

# An overview of p-refined Multilevel quasi-Monte Carlo Applied to the Geotechnical Slope Stability Problem

Philippe Blondeel<sup>1</sup>, Pieterjan Robbe<sup>1</sup>, Stijn François<sup>2</sup>, Geert Lombaert<sup>2</sup>  
and Stefan Vandewalle<sup>1</sup>

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

<sup>2</sup> 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 popular sampling method consists of the classic Multilevel Monte Carlo method (h-MLMC). Its most distinctive feature consists of a hierarchy of h-refined meshes, where most of the samples are taken on coarse and computationally inexpensive meshes, and few are taken on finer but computationally expensive meshes. We present an improvement upon the classic Multilevel Monte Carlo, called the p-refined Multilevel quasi-Monte Carlo method (p-MLQMC). Its key features consist of a mesh hierarchy constructed from a p-refinement scheme combined with a deterministic set of samples points (quasi-Monte Carlo points). In this work we show how the uncertainty needs to be accounted for and present results comparing the total computational cost of the h-ML(Q)MC and p-MLQMC method. Specifically, we present two novel approaches in order to account for the uncertainty in case of p-MLQMC. We benchmarking the different multilevel methods on a slope stability problem, and find that p-MLQMC outperforms h-MLMC up to several orders of magnitude.*

## 1 INTRODUCTION

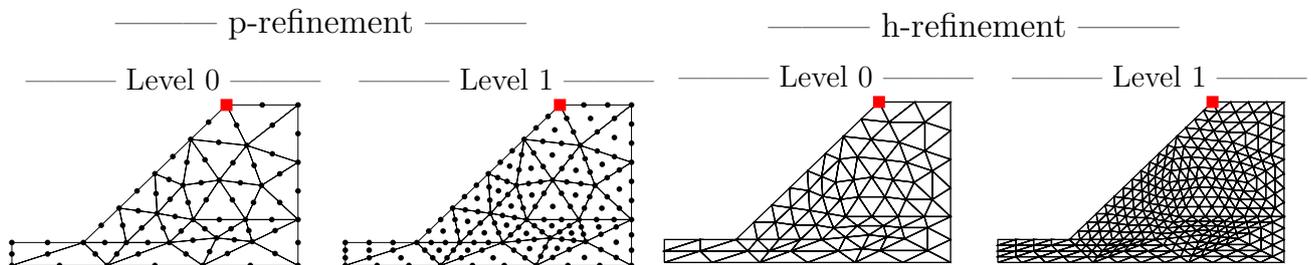
Problems in the engineering sciences are typically subject to uncertainty. In order to assess the uncertainty on the solution of the considered engineering problem, different steps need to be taken. First, the engineering problem is discretized, i.e., the underlying partial differential equation (PDE) governing the problem is approximated, by for example, the Bubnov–Galerkin Finite Element method. Second, the uncertainty present in the material parameters of the model, is to be represented as accurately as possible. Here, we chose to represent the uncertainty by means of a random field obtained through a Karhunen–Loève expansion (KL). Third, the modeled uncertainty needs to be accounted for in the Finite Element method. We consider two methods to achieve this step, i.e., the midpoint method and the integration point method. Fourth, the uncertainty on the solution is to be assessed. A straightforward manner to accomplish this last step, is by means of a stochastic sampling method. A well-known stochastic sampling method consists of the classic Multilevel Monte Carlo (h-MLMC) method. First developed by Giles, see [1, 2], the h-MLMC method 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. In previous

work, we introduced the p-refined Multilevel Quasi-Monte Carlo method (p-MLQMC), see [3], which essentially combines a mesh hierarchy based on a p-refinement scheme, i.e., increasing the polynomial order of the elements's shape function, together with a quasi-Monte Carlo sampling rule based on a rank-1 lattice sequence, e.g., [4]. This combination yields significant computational cost savings with respect to classic Multilevel (quasi-) Monte Carlo (h-ML(Q)MC). When accounting for the uncertainty in the Finite Element model, we observed a greater challenge with the p-MLQMC method than with the h-ML(Q)MC method. In our implementation, h-ML(Q)MC makes use of the midpoint method, while p-MLQMC makes use of the integration point method. In this work we present two novel approaches in order to implement the integration point method, with respect to our previous work see [3], i.e., the Local Nested Approach (LNA) and the Non-Nested Approach (NNA). In addition to this, we will benchmark the h-ML(Q)MC method against the p-MLQMC method on a slope stability problem where the cohesion of the soil is uncertain. The slope stability problem is a geotechnical engineering problem, where the goal is to assess the stability of natural or man-made slopes.

The paper is structured as follows. First we introduce the considered model problem. Second, we present the theoretical background pertaining to multilevel methods. Third, we discuss how the uncertainty is modeled as a random field and focus on how to account for said uncertainty in the Finite Element model. Last, we present the results obtained for p-MLQMC coupled with LNA and NNA, and h-ML(Q)MC with the midpoint method.

## 2 MODEL PROBLEM

The model problem we consider for benchmarking the methods, consists of a slope stability problem where the soil's cohesion has a spatially varying uncertainty, see [5]. We will discuss how to model this uncertainty in §4. 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. Different discretizations of the slope stability problem are presented in Figure 1.



**Figure 1:** An example of a fine and a coarse mesh used for the slope stability problem with the location of QoI indicated by ■.

We consider the displacement in the plastic domain, which is governed by the Drucker–Prager yield criterion. In the plastic domain, the stress–strain relation has a nonlinear behavior. Therefore, in order to compute a strain increment given a stress increment, an elastic predictor–plastic corrector iterative solver is used. In literature, this is commonly referred to as the ‘Return Mapping algorithm’, e.g., [6]. The governing partial differential equations are discretized by means of the Bubnov–Galerkin Finite Element method, giving rise to a system of equations. In order to compute the displacement, 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 load steps are added until the total downward force resulting from the slope’s weight is reached. The discretized system of equation, describing the displacement, that needs to be solved iteratively

by a Newton–Raphson solver is given as

$$\mathbf{K}\Delta\mathbf{u} = \mathbf{f} + \Delta\mathbf{f} - \mathbf{k}, \quad (1)$$

where  $\Delta\mathbf{u}$  stands for the displacement increment and  $\mathbf{K}$  the global stiffness matrix resulting from the assembly of element stiffness matrices  $\mathbf{K}^e$ . The right hand side of Eq. (1) 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 [7, Chapter 2 §4 and Chapter 7 §3 and §4].

### 3 SAMPLING AND MESH HIERARCHIES

The expected value of a function  $P$  against an  $s$ -dimensional probability density function  $\phi$  is defined by

$$\mathbb{E}[P] := \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} P(x_1, \dots, x_s) \phi(x_1, \dots, x_s) dx_1 \cdots dx_s = \int_{\mathbb{R}^s} P(\mathbf{x}) \phi(\mathbf{x}) d\mathbf{x}. \quad (2)$$

In order to approximate the integral in Eq. (2), an equal-weight quadrature rule can be used. An example of such an equal-weight quadrature rules is the Monte Carlo method. In our case, the function  $P$  is obtained by means of a Finite Element method on a chosen discretization level  $L$ , which leads to a first approximation of the integral,  $\mathbb{E}[P] \approx \mathbb{E}[P_L]$ . The computation of the integral itself is performed by defining an estimator, leading to a second approximation,  $\mathbb{E}[P_L] \approx Q_L^{\text{ML(Q)MC}}$ .

#### 3.1 Multilevel Monte Carlo

In multilevel methods, the expected value of  $\mathbb{E}[P_L]$  is written as a telescoping sum

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{\ell=1}^L \mathbb{E}[P_\ell - P_{\ell-1}]. \quad (3)$$

The resulting MLMC estimator used for the approximation of Eq. (2) is then given as

$$Q_L^{\text{MLMC}} := \frac{1}{N_0} \sum_{n=1}^{N_0} P_0(\mathbf{x}_0^{(n)}) + \sum_{\ell=1}^L \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left( P_\ell(\mathbf{x}_\ell^{(n)}) - P_{\ell-1}(\mathbf{x}_\ell^{(n)}) \right) \right\}, \quad (4)$$

where  $\mathbf{x}_\ell^{(n)}$  stands for the  $n$ th sample point. In the MLMC estimator the  $\mathbf{x}_\ell^{(n)}$  are (pseudo-)randomly chosen points, which are distributed according to  $\phi(\cdot)$ , see Eq. (2). The expected value of the quantity of interest on the finest level  $\ell = L$ , 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  $\ell = \{1, \dots, L\}$ , hence the name ‘telescoping sum’. The variance of the MLMC estimator is given by

$$\mathbb{V}[Q_L^{\text{MLMC}}] = \sum_{\ell=0}^L \mathbb{V}[\Delta Q_\ell^{\text{MLMC}}] = \sum_{\ell=0}^L \frac{\mathbb{V}[\Delta P_\ell]}{N_\ell} \approx \sum_{\ell=0}^L \frac{V_\ell}{N_\ell} = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \frac{\left( \Delta P_\ell^{(n)} - \Delta Q_\ell \right)^2}{N_\ell}, \quad (5)$$

where  $\Delta Q_\ell := \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \Delta P_\ell^{(n)}$ , with  $\Delta P_\ell^{(n)} := P_\ell(\mathbf{x}_\ell^{(n)}) - P_{\ell-1}(\mathbf{x}_\ell^{(n)})$  and  $P_{-1} := 0$ . 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  $\ell$  continuously decreases, i.e.,  $\mathbb{V}[\Delta P_1] > \mathbb{V}[\Delta P_2] > \dots > \mathbb{V}[\Delta P_L]$ . This variance reduction is only obtained when a strong positive correlation is achieved between the results of two successive levels,  $P_\ell := P_\ell(\mathbf{x}_\ell^{(n)})$  and  $P_{\ell-1} := P_{\ell-1}(\mathbf{x}_{\ell-1}^{(n)})$ , i.e.,

$$\begin{aligned}\mathbb{V}[\Delta P_\ell] &= \mathbb{V}[P_\ell - P_{\ell-1}] \\ &= \mathbb{V}[P_\ell] + \mathbb{V}[P_{\ell-1}] - 2\text{cov}(P_\ell, P_{\ell-1}),\end{aligned}\quad (6)$$

where  $\text{cov}(P_\ell, P_{\ell-1}) := \rho_{\ell, \ell-1} \sqrt{\mathbb{V}[P_\ell] \mathbb{V}[P_{\ell-1}]}$  is the covariance between  $P_\ell$  and  $P_{\ell-1}$  with  $\rho_{\ell, \ell-1}$  the correlation coefficient.

### 3.2 Multilevel quasi-Monte Carlo

The MLQMC estimator is given by

$$Q_L^{\text{MLQMC}} := \frac{1}{R_0} \sum_{r=1}^{R_0} \frac{1}{N_0} \sum_{n=1}^{N_0} P_0(\mathbf{x}_0^{(r,n)}) + \sum_{\ell=1}^L \frac{1}{R_\ell} \sum_{r=1}^{R_\ell} \left\{ \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left( P_\ell(\mathbf{x}_\ell^{(r,n)}) - P_{\ell-1}(\mathbf{x}_\ell^{(r,n)}) \right) \right\}, \quad (7)$$

with its variance given by

$$\mathbb{V}[Q_L^{\text{MLQMC}}] = \sum_{\ell=0}^L \mathbb{V}[\Delta Q_\ell^{\text{MLQMC}}]. \quad (8)$$

In order to estimate  $\mathbb{V}[\Delta Q^{\text{MLQMC}}]$  we use the sample variance  $\mathcal{V}_\ell$  over the  $R_\ell$  independent shifts, see [8]

$$\mathcal{V}_\ell = \sum_{r=1}^{R_\ell} \frac{1}{R_\ell(R_\ell - 1)} \left( \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \Delta P_\ell^{(r,n)} - \Delta Q_\ell \right)^2, \quad (9)$$

where  $\Delta Q_\ell := \frac{1}{R_\ell} \sum_{r=1}^{R_\ell} \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \Delta P_\ell^{(r,n)}$ , with  $\Delta P_\ell^{(r,n)} := P_\ell(\mathbf{u}_\ell^{(r,n)}) - P_{\ell-1}(\mathbf{u}_\ell^{(r,n)})$  and  $P_{-1} := 0$ . While the MLMC method is based on (pseudo-)random distributed sample points, the MLQMC method uses deterministic sample points (QMC points),  $\mathbf{x}_\ell^{(r,n)}$ . More specifically, here we use a rank-1 lattice sequence. In order to recover unbiased estimates of the estimator, the computation of the estimator and its variance include an averaging over a number of shifts  $r = 1, 2, \dots, R_\ell$  on each level  $\ell$ . The procedure of random shifting consists of adding to each point of the lattice sequence, a uniformly distributed number  $\Xi_r \in [0, 1)^s$ , after which the fractional part is taken. This is illustrated in Figure 2. In our implementation  $R_\ell = 10$  for each  $\ell$ ,  $0 \leq \ell \leq L$ .

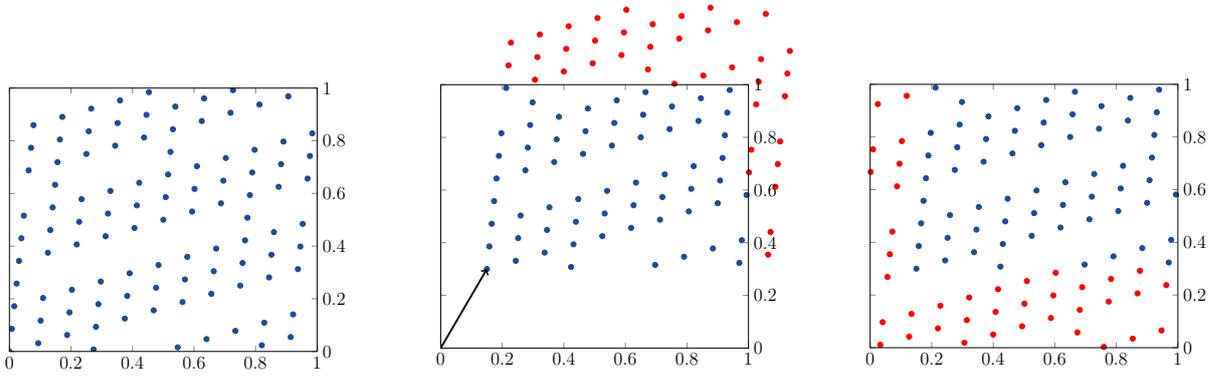
The shifted version of the lattice points is given by

$$\mathbf{x}^{(r,n)} := \Phi^{-1}(\text{frac}(\phi_2(n)\mathbf{z} + \Xi_r)), n \in \mathbb{N}, \quad (10)$$

where  $\Phi^{-1}$  is the inverse of the univariate standard normal cumulative distribution function,  $\text{frac}(x) := x - \lfloor x \rfloor$ ,  $x > 0$ ,  $\phi_2$  is the radical inverse function in base 2, and  $\mathbf{z}$  is an  $s$ -dimensional vector of positive integers. The generating vector  $\mathbf{z}$  was constructed with the component-by-component (CBC) algorithm with decreasing weights,  $\gamma_j = 1/j^2$ , see [9].

### 3.3 Mesh Hierarchies

In the multilevel setting, the levels  $0 \leq \ell \leq L$  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,  $\dots$ . Classically, the mesh hierarchy in the ML(Q)MC method is constructed starting from a coarse Finite Element mesh, to which h-refinement is recursively applied, see



**Figure 2:** Random shifting procedure applied to points belonging to a rank-1 lattice sequence.

[10]. Here, we use a mesh hierarchy based on a p-refinement approach, i.e., increasing polynomial order of the elements's shape function with increasing level. This mesh hierarchy applied to the slope stability problem, is shown in Figure 1. The Finite Element nodal points are represented as black dots. In Figure 1, we also present the h-refined mesh hierarchy of the slope stability problem.

### 3.4 Number of Samples

In Multilevel methods, the error is controlled by imposing a tolerance,  $\varepsilon^2$ , on the Mean Square Error (MSE) of the of the estimator. This MSE is defined as,

$$\begin{aligned} \text{MSE} \left[ Q_L^{\text{ML(Q)MC}} \right] &:= \mathbb{E} \left[ \left( Q_L^{\text{ML(Q)MC}} - \mathbb{E}[P] \right)^2 \right] \\ &= \mathbb{V} \left[ Q_L^{\text{ML(Q)MC}} \right] + \left( \mathbb{E} \left[ Q_L^{\text{ML(Q)MC}} \right] - \mathbb{E}[P] \right)^2 \\ &= \mathbb{V} \left[ Q_L^{\text{ML(Q)MC}} \right] + \left( \mathbb{E}[P_L - P] \right)^2. \end{aligned} \quad (11)$$

The right-hand side of Eq. (11) consists of two parts, i.e., the variance of the estimator,  $\mathbb{V} \left[ Q_L^{\text{ML(Q)MC}} \right]$ , and the squared bias,  $\left( \mathbb{E}[P_L - P] \right)^2$ . The stopping criterion for multilevel schemes is typically based on the requirements that both terms are less than  $\frac{\varepsilon^2}{2}$ . In order to achieve the requested tolerance for the variance of the estimator, the number of samples is increased. In the MLMC method, the optimal number of samples per level is given as

$$N_\ell = \frac{2}{\varepsilon^2} \sqrt{\frac{\mathbb{V}_\ell}{C_\ell}} \sum_{\ell=0}^L \sqrt{\mathbb{V}_\ell C_\ell}, \quad (12)$$

where  $\mathbb{V}_\ell$  stands for the sample variance, see Eq. (5), and  $C_\ell$  is the cost to compute one sample on level  $\ell$ , see [2]. However, in the MLQMC method, the number of samples to be taken is determined by means of a ‘doubling’ algorithm, see [4]. The procedure starts by computing a number of warm-up samples together with a user-defined number of shifts on each level. From these samples  $\mathbb{V} \left[ \Delta Q_\ell^{\text{MLQMC}} \right]$  is estimated on each level  $\ell$ , see Eq. (9). The iterative step consists of selecting the level  $\tau$  on which the ratio of the variance of the estimator with the sample cost is maximal, i.e.,  $\underset{\tau \in L}{\text{argmax}} (\mathbb{V}_\tau / C_\tau)$ . On this level  $\tau$  the number of samples is multiplied with a constant factor. This procedure is repeated until  $\mathbb{V} \left[ Q_L^{\text{MLQMC}} \right] < \frac{\varepsilon^2}{2}$ . In our approach, this constant is chosen as 1.2.

## 4 UNCERTAINTY MODELING AND INCORPORATION

The uncertainty present in the cohesion of the soil of the slope stability problem is modeled as a lognormal random field, i.e., the exponential of a Gaussian random field. Realizations of the Gaussian random field are computed by means of the truncated Karhunen–Loève (KL) expansion,

$$Z(\mathbf{x}, \omega) = \bar{Z}(\mathbf{x}) + \sum_{n=1}^s \sqrt{\theta_n} \xi_n(\omega) b_n(\mathbf{x}), \quad (13)$$

where  $s$  is the number of terms in the expansion, i.e., the number of stochastic dimensions. Here,  $\bar{Z}(\mathbf{x})$  is the mean of the field and  $\xi_n(\omega)$  denote i.i.d. standard normal random variables. The eigenvalues  $\theta_n$  and eigenfunctions  $b_n(\mathbf{x})$  are the solutions of the eigenvalue problem

$$\int_D C(\mathbf{x}, \mathbf{y}) b_n(\mathbf{y}) d\mathbf{y} = \theta_n b_n(\mathbf{x}), \quad (14)$$

where  $C(\mathbf{x}, \mathbf{y})$  is a given covariance kernel. The kernel we consider for the random field is the Matérn covariance kernel

$$C(\mathbf{x}, \mathbf{y}) := \frac{\sigma^2}{2^{\nu-1} \Gamma(\nu)} \left( \frac{\sqrt{2\nu} \|\mathbf{x} - \mathbf{y}\|_2}{\lambda} \right)^\nu K_\nu \left( \sqrt{2\nu} \frac{\|\mathbf{x} - \mathbf{y}\|_2}{\lambda} \right), \quad (15)$$

where  $\nu$  is the smoothness parameter,  $K_\nu(\cdot)$  is the modified Bessel function of the second kind,  $\Gamma(\cdot)$  is the gamma function,  $\sigma^2$  is the variance,  $\lambda$  is the correlation length, and  $\|\cdot\|_2$  is the  $L^2$  norm. The integral in Eq. (14) is approximated by means of a numerical collocation scheme. For more information, we refer to [11, Chapter 7 Section 2]. The lognormal representation of the random field is obtained by applying the exponential to the field obtained in Eq. (13),  $Z_{\text{lognormal}}(\mathbf{x}, \omega) = \exp(Z(\mathbf{x}, \omega))$ .

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,

$$\mathbf{K}^e = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega_e \approx \sum_{i=1}^{|\mathbf{q}|} \mathbf{B}_i^T \mathbf{D}_i \mathbf{B}_i w_i. \quad (16)$$

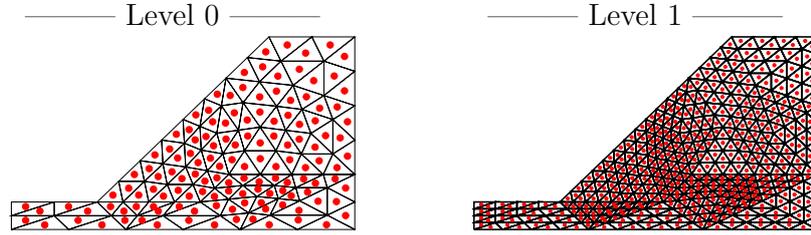
The matrix  $\mathbf{B}$  contains the derivatives of the element shape function, and  $|\mathbf{q}|$  is the number of quadrature points used for the numerical integration. The assembly of the element stiffness matrices results in the global stiffness matrix, see Eq. (1). In practice, the matrix  $\mathbf{K}^e$  is computed by means of a quadrature rule, where  $\mathbf{B}_i$  stands for 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.

We will now present the two methods used to account for the uncertainty in the Finite Element method. The goal consists of selecting the random field evaluation points  $\mathbf{x}$  used for the evaluation of Eq. (13). Because we are considering a multilevel approach, a set of random field evaluation points must be selected for each level, i.e.,  $\mathbf{x}_\ell$  for  $\ell = \{0, \dots, L\}$ .

### 4.1 Midpoint Method

The midpoint method is often used in conjunction with the h-ML(Q)MC method. The random field evaluation points are selected as the centroids of the elements, i.e., Eq. (13) is evaluated

at the centroids of the elements and the resulting values are assigned to the elements. This is shown in Figure 3, where  $\bullet$  represent the spatial locations of the centroids of the elements. In case of the midpoint method, the uncertainty inside each element is assumed to be constant, i.e.,  $\mathbf{D}_1 = \mathbf{D}_2 = \dots = \mathbf{D}_{|q|}$ , see Eq.(16). Note that the resolution of the random field increases with each level, i.e.,  $|\mathbf{x}_0| < |\mathbf{x}_1| < \dots < |\mathbf{x}_L|$ .



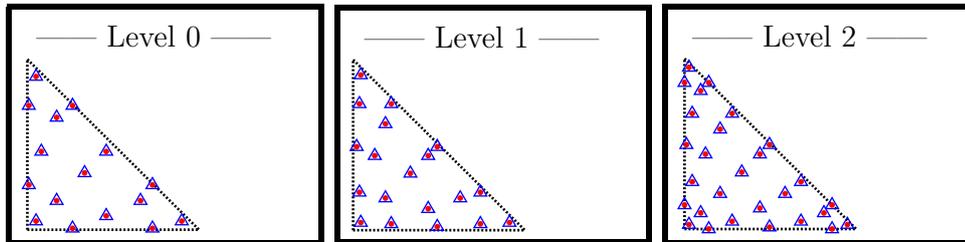
**Figure 3:** Locations of the random field evaluation points  $\bullet$  for the midpoint method.

## 4.2 Integration Point Method

In the p-MLQMC method, the number of elements in the hierarchy of mesh discretizations remains the same. Therefore, the midpoint method can not be used if we want the resolution of the random field to increase with increasing level. In order to obtain a higher resolution of the random field per increasing level, we use the integration point method, see [12], with the added condition that the number of quadrature points used to numerically integrate Eq. (16) also increases with increasing level. In the integration point method, Eq. (13) is evaluated at the locations of the quadrature points, or integration points, meaning that the uncertainty varies inside each individual element, i.e.,  $\mathbf{D}_1 \neq \mathbf{D}_2 \neq \dots \neq \mathbf{D}_{|q|}$ .

### 4.2.1 Non-Nested Approach

The Non-Nested Approach is the most simple way to select random field evaluation points. In this approach, the random field evaluation points are chosen equal to the quadrature points used for the numerical integration of Eq. (16). In practice, these quadrature points are first selected on a reference triangular element, see Figure 4, before being mapped to the global coordinates of the mesh. Note that the sets of quadrature points are not nested across the different levels, i.e.,  $\mathbf{q}_0 \not\subseteq \mathbf{q}_1 \not\subseteq \dots \not\subseteq \mathbf{q}_L$ . Hence the sets of random field evaluation points are not nested across the levels either, i.e.,  $\mathbf{x}_0 \not\subseteq \mathbf{x}_1 \not\subseteq \dots \not\subseteq \mathbf{x}_L$ . The obtained sets of random field evaluations points  $\mathbf{x}_\ell$ , with  $0 \leq \ell \leq L$ , are then used to compute discrete instances of the random field according to Eq. (13). As such, the random field  $Z(\mathbf{x}, \omega)$  is approximated on each level by a discrete set of random variables. Defining  $\mathbf{Z}_\ell := (Z(\mathbf{x}_\ell, \omega), \mathbf{x}_\ell)$  as the the set of random variables representing the random field and their locations, we see that those are not nested across levels, i.e.,  $\mathbf{Z}_0 \not\subseteq \mathbf{Z}_1 \not\subseteq \dots \not\subseteq \mathbf{Z}_L$ . This impacts the variance reduction, see Eq. (6), as it leads to a weak correlation between the solutions on successive levels.



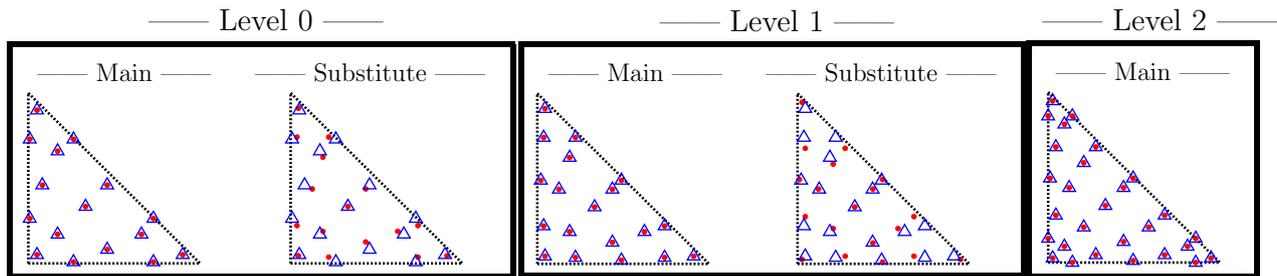
**Figure 4:** Locations of the quadrature points  $\triangle$  and the random field evaluation points  $\bullet$  on a reference triangular element for NNA.

## 4.2.2 Local Nested Approach

In the Local Nested Approach we try to improve the correlation between the solutions on different levels. Ideally, one would have  $\mathbf{Z}_0 \subseteq \mathbf{Z}_1 \subseteq \dots \subseteq \mathbf{Z}_L$ , i.e., the random field on each level is represented by using an exact subset of the information used to represent the random field on the finest level. Such an approach has been tried in [13], with limited success. Here, we suggest an alternative in which we only aim at a good correlation between each set of two successive levels in the mesh hierarchy. Such a two-by-two correlation is sufficient for multilevel sampling methods to achieve a rapid reduction of  $\mathbb{V}[\Delta P_\ell]$ .

Consider the correction  $\mathbb{E}[\Delta P_\ell] := \mathbb{E}[P_\ell - P_{\ell-1}]$ , which is one of the terms in the telescopic sum, Eq. (3). The integral for computing the element stiffness matrices in  $P_\ell$  makes use of the quadrature point set  $\mathbf{q}_\ell$ . At those points, we evaluate the random field  $Z(\mathbf{x}, \omega)$ , i.e., we set  $\mathbf{x}_\ell = \mathbf{q}_\ell$ . The integral for computing the element stiffness matrices in  $P_{\ell-1}$  makes use of the quadrature point set  $\mathbf{q}_{\ell-1}$ . However, we do not evaluate the random field at those locations, but rather evaluate the random field at points which are a subset of  $\mathbf{x}_\ell$ , i.e.,  $\mathbf{x}_{\ell-1, \text{subs}} \subseteq \mathbf{x}_\ell$ , such that they have minimal distance with  $\mathbf{q}_{\ell-1}$ . This is illustrated in Figure 5. (Note that this approximation is done on the level of the reference triangular element, before the mapping to the actual elements of the mesh.)

Define again  $\mathbf{Z}_\ell := (Z(\mathbf{x}_\ell, \omega), \mathbf{x}_\ell)$ , here with  $\mathbf{x}_\ell = \mathbf{q}_\ell$ , and  $\mathbf{x}_{\ell, \text{main}} := \mathbf{x}_\ell$ . The local nested approach ensures that, for each correction  $\mathbb{E}[\Delta P_\ell]$  separately, a relation  $\mathbf{Z}_{\ell-1, \text{subs}} \subseteq \mathbf{Z}_\ell$  is satisfied. Here,  $\mathbf{Z}_{\ell-1, \text{subs}}$  is a 'substitute random field', which approximates  $\mathbf{Z}_{\ell-1}$ . The substitute field correlates well with the discrete field on the  $\ell$ 'th level as it shares part of that field's random variables.



**Figure 5:** Locations of the quadrature points  $\triangle$  and the random field evaluation points  $\bullet$  on a reference triangular element for LNA.

An important note must be made concerning the LNA approach. While it successfully correlates the solutions of two successive levels, the expected value obtained from the telescoping sum is biased. We have observed a small bias of the order of  $10^{-6}$  with respect to the actual values, an error that is well below the discretization error of the finite element discretization. The reasons behind this additional bias stems from the fact that substitute random fields are used. We are currently investigating how this additional bias can be avoided.

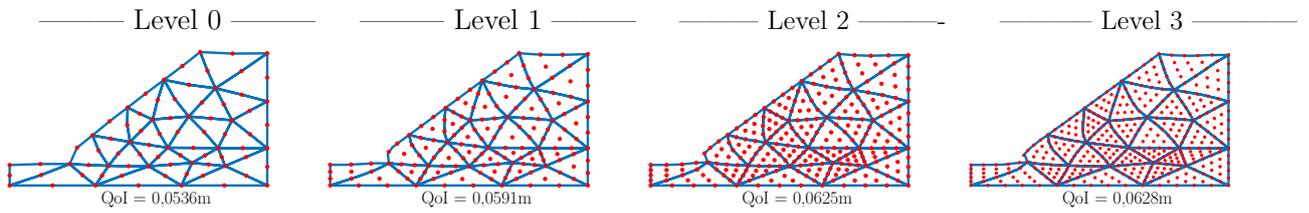
## 5 RESULTS

In this section we discuss the results obtained with the p-ML(Q)MC-LNA/NNA and the h-ML(Q)MC methods. The quantity of interest (QoI) is taken as the vertical displacement in meters of the upper left node of the model. This location of the QoI is depicted in Figure 1 by  $\blacksquare$ . The mesh hierarchies shown in Figure 1 are generated by using a combination of the open source mesh generator GMSH, see [14], and MATLAB, see [15]. In this paper we consider two-dimensional Lagrange triangular elements. The random field, computed by means of the Julia package **GaussianRandomFields.jl** [16] has the following parameters  $\nu = 0.4$ ,  $\sigma^2 = 1.0$ ,  $\lambda = 1.5$ . 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<sup>3</sup> and a friction angle of 20°. The number of stochastic dimensions considered for the generation of the Gaussian random field is  $s = 400$ , see Eq. (13). With a value  $s = 400$  at least 99% of the variability of the random field is accounted for. The stochastic sampling was performed with the Julia packages **MultilevelEstimators.jl**, see [17]. 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, Xeon Gold 6240 CPU’s, each with 18 logical cores, clocked at 2.60 GHz, and a total of 192 GB RAM.

### 5.1 Displacement of the Mesh

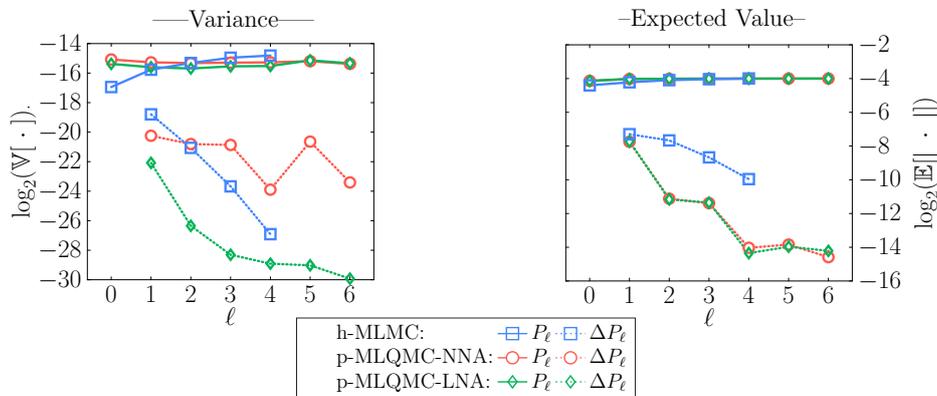
In Figure 6 we show the displacement of the mesh and the value of the QoI for four samples of the random field computed on the first four levels.



**Figure 6:** Displacement of the mesh and QoI for different samples of the random field.

### 5.2 Variance and Expected Value

In Figure 7 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  $\mathbb{E}[\Delta P_\ell]$ .

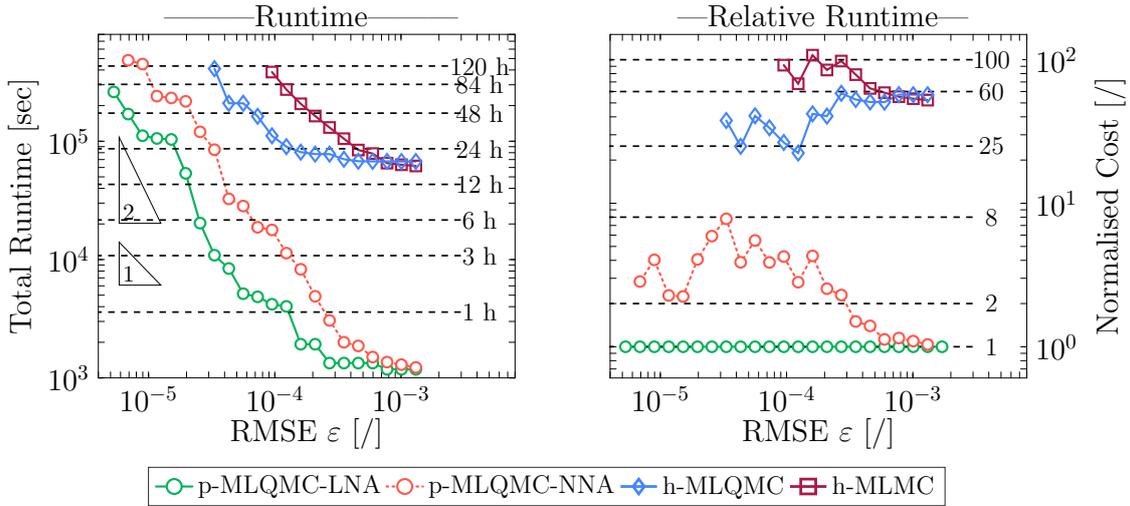


**Figure 7:** Variance and Expected Value over the levels.

As expected, 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 §3.1, multilevel methods are based on a variance reduction. 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-ML(Q)MC. For p-ML(Q)MC-NNA we observe that  $\mathbb{V}[\Delta P_\ell]$  does not decrease, but oscillates. From Figure 7, 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.

### 5.3 Runtimes

We show the absolute and relative runtime as a function of the user requested tolerance  $\varepsilon$  on the RMSE in Figure 8.



**Figure 8:** 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. We observe that p-MLQMC combined with the LNA approach outperforms all other considered methods. p-MLQMC-LNA outperforms p-MLQMC-NNA by a factor 2 to 8. In addition, the p-refined Multilevel methods outperform the h-refined Multilevel methods. p-refined MLQMC achieves a speedup up to a factor 60 with respect to h-MLQMC and a factor 100 with respect to h-MLMC.

## 6 CONCLUSION

In this work, we have benchmarked the p-MLQMC method on a slope stability problem where the soil has a spatially varying uncertainty. We also investigated how the evaluation points of the random field 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 and the Local Nested Approach. We showed that the approaches impact the variance reduction over the levels, and thus the total computational cost. p-MLQMC combined with LNA exhibits a much better decrease of  $\mathbb{V}[\Delta P_\ell]$  due to a better correlation between the levels than with NNA. This is reflected in the total computational cost where the LNA approach outperforms NNA by a factor between 2 to 8. We also showed that the p-MLQMC-LNA method outperforms h-Multilevel Monte Carlo (h-MLMC) by a factor ranging between 60 and 100, and classic Multilevel quasi-Monte Carlo (h-MLQMC) by a factor 25 to 60. 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., 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.
- [4] Giles, M. B. and Waterhouse, B. J. Multilevel Quasi-Monte Carlo path simulation. *Rad. Ser. Comp. App.* (2009)**8**:1–18.
- [5] 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.
- [6] Simo, J. C. and Taylor, R. L. A return mapping algorithm for plane stress elastoplasticity. *Int. J. Numer. Meth. Eng.* (1986)**22**(3):649–670.
- [7] de Borst, R., Crisfield, M. A., and Remmers, J. J. C. *Non Linear Finite Element Analysis of Solids and Structures*. Wiley, U.K. (2012).
- [8] Kuo, F. Y., Scheichl, R., Schwab, C., Sloan, I. H., and Ullmann, E. Multilevel Quasi-Monte Carlo methods for lognormal diffusion problems. *Math. Comput.* (2017)**86**(308):2827–2860.
- [9] 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.
- [10] 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.
- [11] Lord, G. J., Powell, C. E., and Shardlow, T. *An Introduction to Computational Stochastic PDEs*. Cambridge Texts in Applied Mathematics. Cambridge University Press (2014).
- [12] Matthies, H. G., Brenner, C. E., Bucher, C. G., and Guedes Soares, C. Uncertainties in probabilistic numerical analysis of structures and solids-stochastic finite elements. *Struct. Saf.* (1997)**19**(3):283–336.
- [13] Blondeel, P., Robbe, P., François, S., Lombaert, G., and Vandewalle, S. On the selection of random field evaluation points in the p-MLQMC method. *arXiv Preprint* (2020).
- [14] 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.
- [15] MATLAB. *version 9.2.0 (R2017a)*. The MathWorks Inc., Natick, Massachusetts (2017).
- [16] Robbe, P. Gaussianrandomfields.jl (2017). Online <https://github.com/PieterjanRobbe/GaussianRandomFields.jl>, accessed on 05/11/2020.
- [17] Robbe, P. Multilevelestimators.jl (2018). Online <https://github.com/PieterjanRobbe/MultilevelEstimators.jl>, accessed on 05/11/2020.