Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2021) 194-208

ECCOMAS
Proceedia

UNCECOMP 2021
4th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Streamed from Athens, Greece, 28 -30 June 2021

IMPROVING THE RATE OF CONVERGENCE OF THE QUASI-MONTE CARLO METHOD IN ESTIMATING EXPECTATIONS ON A GEOTECHNICAL SLOPE STABILITY PROBLEM

Philippe Blondeel[1], Pieterjan Robbe[1], Dirk Nuyens[1], Geert Lombaert[2] and Stefan Vandewalle[1]

[1] KU Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Leuven, Belgium
{philippe.blondeel, pieterjan.robbe, dirk.nuyens, stefan.vandewalle}@kuleuven.be
[2] KU Leuven, Department of Civil Engineering
Kasteelpark Arenberg 40, 3001 Leuven, Belgium
geert.lombaert@kuleuven.be

Abstract. *The propagation of parameter uncertainty through engineering models is a key task in uncertainty quantification. In many cases, taking into account this uncertainty involves the estimation of expected values by means of the Monte Carlo method. While the performance of the classical Monte Carlo method is independent of the number of uncertainties, its main drawback is the slow convergence rate of the root mean square error, i.e., $O(N^{-1/2})$ where N is the number of model evaluations. Under appropriate conditions, the quasi-Monte Carlo method improves the order of convergence to $O(N^{-1})$ by using deterministic sample points instead of random sample points. Two examples of such point sets are rank-1 lattice sequences and Sobol' sequences. However, it is possible to further improve the order of convergence by applying the so-called "tent transformation" to a rank-1 lattice sequence, and by "interlacing" a Sobol' sequence. In this work, we benchmark these two techniques on a slope stability problem from geotechnical engineering, where the uncertainty is located in the cohesion of the soil. The soil cohesion is modeled as a lognormal random field of which realizations are computed by means of the Karhunen–Loève (KL) expansion. The quasi-Monte Carlo points are mapped to the normal distribution required in the KL expansion using a novel truncation strategy. We observe an order of convergence of $O(N^{-1.5})$ in our numerical experiments.*

Keywords: Slope Stability, Geotechnical Engineering, Uncertainty Quantification, quasi-Monte Carlo Methods

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2021.
doi: 10.7712/120221.8032.18886

1 INTRODUCTION

In practical engineering problems, uncertainty plays an essential role. This uncertainty can, for example, be present in the material parameters such as the cohesion of the soil in a slope stability problem. In this type of problem, the goal is to assess the stability of natural or man-made slopes. Classically, this assessment is carried out in a deterministic way, i.e., no uncertainty is taken into account. However, this approach offers only limited insight. In order to gain a better insight into the stability of the slope, the uncertainty of the soil needs to be propagated through its mathematical model, which consists of a discretized partial differential equation (PDE). A popular and straightforward method to account for this uncertainty is by using a "sampling method". The most well known method belonging to this family is the Monte Carlo method, see [1]. In this method, the expected value of a user-chosen quantity of interest is computed as an average of multiple simulation outputs, each resulting from a different "sample" of the uncertainty. While the performance of the Monte Carlo method is independent of the stochastic dimension, i.e., the number of random variables used to represent the uncertain parameters, the computational cost measured in terms of the number of model evaluations is often still too large. This high computational cost stems on the one hand from the fact that all the samples are computed on one, possibly fine, discretization level (e.g., in order to approximate a PDE), and on the other hand from the slow convergence rate of the root mean square error, i.e., $O(N^{-1/2})$ where N is the number of samples. As engineering problems grow more complex and thus more costly, improvements which lower the computational cost of the Monte Carlo method have been proposed. One such improvement consists of converting the (single-level) Monte Carlo method into a Multilevel Monte Carlo method, see, e.g., [2]. In the Multilevel Monte Carlo method, the engineering problem is discretized multiple times with different mesh resolutions. The meshes resulting from the discretization are then grouped in a mesh hierarchy. The Multilevel Monte Carlo method achieves a speedup by taking many samples on computationally cheap low resolution meshes, and few samples on computationally expensive high resolution meshes. Another possible improvement consists of replacing the Monte Carlo sampling method by a quasi-Monte Carlo sampling method, see [3]. Instead of the random points used in the Monte Carlo method, the quasi-Monte Carlo method computes its samples at well chosen deterministic points. By using this approach, the order of convergence can be improved to $O(N^{-1})$, see [3, 4]. Most of these methods employ only a first-order Finite Element discretization of the underlying PDE. In previous work, see [5], we obtained an order of convergence close to $O(N^{-1})$ when combining the Multilevel Monte Carlo method with a quasi-Monte Carlo sampling method. In [6, 7], we combined the p-refinement of the Finite Element method (FEM) discretization with the Multilevel quasi-Monte Carlo sampling method, applied to a slope stability problem. There, the multilevel mesh hierarchy is constructed following a p-refinement approach, i.e., the order of the elements in the subsequent meshes is increased.

However, it is known that the $O(N^{-1})$ order of convergence can be improved for sufficiently smooth problems by using certain techniques, such as the *tent transformation* applied to a rank-1 lattice sequence [8] or by using an *interlacing* technique applied to a Sobol' sequence, see, e.g., [3, 9]. In this work, we investigate if a higher-order quasi-Monte Carlo convergence can be obtained in a single-level setting, by applying the *tent transformation* and the *interlacing* technique. The investigation is carried out by applying the above mentioned techniques on a slope stability problem, where realizations of the random field, that is used to model the uncertainty, are computed using a truncated KL expansion.

The slope stability problem itself is discretized by means of triangular quadratic Finite Elements.

The paper is structured as follows. First, we present the model problem, and briefly discuss the underlying Finite Element solver. Second, we review the theoretical background of the quasi-Monte Carlo method as well as the tent transformation and interlacing techniques. Last, we present numerical results obtained by tent-transformed rank-1 lattice sequences and interlaced Sobol' sequences. Both results will be compared to the ones obtained by means of the standard rank-1 lattice and Sobol' sequence.

2 MODEL PROBLEM AND MESH DISCRETIZATION

The model problem we consider consists of a slope stability problem where the cohesion of the soil has a spatially varying uncertainty, see [10]. In a slope stability problem, the safety of the slope can be assessed by evaluating the vertical displacement of a point near 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, an incremental load approach is applied, 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 total downward force resulting from the weight of the slope is reached. This approach results in the following system of equations for the displacement,

$$\mathbf{K}\,\Delta\mathbf{u} = \mathbf{f} + \Delta\mathbf{f} - \mathbf{k}, \tag{1}$$

where $\Delta\mathbf{u}$ stands for the displacement increment and $\mathbf{K}$ is the global stiffness matrix resulting from the assembly of element stiffness matrices $\mathbf{K}^{\mathbf{e}}$, see § 3.2. 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 [11, Chapter 2 §4 and Chapter 7 §3 and §4].

For the mesh discretization of the slope stability problem we use second-order triangular Lagrangian Finite Elements, see Fig. 1. Here, the Finite Element nodal points are represented as black dots.

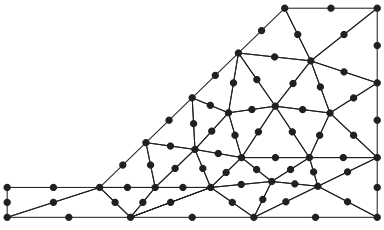

Figure 1: Mesh discretization used for the slope stability problem.

3 THEORETICAL BACKGROUND

In this section, we present some basic background on the usage of quasi-Monte Carlo (QMC) methods in estimating the expected value of a quantity of interest pertaining to the solution of

a PDE under uncertainty. We first explain the QMC estimator for estimating an integral and how to obtain an estimator on its variance, both for (tent-transformed) lattice sequences and (interlaced) Sobol' sequences. Next, we review how the uncertainty is modeled, in our slope stability problem, by means of a Karhunen–Loève expansion, and how it is accounted for in the equations of the Finite Element model. Last, we discuss how quasi-Monte Carlo points are generated according to (tent-transformed) lattice sequences and (interlaced) Sobol' sequences.

3.1 Quasi-Monte Carlo Estimator

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

$$\mathbb{E}[P] := \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} P(x_1, \ldots, x_s)\, \phi(x_1, \ldots, x_s)\, \mathrm{d}x_1 \cdots \mathrm{d}x_s = \int_{\mathbb{R}^s} P(\mathbf{x})\, \phi(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{2}$$

The integral in Eq. (2) can be approximated by means of an equal-weight quadrature rule, such as the Monte Carlo and quasi-Monte Carlo methods. In our setting, the function P stands for the quantity of interest based on the solution of our PDE which has stochastic parameters modeled by the variables $\mathbf{x} \sim \phi$. In order to approximate this expected value, both the multivariate integral, as well as the solution to the PDE for a given sample of the stochastic parameters, i.e., $P(\mathbf{x})$, will have to be approximated. We obtain an approximation for the quantity $P(\mathbf{x})$, resulting from our FEM discretization, which will be referred to as $P_L(\mathbf{x})$, where the L stands for the "discretization level". Likewise, we do not compute the exact solution of the multidimensional integral, but approximate it by means of the quasi-Monte Carlo method.

To approximate Eq. (2) we employ a randomized quasi-Monte Carlo estimator of the form

$$Q_L^{\mathrm{QMC}} := \frac{1}{R_L} \sum_{r=1}^{R_L} \frac{1}{N_L} \sum_{n=1}^{N_L} P_L(\Phi^{-1}(\mathbf{u}_L^{(n,r)})). \tag{3}$$

In here, $\mathbf{u}_L^{(n,r)}$ represent the points of our quasi-Monte Carlo point set, where n denotes the index of the point and r denotes the particular "random shift". Since quasi-Monte Carlo points are defined on the unit cube $[0,1]^s$ with respect to the uniform distribution, we need a mapping such that they act as samples from the density ϕ. For product densities, this can be achieved by applying the inverse of the cumulative distribution function component-wise. This is denoted by the mapping Φ^{-1}. Note that we have $\mathbb{E}[P] \approx \mathbb{E}[P_L] \approx Q_L^{\mathrm{QMC}}$. The "random shifting" and mapping will be explained in §3.3 and §3.3.3, see also Fig. 2 for an illustration of Monte Carlo (MC) sampling points versus quasi-Monte Carlo (QMC) sampling points.

The reason that we use "randomized" quasi-Monte Carlo estimators is to obtain an unbiased estimator, as well as an error estimator.

By means of the R_L independent random shifts, Eq. (3) is in fact averaged over R_L estimators. Hence, the variance of the estimator can be estimated by

$$\mathbb{V}\left[Q_L^{\mathrm{QMC}}\right] = \frac{\mathbb{V}[P_L]}{N_L R_L} \approx V_L^{\mathrm{QMC}} := \frac{1}{(R_L - 1)} \frac{1}{R_L} \sum_{r=1}^{R_L} \left(P_L^{(n,r)} - Q_L^{\mathrm{QMC}}\right)^2, \tag{4}$$

where $P_L^{(n,r)} := P_L(\Phi^{-1}(\mathbf{u}_L^{(n,r)}))$. From Eq. (4), the root mean square error is estimated as $\mathrm{RMSE}(Q_L^{\mathrm{QMC}}) = \sqrt{\mathbb{V}\left[Q_L^{\mathrm{QMC}}\right]} \approx \sqrt{V_L^{\mathrm{QMC}}}$. The RMSE will be used as an error estimator for the

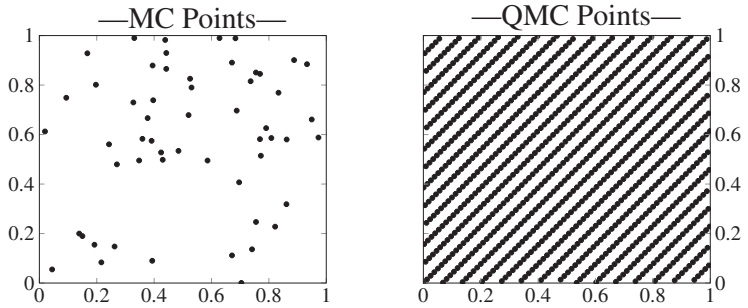

Figure 2: Example of MC and QMC sample points.

QMC estimator. We will plot this quantity in §4 in order to assess the accuracy of our numerical experiments.

3.2 Modeling the Uncertainty

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

$$Z(\mathbf{x},\omega) = \overline{Z}(\mathbf{x}) + \sum_{n=1}^{s} \sqrt{\theta_n}\,\xi_n(\omega)\,b_n(\mathbf{x}), \tag{5}$$

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

$$\int_D C(\mathbf{x},\mathbf{y})\,b_n(\mathbf{y})\,\mathrm{d}\mathbf{y} = \theta_n\,b_n(\mathbf{x}), \tag{6}$$

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

$$C(\mathbf{x},\mathbf{y}) := \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,\|\mathbf{x}-\mathbf{y}\|_2}{\lambda}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\,\frac{\|\mathbf{x}-\mathbf{y}\|_2}{\lambda}\right), \tag{7}$$

where ν is the smoothness parameter, $K_\nu(\cdot)$ is the modified Bessel function of the second kind, $\Gamma(\cdot)$ is the gamma function, σ^2 is the variance, λ is the correlation length, and $\|\cdot\|_2$ is the L^2 norm. The integral in Eq. (6) is approximated by means of a numerical collocation scheme. For more information, we refer to [12, Chapter 7 Section 2].

In order to incorporate the uncertainty in the Finite Element model, we consider the integration point method, i.e., point evaluations of the random field are computed by means of Eq. (5) at the quadrature points and accounted for during numerical integration of the element stiffness matrix, see [13]. Here, the uncertainty resides in the elastoplastic constitutive matrix $\mathbf{D}$. This matrix is used for constructing the element stiffness matrices

$$\mathbf{K^e} = \int_{\Omega_e} \mathbf{B}^T\mathbf{D}\mathbf{B}\,\mathrm{d}\Omega_e, \tag{8}$$

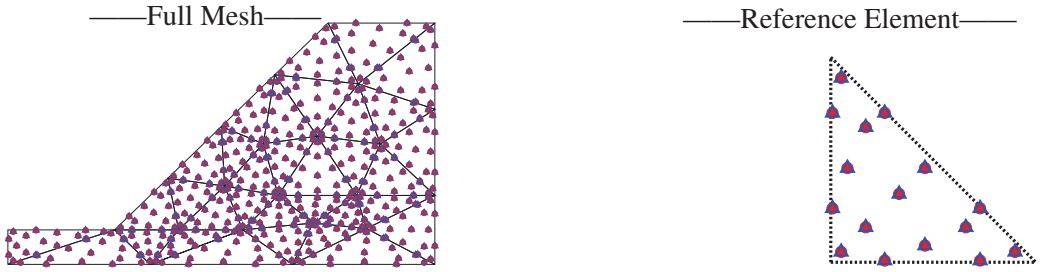

Figure 3: Locations of the quadrature points △ and the evaluation points of the random field ●.

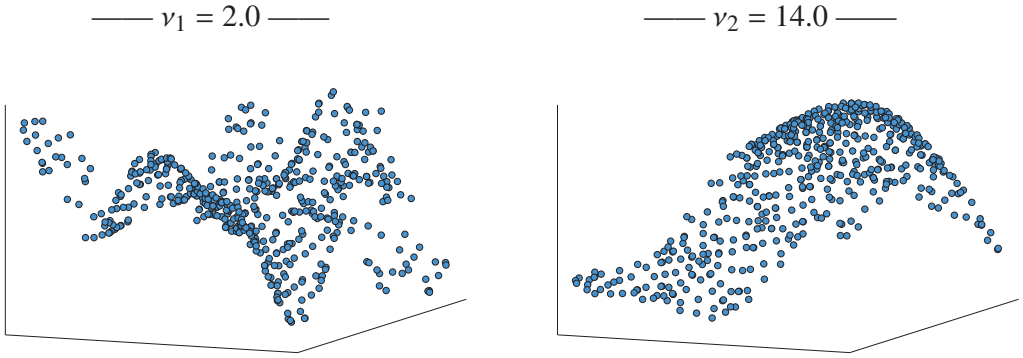

Figure 4: Point evaluations of an instance of a random field with $\nu_1 = 2.0$ and $\nu_2 = 14.0$.

with $\mathbf{B}$ the matrix containing the derivatives of the element shape functions. In practice, the right-hand side in Eq. (8) is computed as

$$\mathbf{K}^{\mathbf{e}} \approx \sum_{i=1}^{|\mathbf{q}|} \mathbf{B}_i^T \mathbf{D}_i \mathbf{B}_i \, w_i, \tag{9}$$

with $\mathbf{B}_i = \mathbf{B}(\mathbf{q}^{(i)})$ the matrix $\mathbf{B}$ evaluated at the ith quadrature point $\mathbf{q}^{(i)}$, $\mathbf{D}_i = \mathbf{D}(\omega^{(i)})$ the elasto-plastic constitutive matrix $\mathbf{D}$ containing the value of the uncertain soil cohesion $\omega^{(i)}$, computed as a point evaluation of the random field at $\mathbf{q}^{(i)}$, and w_i the quadrature weight.

With the integration point method, Eq. (5) is evaluated in the quadrature points used for the numerical integration of Eq. (9), i.e., $\mathbf{x} = \mathbf{q}$. This is illustrated in Fig. 3, where the set $\mathbf{q}$ is represented by the △, and the random field evaluation points, $\mathbf{x}$, are represented by ●.

In order to represent the uncertainty of the cohesion of the soil, we use a lognormal random field. This field is obtained by applying the exponential to the field obtained in Eq. (5) component-wise, i.e., $Z_{\text{lognormal}}(\mathbf{x}, \omega) = \exp(Z(\mathbf{x}, \omega))$. After this mapping, the lognormal random field has a mean of 8.02 kPa and a standard deviation of 100 Pa. We consider a random field with a correlation length $\lambda = 1.5$, a variance $\sigma^2 = 1$, and a stochastic dimension $s = 100$. Two different values for the smoothness ν of the random field are considered: $\nu_1 = 2.0$ and $\nu_2 = 14.0$. The smoothness parameter governs the smoothness of the random field: a lower value for ν implies a rougher random field and vice versa. In Fig. 4, we show instances of this random field, for the two different smoothness parameters ν_1 and ν_2.

3.3 Quasi-Monte Carlo Sampling

Quasi-Monte Carlo points are deterministic low-discrepancy points used for numerical integration. Different approaches exist to generate these points. We consider two approaches, rank-1 lattice sequences and Sobol' sequences. Classically, these point sets have been constructed in order to be used for integration against the uniform distribution on the unit cube $[0,1]^s$. In §3.3.3 we will describe how to use these point sets for integration against the normal distribution, as required in the KL expansion in Eq. (5). Important to note is that although we describe the technical details behind the construction of these point sets here, the end user can just use a library routine to generate them, see, e.g., [14, 15, 16].

3.3.1 Lattice Sequences

The points belonging to a rank-1 lattice sequence are determined by a generating vector $\mathbf{z} \in \mathbb{Z}^s$, which consists of one integer value per considered stochastic dimension s. The choice of this generating vector determines the quality of the sample points. When the total number of points is a fixed number N then the nth point of the "lattice rule" (instead of a lattice sequence) is given by

$$\mathbf{u}^{(n)} := \mathrm{frac}\left(\frac{n}{N}\mathbf{z}\right) \quad \text{for} \quad n = 0, ..., N-1, \tag{10}$$

where frac$(\cdot)$ denotes the function that takes the fractional part. When written in this form, the number of points cannot be extended beyond the maximal number of points N, and there is no guarantee that using an initial amount of these points will have a "nice" distribution in the unit cube. In order to obtain a sequence of "nicely" distributed sample points, we instead define the nth point by

$$\mathbf{u}^{(n)} := \mathrm{frac}\left(\phi_2(n)\,\mathbf{z}\right) \quad \text{for} \quad n = 0, 1, \ldots, \tag{11}$$

where ϕ_2 stands for the radical inverse function in base 2, see, e.g., [17, 18], and possibly limiting $n < N_{\max}$ for some large enough $N_{\max}$. The radical inverse function in base 2 mirrors the binary representation of a number around its binary point, e.g., $6 = (110)_2$, then $\phi_2((110)_2) = (0.011)_2 = 0.375$. Algorithms to find good generating vectors for Eq. (10) and Eq. (11) are known, and such vectors can be found in the literature, see, e.g., [18].

As already touched upon in §3.1, the deterministic nature of quasi-Monte Carlo sample points introduces an additional bias on the stochastic quantities of the computed solutions. Therefore, "randomness" needs to be reintroduced in order to obtain unbiased estimates. This is accomplished by a procedure called "random shifting". The procedure consists of adding to each point of the lattice sequence, a uniformly distributed number $\mathbf{w} \in [0,1)^s$, after which the fractional part is taken. This is illustrated in Fig. 5. By using R independent random shifts, the resulting R independent estimators can also be used to estimate the variance of our QMC estimator, and hence providing us with an error estimator.

The shifted version of Eq. (11) is then given by

$$\mathbf{u}^{(n,r)} := \mathrm{frac}\left(\phi_2(n)\,\mathbf{z} + \mathbf{w}_r\right) \qquad \text{for} \quad n = 0, 1, \ldots \quad \text{and} \quad r = 1, 2, \ldots, R. \tag{12}$$

The use of such a lattice sequence, constructed with a "good" generating vector, in the QMC estimator from Eq. (3), can achieve a theoretical order of convergence of $O(N^{-1})$ in a certain

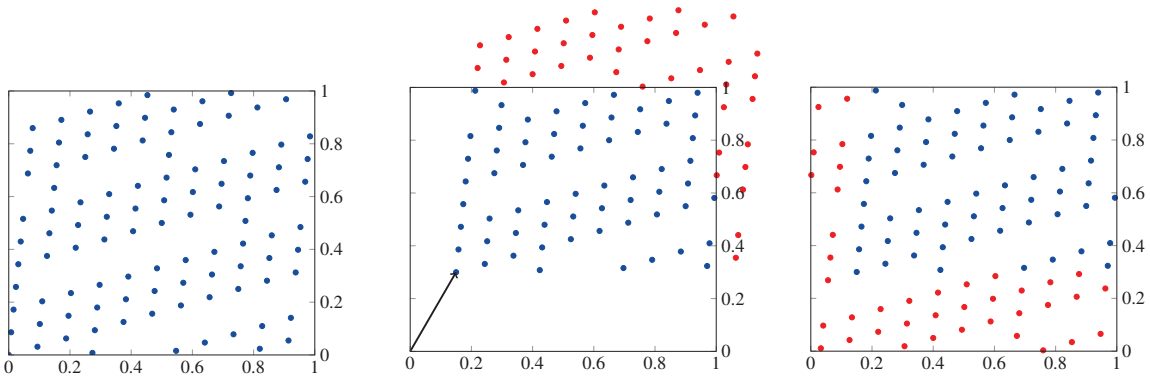

Figure 5: Random shifting procedure applied to points belonging to a rank-1 lattice sequence.

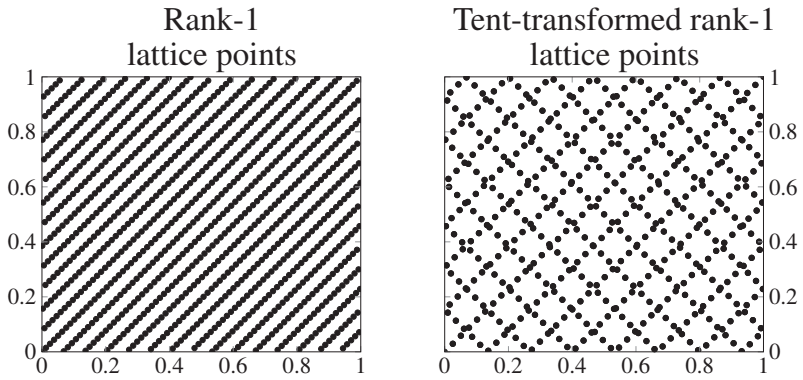

Figure 6: Rank-1 lattice points and tent-transformed rank-1 lattice points in the unit cube.

Sobolev space which contains integrands with square-integrable mixed first derivatives see, e.g., [4] for details. However, by applying the tent transformation

$$T(y) := 1 - |2y - 1|, \tag{13}$$

component-wise to the points generated by Eq. (12), and defining

$$\mathbf{v}^{(n,r)} := T\left(\mathbf{u}^{(n,r)}\right), \tag{14}$$

for usage in Eq. (3) instead of the original points $\mathbf{u}^{(n,r)}$, it is possible to obtain a better order of convergence than $O(N^{-1})$, provided that the integrand is sufficiently smooth, see, e.g., [8, 19, 20]. In our numerical experiments, this smoothness will be influenced by the smoothness parameter ν of the random field, and by the order of approximation of the FEM solution. We note that in order to achieve higher-order convergence it is necessary to only consider estimators where the value for N_L is a power of 2. This has the effect that the point set will act like a sequence of embedded lattice rules. See Fig. 6 for an illustration of tent-transformed lattice points.

3.3.2 Sobol' Sequences

The Sobol' sequence is a "digital net" in base 2, that uses a binary generating matrix per dimension. We denote these matrices by $C_1, \ldots, C_j$ for $j = 1, \ldots, s$. For the Sobol' sequence these

Decimal	Binary	Gray code	Gray code binary	Multiplication with C_4	Shifted with $(0.010)_2$	Decimal
0	000	0	000	0.000	0.010	0.25
1	001	1	001	0.100	0.110	0.75
2	010	3	011	0.010	0.000	0.00
3	011	2	010	0.110	0.100	0.50
4	100	6	110	0.111	0.101	0.625

Table 1: Gray coded and digitally shifted Sobol' points in dimension 4.

matrices are upper triangular. To obtain the jth dimension of the nth Sobol' point, we write $n = (n_{m-1} \cdots n_1 n_0)_2$ in binary representation, and use the $m \times m$ binary matrix C_j to compute

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} & & \\ & C_j & \\ & & \end{pmatrix} \begin{pmatrix} n_0 \\ \vdots \\ n_{m-1} \end{pmatrix}, \tag{15}$$

where all additions and multiplications are carried out modulo 2. The jth dimension of the nth point is then given by interpreting the output vector as the binary expansion, i.e., $u_j^{(n)} = (0.y_1 y_2 \cdots y_m)_2$. We demonstrate the approach stated above for the computation of the first five points of the fourth dimension. Suppose the 3×3 subset of the generating matrix C_4 is given by

$$C_4 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{16}$$

The first five points generated by C_4 for $n = 0,1,2,3,4$, which in binary are $0 = (000)_2, 1 = (001_2), 2 = (010)_2, 3 = (011)_2$ and $4 = (100)_2$, hence are equal to $(0.000)_2 = 0$, $(0.100)_2 = 0.5$, $(0.110)_2 = 0.75$, $(0.010) = 0.25$, and $(0.001)_2 = 0.125$. In practice, one avoids multiplying with the full matrix C_j by generating the points in Gray code ordering. This has the benefit that the next point can be obtained from the previous one by adding only the column of the generating matrix where the bit was changed. This changes the ordering of the Sobol' points, but they will still have their "nice" distribution properties.

In order to obtain an unbiased estimator, and an error estimator, we need to introduce some "randomness" on the deterministic points. This is accomplished by means of a "random digital shift". The digitally shifted nth Sobol' point is obtained by adding the bits of the binary expansion of the shift to each digit of the Sobol' point modulo 2, for each dimension, i.e., $\mathbf{u}^{(n,r)} = \mathbf{u}^{(n)} \oplus \mathbf{w}_r$. This is illustrated in Tab. 1 for the first 5 Gray coded points in dimension 4 with a shift $\mathbf{w}$ that has the value $w_4 = 0.25 = (0.010)_2$ as its fourth component.

Is has been shown in the work of Dick, see e.g., [21], that higher-order convergence can be obtained by a "digit interlacing" technique if the integrand is sufficiently smooth. Again, in our numerical experiments this smoothness will be influenced by the smoothness of the random field and by the order of approximation of the FEM solution. In order to obtain Sobol' points with interlacing factor α in s dimensions, one starts with Sobol' points in αs dimensions and then "interlaces" α dimensions into a single dimension by interlacing the bits of the points. For example, for an interlacing factor of 2 we take the binary representations of the Sobol' points in the first two dimensions as $x = (0.x_1 x_2 \cdots x_m)_2$ and $y = (0.y_1 y_2 \cdots y_m)_2$ and then form the point $(0.x_1 y_1 x_2 y_2 \cdots x_m y_m)_2$. In practice, the interlacing of the Sobol' sequence can also be done by interlacing the rows of the generating matrices of the original Sobol' sequence. The interlaced

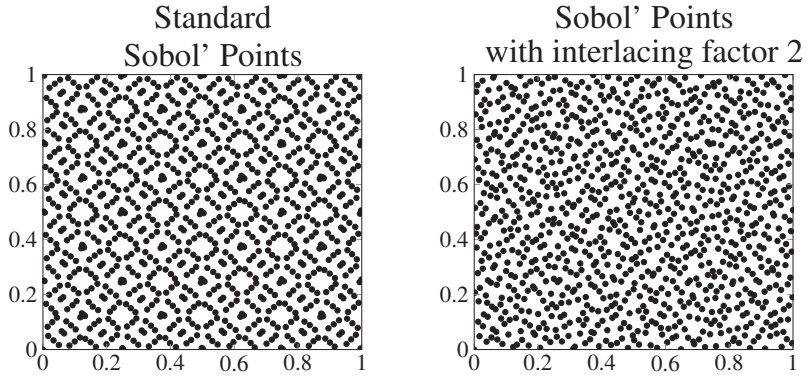

Figure 7: Standard Sobol' points and Sobol' point with interlacing factor 2.

Sobol' sequence points are then generated from these interlaced matrices which have dimension $\alpha m \times m$. As was the case with the tent-transformed lattice points, we need to ensure that we only consider estimators where N_L is a power of 2 in order to achieve higher order convergence. An illustration of interlaced Sobol' points is given in Fig. 7.

3.3.3 Using QMC points for integration against the truncated normal density

Up till now the discussion was centered on quasi-Monte Carlo points which are sampled uniformly from the unit cube $[0,1]^s$. In order to use these points for integration against some other distribution and domain, see Eq. (2), we need to map them to the adequate distribution. This can be achieved by applying the inverse of the cumulative distribution function component-wise.

Our modeling of the random field, see § 3.2, involves standard normally distributed numbers. However, for our experiments in which we hope to achieve a convergence better than order 1 in estimating the expectations, i.e., $O(N^{-1})$, we will truncate the domain from $(-\infty,\infty)^s$ to $[-b,b]^s$ for some choice of $0 < b < \infty$. With respect to our general scheme of approximating Eq. (2) this will introduce an additional domain truncation error in our sequence of approximations $\mathbb{E}[P] \approx \mathbb{E}[P_L] \approx Q_L^{\text{QMC}}$. There is no established analysis for obtaining higher order quasi-Monte Carlo convergence on the infinite domain $(-\infty,\infty)^s$, except for on the unit cube, see, e.g., [3, 9]. Therefore, we follow the truncation approach which has been used in other references as well, see, e.g., [22, 23]. The following errors need to balanced: (1) the Finite Element discretization error which is adjusted by choosing different discretization parameters L and p, (2) the dimension truncation error which results from truncating the infinite KL expansion to only s terms, see Eq. (5), (3) the domain truncation error which results from the truncation of $(-\infty,\infty)^s$ to $[-b,b]^s$ and (4) the quadrature/cubature approximation error which results from approximating an s-dimensional integral with an s-dimensional QMC rule using N_L sample points. The theoretical analysis and careful balancing of these different sources of error will be the topic of future research. In this work, we demonstrate numerically that it is possible to achieve a higher-order convergence.

4 Results

In this section, we present numerical results obtained by applying a tent-transformed lattice sequence and an interlaced Sobol' sequence to the slope stability problem introduced in §2.

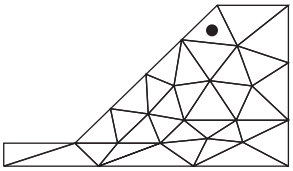

Figure 8: The QoI as the vertical displacement of the center point of the upper left element of the model, indicated by a dot.

Rank-1 lattice points are computed with the Julia package **LatticeRules.jl**, where the generating vector $\mathbf{z}$ was constructed with the component-by-component (CBC) algorithm with order 2 weights, see [24]. The Sobol' points are computed with the Julia package **DigitalNets.jl**, see [15]. We used the Sobol' points and the pre-interlaced Sobol' generating matrices from [16, 25]. Instances of the random field are computed with the Julia package **GaussianRandomFields.jl**, see [26]. We use the in-house Finite Element MATLAB code developed by the Structural Mechanics section of the KU Leuven. All 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.

As the quantity of interest (QoI), we choose the vertical displacement of the center point of the upper left element of the model, see Fig. 8. By choosing a QoI located inside the element, we ensure that the displacement is represented by a quadratic polynomial, since quadratic shape functions are used to compute a displacement at this center point. This approach is followed as to ensure that the higher-order derivatives are continuous. Between elements there exists only C0 continuity. However, inside the elements, the continuity is as high as the order of the shape functions used for the representation of the solution. Hence, we interpolate to a point located inside the element. The spatial dimensions of the slope, in our slope stability problem, consist of 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 characteristics of the random field considered to model the uncertainty in the cohesion of the soil, are given in § 3.2. For the truncation of the domain, see § 3.3.3, we take a value $b = 2$, and thus truncated the domain to $[-2, 2]^s$.

First, in §4.1, we numerically verify that each method computes the same expected value. Next, in §4.2, we show the convergence of the root mean square error of each method with respect to the number of samples taken.

4.1 Convergence of the Expected Value

In Fig. 9, we show the maximum absolute error on the bound of the 95% confidence interval of the expected value in function of the number of samples,

$$\text{Max Error} := \max\Big\{\big|Q[P_L]_{95\%,\text{top}} - Q[P_{\text{L,Ref}}]\big|,\ \big|Q[P_L]_{95\%,\text{bottom}} - Q[P_{\text{L,Ref}}]\big|\Big\}, \qquad (17)$$

where $Q[P_L]$ is computed according to Eq. (3) and the top and the bottom of the 95% confidence interval are obtained by $Q[P_L]_{95\%,\text{top}} := Q[P_L] + 1.96\, V_L^{\text{QMC}}$ and $Q[P_L]_{95\%,\text{bottom}} := Q[P_L] - 1.96\, V_L^{\text{QMC}}$ with V_L^{QMC} computed according to Eq. (4) for each method. As the reference value $Q[P_{L,\text{Ref}}]$ we take the average of our final approximations obtained by the interlaced Sobol'

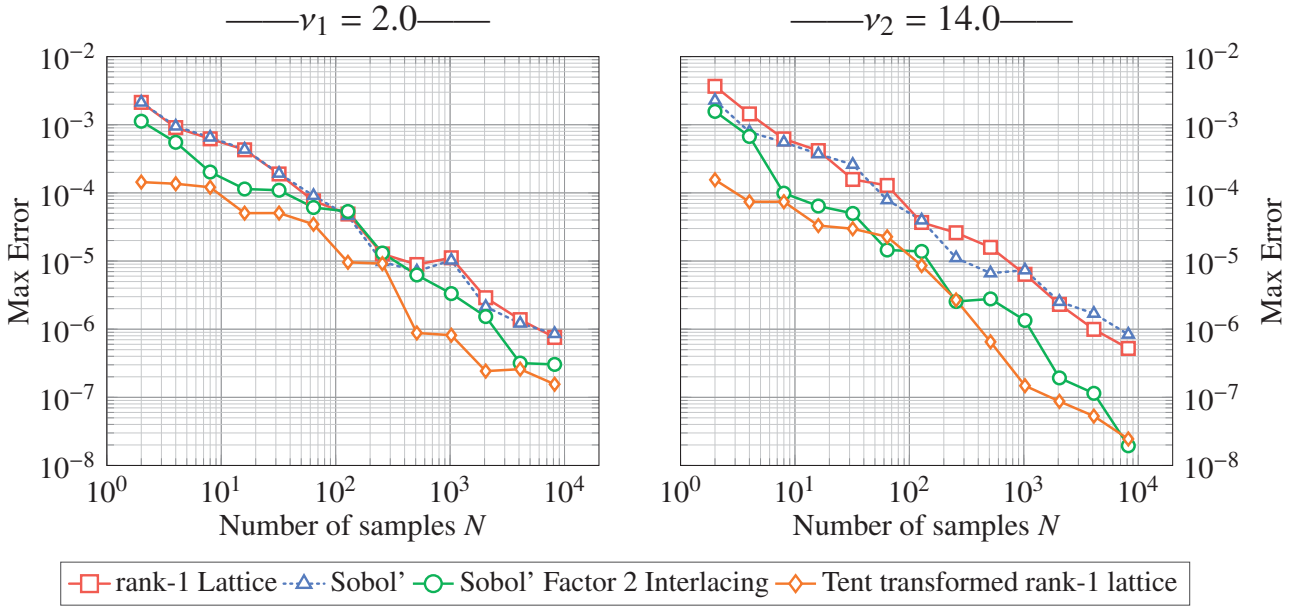

Figure 9: Absolute error on the expected value in function of the number of samples.

sequence and the tent-transformed lattice sequence, computed with 8192 samples and 8 shifts. The results from this simulation are chosen, because the root mean square error yields the lowest value, see Fig. 10, and we thus expect these values to be the most accurate. The numerical values for $Q[P_{L,\text{Ref}}]$ are 0.05415452 for $\nu_1 = 2.0$, and 0.05419605 for $\nu_2 = 14.0$.

From the graphs in Fig. 9 we can confirm that all estimators converge to the same value. For the case with $\nu_1 = 2.0$ all estimators converge approximately with $O(N^{-1})$ which is as expected due to the limited smoothness of the random field. For the case with $\nu_2 = 14.0$ we obtain $O(N^{-1})$ convergence for the plain QMC sequences, i.e., the lattice sequence and the Sobol' sequence, and we observe, as expected, an improved convergence of $O(N^{-1.5})$ for the tent-transformed lattice sequence and the interlaced Sobol' sequence.

4.2 Convergence of the Root Mean Square Error of the Estimator

In Fig. 10, we show the RMSE (root mean square error) of the estimators in function of the number of samples N, see Eq. (4). We observe that for a smoothness parameter $\nu_1 = 2.0$ in the Matérn kernel, the order of convergence for all approaches is $O(N^{-1})$ which is as expected. We do notice that in our example the tent-transformed lattice sequence has an RMSE which is approximatively a factor 10 lower than the standard lattice sequence. For the case of the higher smoothness $\nu_2 = 14.0$, we observe that using the tent-transformed lattice sequence and the interlaced Sobol' sequence achieves an order of convergence close to $O(N^{-1.5})$. For the vanilla versions of the lattice sequence and the Sobol' sequence, a classical quasi-Monte Carlo convergence of $O(N^{-1})$ is achieved, as expected. We conclude that the combination of a smooth random field, i.e., ν large, and higher order elements, in combination with a tent-transformed lattice sequence or an interlaced Sobol' sequence shows higher order convergence in our experiment.

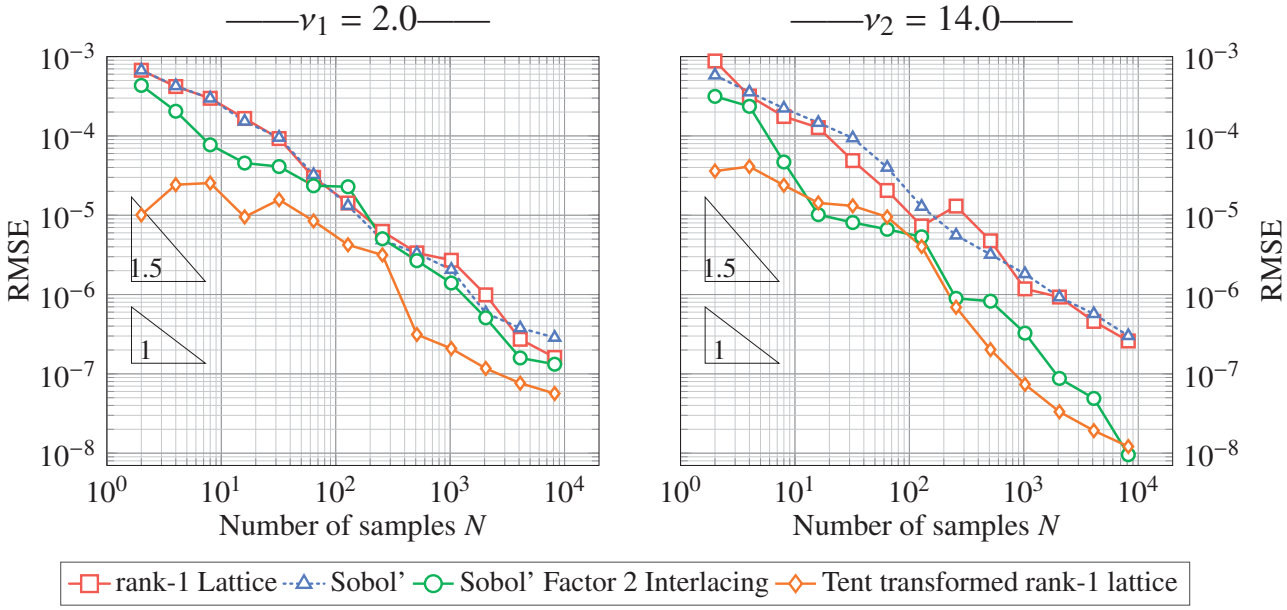

Figure 10: RMSE of the estimator in function of the number of samples.

5 CONCLUSIONS

In this work, we investigated two techniques to attain higher-order convergence by means of quasi-Monte Carlo methods in approximating expectations in a geotechnical slope stability problem. These techniques are tent-transformed lattice sequences and interlaced Sobol' sequences. We discussed how the deterministic quasi-Monte Carlo points for these point sets are obtained. We benchmarked these two techniques on a slope stability problem where the cohesion of the soil has a spatially varying uncertainty. The uncertainty is represented as a lognormal random field, and realizations of the random field are computed using a truncated KL expansion. We illustrated that, for a sufficiently smooth random field combined with tent-transformed rank-1 lattice points or with interlaced Sobol' points, and higher order elements, an order of convergence of $O(N^{-1.5})$ can be obtained. We compared the numerical results obtained by these two techniques against numerical results where a standard rank-1 lattice sequence, or a non-interlaced Sobol' sequence is used. For these latter two approaches, we observed the classical order of convergence of $O(N^{-1})$.

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] Fishman, G. S. *Monte Carlo: Concepts, algorithms and applications*. Springer-Verlag, New York (1996).

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

[3] Kuo, F. Y. and Nuyens, D. Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients: A survey of analysis and implementation. *Found. Comput. Math.* (2016)**16(6)**:1631–1696.

[4] Dick, J., Kuo, F. Y., and Sloan, I. H. High-dimensional integration: The quasi-Monte Carlo way. *Acta Num.* (2013)**22**:133–288.

[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 S-space Seoul National University Open 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] Blondeel, P., Robbe, P., François, S., Lombaert, G., and Vandewalle, S. Benchmarking the p-MLQMC Method on a Geotechnical Engineering Problem. In F. Chinesta, R. Abgrall, O. Allix, and M. Kaliske, editors, *Proceedings of the 14th World Congress on Computational Mechanics (WCCM) and ECCOMAS Congress 2020, Virtual Congress*. Published in Scipedia (2021) pages 1–12.

[8] Dick, J., Nuyens, D., and Pillichshammer, F. Lattice rules for nonperiodic smooth integrands. *Num. Math.* (2014)**126**:259–291.

[9] Dick, J., Kuo, F. Y., Le Gia, Q. T., Nuyens, D., and Schwab, C. Higher order QMC petrov–galerkin discretization for affine parametric operator equations with random field inputs. *SIAM J. Numer. Anal.* (2014)**52(6)**:2676–2702.

[10] 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.

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

[12] Lord, G. J., Powell, C. E., and Shardlow, T. *An Introduction to Computational Stochastic PDEs*. Cambridge Texts in Applied Mathematics. Cambridge University Press (2014).

[13] Brenner, C. E. and Bucher, C. A contribution to the sfe-based reliability assessment of nonlinear structures under dynamic loading. *Probab. Eng. Mech.* (1995)**10(4)**:265 – 273. ISSN 0266–8920.

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

[15] Blondeel, P. Digitalnets.jl (2020). Online `https://github.com/Philippe1123/DigitalNets.jl`, accessed on 05/11/2020.

[16] Nuyens, D. The "Magic Point Shop" of QMC point generators and generating vectors. Online `https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/`, accessed on 05/05/2020.

[17] Hickernell, F. J., Hong, H. S., L'Ecuyer, P., and Lemieux, C. Extensible lattice sequences for Quasi-Monte Carlo quadrature. *SIAM J. Sci. Comput.* (2000)**22(3)**:1117–1138.

[18] Cools, R., Kuo, F. Y., and Nuyens, D. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comput.* (2006)**28(6)**:2162–2188.

[19] Cools, R., Kuo, F. Y., Nuyens, D., and Suryanarayana, G. Tent-transformed lattice rules for integration and approximation of multivariate non-periodic functions. *J. Complexity* (2016)**36**:166–181.

[20] Hickernell, F. J. Obtaining o(n- 2+ ϵ) convergence for lattice quadrature rules. In K.-T. Fang, H. Niederreiter, and F. J. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer Berlin Heidelberg, Berlin, Heidelberg (2002) pages 274–289.

[21] Dick, J. Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands. *Ann. Stat.* (2011)**39(3)**:1372 – 1398.

[22] Nguyen, D. T. P. and Nuyens, D. MDFEM: Multivariate decomposition finite element method for elliptic PDEs with lognormal diffusion coefficients using higher-order QMC and FEM. *arXiv: 1904.13327 [math.NA]* (2020).

[23] Dick, J., Irrgeher, C., Leobacher, G., and Pillichshammer, F. On the optimal order of integration in hermite spaces with finite smoothness. *SIAM J. Numer. Anal.* (2018)**56(2)**:684–707.

[24] Nuyens, D. Lattice rule generating vectors (2007). Exod2_base2_m20_CKN at Online `https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/`, accessed on 12/04/2020.

[25] Joe, S. and Kuo, F. Sobol sequence generator. Online `https://web.maths.unsw.edu.au/~fkuo/sobol/`, accessed on 05/05/2020.

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