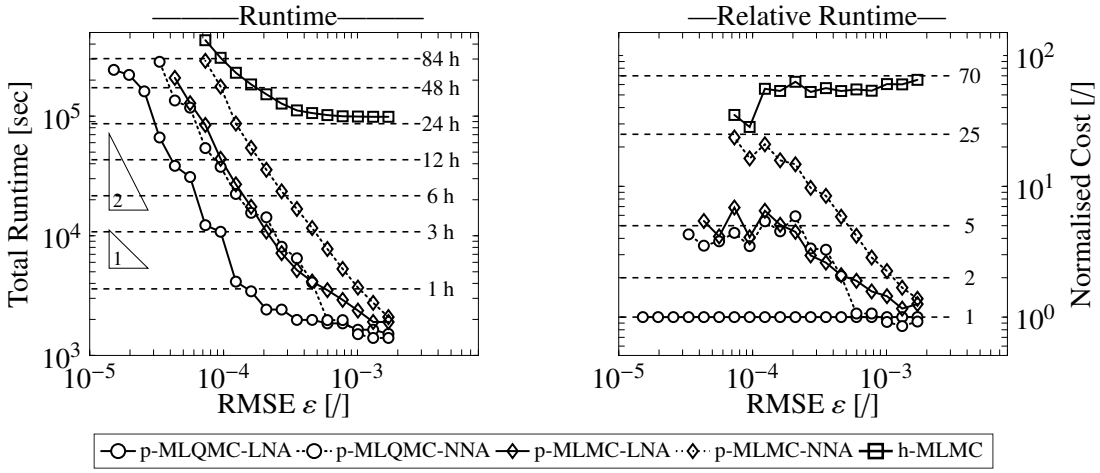

Figure 10: Absolute runtimes in function of requested user tolerance.

The results for the absolute runtime are expressed in seconds. For the relative runtime, we have normalised the computational cost of all the methods such that the results for p-MLQMC-LNA have unity cost for each tolerance.

The p-MLQMC-LNA method outperforms all other considered methods in terms of computational cost. We observe a speedup up to a factor 70 with respect to h-MLMC, and a factor 24 and 7 with respect to p-MLMC-NNA and p-MLMC-LNA respectively. p-MLQMC-LNA outperforms p-MLQMC-NNA by a factor 2 to 5. Interestingly, we observe that for this specific engineering problem, the computational cost of p-MLQMC-NNA is in the same range as the one of p-MLMC-LNA.

We conclude that a better correlation between the levels, i.e., using LNA instead of NNA, in the p-MLQMC method leads to a lower computational cost.

5 CONCLUSION

In this work, we have benchmarked the p-MLQMC method on a slope stability problem where the soil has a spatially varying uncertainty. In conjunction with this, we also investigated how the evaluation points of the random field, evaluated by means of the Karhunen-Loève expansion, are to be selected in the p-MLQMC method in order to obtain a lower computational cost. We distinguished two different approaches, the *Non-Nested Approach* (NNA) and the *Local Nested Approach* (LNA). We showed that the approaches impact the variance reduction over the levels $\mathbb{V}[\Delta P_\ell]$, and thus the total computational cost. p-ML(Q)MC combined with LNA exhibits a much better decrease of $\mathbb{V}[\Delta P_\ell]$ due to a better correlation between the levels than if it were combined with NNA. This is reflected in the total computational cost where p-MLQMC-LNA outperforms p-MLQMC-NNA by a factor ranging between 2 to 5. We also showed that the p-MLQMC-LNA method outperforms classic Multilevel Monte Carlo (h-MLMC) by a factor ranging between 24 and 70. Furthermore, p-MLQMC-LNA was benchmarked against p-MLMC-NNA and p-MLMC-LNA, outperforming both methods by a factor 24 and 7 respectively. From these findings, we conclude that, of the considered approaches, the p-MLQMC-LNA method offers the lowest computational cost for a given tolerance on the RMSE.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from the Research Council of KU Leuven through project C16/17/008 "Efficient methods for large-scale PDE-constrained optimization in the presence of uncertainty and complex technological constraints". The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI.

REFERENCES

[1] Giles, M. B. Multilevel Monte Carlo methods. *Acta Num.* (2015)**24**:259–328.

[2] Giles, M. B. Multilevel Monte Carlo path simulation. *Oper. Res.* (2008)**56(3)**:607–617.

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

[4] Giles, M. B. and Waterhouse, B. J. Multilevel Quasi-Monte Carlo path simulation. *Rad. Ser. Comp. App.* (2009)**8**:1–18.

[5] Blondeel, P., Robbe, P., Van hoorickx, C., Lombaert, G., and Vandewalle, S. Multilevel sampling with Monte Carlo and Quasi-Monte Carlo methods for uncertainty quantification in structural engineering. In *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP13, Seoul, South Korea*. Published in SNU repository (2019) pages 383–390.

[6] Blondeel, P., Robbe, P., Van hoorickx, C., François, S., Lombaert, G., and Vandewalle, S. p-refined Multilevel Quasi-Monte Carlo for Galerkin Finite Element Methods with Applications in Civil Engineering. *Algorithms* (2020)**13(5)**:1–30.

[7] Kuo, F. Lattice rule generating vectors (2007). Online `https://web.maths.unsw.edu.au/~fkuo/lattice/index.html` and `https://web.maths.unsw.edu.au/~fkuo/lattice/lattice-32001-1024-1048576.3600`, accessed on 12/04/2019.

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

[9] Whenham, V., De Vos, M., Legrand, C., Charlier, R., Maertens, J., and Verbrugge, J.-C. Influence of soil suction on trench stability. In T. Schanz, editor, *Experimental Unsaturated Soil Mechanics*. Springer Berlin Heidelberg (2007) pages 495–501.

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

[11] Geuzaine, C. and Remacle, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Meth. Eng.* (2009)**79(11)**:1309–1331.

[12] MATLAB. *version 9.2.0 (R2017a)*. The MathWorks Inc., Natick, Massachusetts (2017).

[13] Robbe, P. Multilevelestimators.jl (2018). Online `https://github.com/PieterjanRobbe/MultilevelEstimators.jl`, accessed on 05/11/2020.

[14] Robbe, P. Gaussianrandomfields.jl (2017). Online `https://github.com/PieterjanRobbe/GaussianRandomFields.jl`, accessed on 05/11/2020.