

General Introduction to Monte Carlo and Multi-level Monte Carlo Methods



Robin Schmidt, Matthias Voigt, Michele Pisaroni, Fabio Nobile,
Penelope Leyland, Jordi Pons-Prats and Gabriel Bugada

Introduction

With Monte Carlo (MC) methods, we identify a broad class of approaches that rely on the idea of approximating statistics of the response of a system by simulation through sampling. Because of its simplicity, robustness, and dimension independent convergence rate, MC methods can be used to characterize, in principle, any system that has a probabilistic interpretation. MC methods are often the easiest way, and sometimes the only feasible one, to solve a wide range high-dimensional problems.

Hereafter, we will denote random variables with capital letters and realizations of those with lower case letters. Vectors and matrices are shown in bold.

R. Schmidt (✉) · M. Voigt
Institute of Fluid Mechanics, Technische Universität Dresden, 01062 Dresden, Germany
e-mail: robin.schmidt@tu-dresden.de

M. Voigt
e-mail: matthias.voigt@tu-dresden.de

M. Pisaroni · F. Nobile · P. Leyland
Scientific Computing and Uncertainty Quantification, Ecole Polytechnique Fédérale de Lausanne,
1015 Lausanne, Switzerland
e-mail: michele.pisaroni@epfl.ch

F. Nobile
e-mail: fabio.nobile@epfl.ch

P. Leyland
e-mail: penelope.leyland@epfl.ch

J. Pons-Prats · G. Bugada
CIMNE, Aeronautical Group, 08034 Barcelona, Spain
e-mail: jpons@cimne.upc.edu

G. Bugada
Universitat Politècnica de Catalunya - BarcelonaTech, 08034 Barcelona, Spain
e-mail: bugeda@cimne.upc.edu

Suppose we are interested in computing the expected value $\mathbb{E}[Q]$ of a quantity of interest (QoI) $Q = Q(\mathbf{X})$ of a problem having some random elements \mathbf{X} . Assume we can generate N independent and identically distributed (i.i.d.) realizations $X^{(i)}$, $i = 1, \dots, N$ and for each of them evaluate the corresponding QOI $Q^{(i)} = Q(X^{(i)})$. Then the MC estimator for the expectation $\mathbb{E}[Q]$ of Q is defined as:

$$\mathbb{E}^{\text{MC}}[Q] := \frac{1}{N} \sum_{i=1}^N Q(X^{(i)}). \quad (1)$$

The simulation procedure that makes use of i.i.d. samples and the MC estimator introduced in Eq. (1) to approximate $\mathbb{E}[Q]$ is generally called Crude Monte Carlo (CMC).

Thanks to the Strong Law of Large Numbers, the approximation of $\mathbb{E}^{\text{MC}}[Q]$ converges with probability one (converges almost surely) to $\mathbb{E}[Q]$ as $N \rightarrow \infty$ as long as Q is integrable.

Moreover, $\mathbb{E}^{\text{MC}}[Q]$ is an unbiased estimator:

$$\mathbb{E}[\mathbb{E}^{\text{MC}}[Q]] = \mathbb{E}[Q] \quad (2)$$

meaning that the expected value of the MC estimator equals $\mathbb{E}[Q]$ for any N .

The rate of convergence of MC method can be described by the Central Limit Theorem (CLT) using the concept of convergence in distribution (weak convergence, size of the error with some probability). If the variance of Q , hereafter denoted with $\mathbb{V}\text{ar}[Q]$, is finite then the CLT asserts that

$$\sqrt{N} (\mathbb{E}^{\text{MC}}[Q] - \mathbb{E}[Q]) \implies \sqrt{\mathbb{V}\text{ar}[Q]} \mathcal{N}(0, 1) \quad (3)$$

as $N \rightarrow \infty$, where $\mathcal{N}(0, 1)$ is a normal random variable with mean zero and unit variance and \implies means convergence in distribution. From Eq. 3, for N large enough, we can derive confidence intervals for the estimator $\mathbb{E}^{\text{MC}}[Q]$ of $\mathbb{E}[Q]$:

$$|\mathbb{E}^{\text{MC}}[Q] - \mathbb{E}[Q]| \leq C_\alpha \frac{\sqrt{\mathbb{V}\text{ar}[Q]}}{\sqrt{N}} \quad (4)$$

with probability $1 - \alpha$, where C_α is a confidence parameter such that the cumulative distribution function Φ of a standard normal random variable $\Phi(C_\alpha) = 1 - \frac{\alpha}{2}$.

From Eq. (4), we can draw three crucial conclusions:

- the rate of convergence of MC is $O(N^{-1/2})$,
- for large N the error is normally distributed,
- the complexity of the computation depends solely on $\mathbb{V}\text{ar}[Q]$.

If an exact representation of the QoI is not accessible and we rely on a numerical approximation (e.g., a finite volume (FV) or finite element (FE) approximation in fluid dynamics computations) with a discretization parameter M (number of spatial

degrees of freedom), then Q is approximated by Q_M . The accuracy in estimating $\mathbb{E}[Q]$ by $\mathbb{E}^{\text{MC}}[Q_M]$ can be quantified by considering the mean square error (MSE) of the estimator:

$$e(\mathbb{E}^{\text{MC}}[Q_M])^2 := \mathbb{E}[(\mathbb{E}^{\text{MC}}[Q_M] - \mathbb{E}[Q])^2] = \underbrace{(\mathbb{E}[Q_M - Q])^2}_{(\text{B-}\mathbb{E}^{\text{MC}})} + \underbrace{\frac{\text{Var}[Q_M]}{N}}_{(\text{SE-}\mathbb{E}^{\text{MC}})}. \quad (5)$$

On the right-hand side, we can isolate two distinct contributions. The first term, the discretization error or bias (B- \mathbb{E}^{MC}), is the square error in mean between Q_M and Q and depends solely on the space discretization parameter M . The second term, the statistical error (SE- \mathbb{E}^{MC}), represents the variance of the estimator and decays inversely with the number of samples N .

The Crude Monte Carlo (CMC) approach is summarized in the algorithm below.

Algorithm 1: Crude Monte Carlo Algorithm

```

CMC( $N, M$ )
  for  $i = 0 : N$  do
    | Generate random samples:  $Q_M(X^{(i)})$ 
  compute  $\mathbb{E}^{\text{MC}}[Q_M] := \frac{1}{N} \sum_{i=1}^N Q_M(X^{(i)})$ 
    
```

CMC is a very elegant approach and has been proven to be robust and accurate for non-smooth problems, nevertheless its very slow convergence rate $O(N^{-1/2})$ prevents to achieve reasonably estimations in acceptable time for large-scale aerodynamic applications that require the solution of computational expensive CFD simulations.

Different strategies have been investigated in the last decades to accelerate MC methods. They are all based on the idea of reducing the ratio on the right-hand side of Eq. (4) $\frac{\sqrt{\text{Var}[Q]}}{\sqrt{N}}$. The two most prominent categories of approaches are:

- *Alternative Sampling techniques:* increase the denominator term to converge more rapidly by using deterministic (low-discrepancy) sequences, stratified sampling, or Latin Hypercube Sampling rather than pseudorandom numbers.
- *Variance Reduction techniques:* reduce the numerator term $\text{Var}[Q]$ by suitably modifying the quantity Q in a consistent way (i.e., without changing the expectation) as in the Multi-Level Monte Carlo approach.

These methodologies will be presented in the following sections and chapters.

Choice of Sampling Sequences

The generation of the $x^{(i)}$ samples with predefined probability distribution is a pivotal procedure in MC methods. In this section, we review different approaches used to generate pseudorandom and quasi-random numbers and methodologies required to prescribe appropriate correlations to random variables.

Pseudorandom Numbers

The simplest procedure is random sampling. However, true random numbers are the result of physical phenomenon as, for example, radioactive decay processes. Practical applications utilize pseudorandom numbers. Those results from pseudorandom number generators (PRNGs), also referred to as deterministic random number generators, are based on some reproducible mathematical formulation. Starting from a certain seed, the goal is to produce a sequence of uniform pseudorandom numbers in the interval $(0, 1)$ with statistical properties that are in very good agreement with those of a true sequence of i.i.d. random variables. The period length of the PRNG describes the number of random numbers until the sequence repeats itself. In general, a small period seems bad; however, a larger period is not necessarily better. A good PRNG has good performance in different criteria. A variety of theoretical and empirical tests, see, e.g., [1], can be conducted to decide whether a PRNG can be considered a good one.

The most common PRNG is based on recursive arithmetic, as, for example, linear congruential generators. Popular PRNG is the Mersenne Twister [2] or the combined multiple recursive generator according to [3].

Quasi-random Numbers

Quasi-random numbers are the result of low-discrepancy sequences. The resulting realizations are uniformly distributed in the interval $[0,1)$. They exhibit much more uniformity compared to random or pseudorandom numbers. Therefore, they increase the convergence rate if used within MC methods. In order to specify the application of low-discrepancy sequences in MC methods, the term Quasi-Monte Carlo (QMC) is used. The convergence rate of QMC is usually close to $O(N^{-1})$, which is higher compared to CMC, see Eq. (4).

Uniformity is measured by utilizing discrepancy which is defined as follows. Let B be a rectangle in the d -dimensional unit hypercube J with sides parallel to the coordinate axes and $m(B)$ its volume. The discrepancy of a set of N points in $[0, 1)^d$ is defined as

$$D_N = \sup_{B \in \mathcal{J}} \left| \frac{\text{Number of points in } B}{N} - m(B) \right|. \tag{6}$$

The most common low-discrepancy sequences are *Halton* and *Sobol* sequences. Both are based on the *van der Corput* sequence which is constructed by reversing the base- b representation of the sequence of natural numbers. For more details, concerning the construction of low-discrepancy sequences the interested reader is referred to [1, 4, 5].

Although low-discrepancy sequences possess high uniformity in low dimensions d (and large N), they can exhibit poor space-filling behavior for small N and large d . The d -dimensional *Halton* sequence, e.g., is constructed by pairing d one-dimensional sequences based on d different prime numbers (usually the first d primes). In the case of high dimensions, the base b must be large. The corresponding *van der Corput* sequences with large bases produce long linearly growing segments. If these are paired with each other, a strongly linear space filling of the unit square is obtained. Different techniques designated *leap* [6] and *scramble* [7] were created in order to overcome these problems.

Pseudorandom Variables with Non Uniform Distribution

In order to generate a random variable X from an arbitrary distribution the following two steps are involved.

1. Generation of uniform random numbers U_1, \dots, U_N with the PRNG.
2. Transformation of U_i according to its respective probability density function $f(X)$ or joint probability density function $f(\mathbf{X})$.

In the previous sections, we briefly presented how uniform pseudorandom numbers or uniform quasi-random numbers can be created. Here, we will describe two transformation methods in order to get a random variable X from such a uniform distributed random variable. The most notable transformation methods are the inverse transform method and the acceptance–rejection method.

In the inverse transform method, the random variable is calculated with the inverse of the CDF $F(X)$, see Algorithm 2.

Algorithm 2: Inverse-Transform Method

Generate U from $U(0, 1)$
 Return $X = F^{-1}(U)$

The acceptance–rejection method directly works with the PDF $f(X)$ of the considered random variable X . Moreover a further PDF $g(X)$ is needed, such that for some $c \geq 1$, $c g(X) \geq f(X)$ for all x . It is assumed that random numbers can be easily generated from $g(X)$. The resulting method is described in Algorithm 3.

Algorithm 3: Acceptance-Rejection Method

Generate X from $g(X)$

Generate U from $U(0, 1)$

If $U \leq f(X)/(c g(X))$ use X , otherwise reject X and repeat the process.

Stratification

If it is possible to divide a heterogeneous population into subpopulations each of which is homogeneous, a precise estimate of, e.g., the subpopulations mean can be obtained from a small sample. A combination of such estimates can deliver a precise estimate of the whole population with smaller number of realizations compared to CMC. This line of thought leads to *stratified sampling*. The idea behind stratified sampling is to divide the population of N units into m non-overlapping subpopulations, called strata. Each strata has N_i units with $i = 1, \dots, m$ and $\sum_i N_i = N$. A sample of size n_i with $i = 1, \dots, m$ and $\sum_i n_i = n$ is selected by some design within each stratum. In case of a random sample in each stratum, the term *stratified random sampling* is used. How to chose the strata depends on the particular problem. The population mean per unit $E^{\text{St}}[Q]$ can be estimated with

$$E^{\text{St}}[Q] = \frac{\sum_{i=1}^m N_i E_i[Q]}{N} = \sum_{i=1}^m W_i E_i[Q], \quad (7)$$

where W_i denotes the stratum weight. Only when the sampling fraction is the same in all strata which means e.g.

$$\frac{n_i}{n} = \frac{N_i}{N}, \quad (8)$$

the population mean corresponds to the sample mean. Such a stratified sampling is called *proportional*. If a predefined cost function is available, an optimal allocation of sample size can be achieved, e.g., in order to minimize the variance for $E^{\text{St}}[Q]$. A simple cost function can be a linear one where the cost is proportional to the size of the sample but varies from stratum to stratum.

The variance of an estimated mean of random sampling is denoted V_{ran} , of stratified sampling with proportional sample allocation V_{prop} and with optimum allocation for fixed n it is V_{opt} . It is shown in [8] that the following relation holds.

$$V_{opt} \leq V_{prop} \leq V_{ran} \tag{9}$$

Therefore, it can be argued that stratified sampling is always better compared to random sampling when enough information is available for its appropriate implementation. However, enough information is represented, e.g., by the frequency distribution of the result quantity, which is often only estimated prior to a probabilistic investigation. Therefore, the necessity of defining the strata is a major problem in stratified sampling. In case of one result quantity and if it is known a priori, for example, due to a reasonable number of measurements, a procedure to calculate the strata and number of strata is given in [8]. The determination of the strata becomes further complicated when many result quantities should be considered. The strata definition for one result quantity may be inappropriate for other quantities.

Correlation and Discrepancy Control

So far, only the marginal distributions of single variables were taken into account when creating random vectors. An $N \times d$ sample vector can be obtained by repeating d times the generation of one-dimensional random variables with N realizations. This naive approach can lead to undesired dependencies between the variables which must be avoided. On the other hand, specific interrelationships between the input variables might be explicitly desired for a variety of probabilistic simulations, for example, when treating measurements of a real system or in the context of sensitivity analyses, where correlation is of great importance and must be considered.

Relations between input variables can be represented by correlation, for example, using Pearson correlation coefficients (generally denoted with ρ) or Spearman rank correlation coefficient (denoted with r). According to Pearson, the correlation coefficient for two random variables X_i and X_j is defined as:

$$\rho_{ij} = \frac{\sum_{k=1}^N (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^N (x_{ki} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^N (x_{kj} - \bar{x}_j)^2}} \tag{10}$$

The rank correlation coefficient is calculated with the ranks of the data.

A correlation matrix \mathbf{C} of size $d \times d$ is obtained from a sample of the size $N \times d$. In the case of the correlation coefficient according to Pearson $C_{ij} = \rho_{ij}$. Furthermore, it is assumed that the desired correlation structure is known and predefined by a target correlation matrix \mathbf{T} .

There are two main groups of methodologies used to generate correlated random vectors with arbitrary given marginal distribution and correlations:

1. Methods that transform a correlated standard normal random variable into a target non-normal variable
2. Methods that optimize the rank correlation structure of a sample.

The popular Nataf model [9] belongs to the first group. A standard normal random vector \mathbf{Z} with a correlation matrix \mathbf{T}' is transformed component-wise into the desired vector \mathbf{X} with a correlation matrix \mathbf{T} . The marginal transformation is obtained by:

$$X_i = F_i^{-1}(\Phi(Z_i)), \tag{11}$$

where Φ is the standard normal CDF and $F_i(X_i)$ the CDF of X_i . The Nataf model approach assumes that \mathbf{Z} is jointly normal and uses the Pearson correlation coefficient (invariant under nonlinear strictly increasing transformations) as in Eq. (11). Thus, the relation $\mathbf{T}' \neq \mathbf{T}$ holds. In order to get the unknown matrix \mathbf{T}' , each element ρ'_{ij} must be computed by solving:

$$\rho_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{x_i - \mu_i}{\sigma_i} \frac{x_j - \mu_j}{\sigma_j} \varphi_2(z_i, z_j, \rho'_{ij}) dz_i dz_j, \tag{12}$$

where $\varphi_2(z_i, z_j, \rho'_{ij})$ designates the PDF of the bivariate standard normal distribution.

In order to avoid the elaborate solution of Eq. (12), empirical equations have been developed such that $\rho'_{ij} = f(\rho_{ij})$ can be computed, see e.g. [10].

If the matrix \mathbf{T}' is available, uncorrelated standard normal distributed random vectors can be transformed into correlated ones by means of Cholesky transformation. The Cholesky decomposition $\mathbf{T}' = \mathbf{L}\mathbf{L}^T$ provides the lower triangular matrix \mathbf{L} . The correlated random vectors are then obtained by applying $\mathbf{X}\mathbf{L}^T$.

The idea of converting uncorrelated random variables into correlated ones by orthogonal transformation is also the basis of methods belonging to the second group. One of these was developed by Iman and Conover [11] and is known as *Restricted Pairing*. The random vectors of individual random variables are generated according to their respective probability distribution without taking into account correlations. The restricted pairing technique uses the rank correlation coefficient. Compared to the Pearson correlation coefficient, the latter has the advantage of being invariant under monotonic transformations of the marginals. Algorithm 4 describes the procedure of Restricted Pairing.

Algorithm 4: Restricted Pairing

- Calculate \mathbf{C} from \mathbf{X}
 - Calculate lower triangular matrix \mathbf{Q} from $\mathbf{Q}\mathbf{Q}^T = \mathbf{C}$
 - \mathbf{P} from $\mathbf{P}\mathbf{P}^T = \mathbf{T}$
 - $\mathbf{S} = \mathbf{P}\mathbf{Q}^{-1}$
 - $\mathbf{R} = \mathbf{X}\mathbf{S}^T$
 - Change the order in \mathbf{X} in order to achieve the same ranks as in \mathbf{R}
-

The method proceeds from uncorrelated random variables. Practically, this is only possible to a limited extent. Therefore, the correlation of the input sample is taken into account by incorporating the rank correlation matrix \mathbf{C} which results from the available sample. In case of perfect correlation $r_{ij} = 1$ with $i \neq j$ in the input sample, the rows in each column of \mathbf{X} can be randomly shuffled.

An iteratively improved implementation of the Restricted Pairing technique has been presented in [12].

Besides the two aforementioned groups of methodologies, other approaches exist, as, e.g., the usage of Copulas to construct a multivariate random vector of dependent components.

A desired order within the sample can also be set up by solving a combinatorial optimization problem. The optimization is based on a scalar quantity which measures the deviation $\mathbf{E} = \mathbf{T} - \mathbf{A}$ between target correlation matrix and the actual correlation matrix \mathbf{A} . Vořechovský and Novák [13] described the deviation by *root mean square correlation* r_{rms} and minimized it by using Simulated Annealing and interchanging a pair of two realizations x_{ik} and x_{jk} .

$$r_{rms} = \sqrt{\frac{2 \sum_{i=1}^{d-1} \sum_{j=i+1}^d (E_{ij})^2}{d(d-1)}} \tag{13}$$

A suitable matrix norm can also be used to measure the maximum absolute correlation error:

$$r_{max} = \max_{1 \leq i \leq j \leq d} |E_{ij}| \tag{14}$$

If the correlation adjustment can be formulated as an optimization problem, a discrepancy improvement can be obtained with the same approach only by exchanging the objective. As an example, Liefvendahl and Stocki [14] used a genetic algorithm to solve the optimization problem.

The description of the space-filling proprieties of samples by means of a scalar quantity is possible with a multitude of criteria. An overview and an evaluation of existing criteria can be found in [15]. Beside all, the *centered L^2 discrepancy* [16] shows good performances for projections in 2D subspaces.

$$C^2 = \left(\frac{13}{12}\right)^d - \frac{2}{N} \sum_{i=1}^N \prod_{k=1}^d \left(1 + \frac{1}{2}|x_k^{(i)} - 0.5| - \frac{1}{2}|x_k^{(i)} - 0.5|^2\right) + \frac{1}{N^2} \sum_{i,j=1}^N \prod_{k=1}^d \left(1 + \frac{1}{2}|x_k^{(i)} - 0.5| - \frac{1}{2}|x_k^{(j)} - 0.5| - \frac{1}{2}|x_k^{(i)} - x_k^{(j)}|\right) \tag{15}$$

Latin Hypercube Sampling Method

Latin Hypercube Sampling (LHS) was first published by McKay et al. [17] and further developed by Iman and Conover [18]. The method can reduce the variance of an estimator compared to random sampling, which results in a reduction of the sample size while maintaining the statistical significance.

A mathematical proof of the variance reduction compared to CMC was given by McKay et al. [17] under the condition that the system behavior is monotonic in each of its inputs. Iman and Conover [18] show for an additive model with uniform inputs that the variance of an estimated mean converges with a factor of N^{-2} faster compared to CMC. Stein [19] demonstrated that the amount of variance reduction increases with the degree of additivity in the model response. An experimental comparison of LHS against CMC was carried out by Manteufel [20]. LHS estimates an unbiased mean value as well as the distribution function. The bias in the estimation of the variance is low and associated with a significantly lower sampling variability.

The idea behind LHS relates to stratified sampling. In LHS, only the marginal distributions are stratified in such a way that each random variable X is divided into N contiguous intervals of equal probability with respect to the corresponding CDF $F(X)$. For that purpose, the unit probability is divided into N intervals of identical probability $1/N$. These probability intervals are bounded by a lower ϕ_{k-1} and upper bound ϕ_k .

$$\phi_k = \frac{k}{N} \text{ with } k = 1, \dots, N \tag{16}$$

The calculation of the corresponding interval bounds ξ_k over the values of the random variable X can be performed by utilizing the inverse of the CDF $F(X)$

$$\xi_k = F^{-1}(\phi_k) . \tag{17}$$

In each probability interval one realization x_k must be selected. Therefore, $x_k \in (\xi_{k-1}, \xi_k)$ holds. Besides *random* LHS where each realization x_k is uniformly distributed in its respective interval, *mean* and *median* LHS exist. For those methods different ways of selecting the sample values from the probability intervals are applied. In case of *median* LHS each probability interval is selected by taking the following set of sampling probabilities.

$$\mathbf{p} = (p_1, p_2, \dots, p_k, \dots, p_N) \text{ with } p_k = \frac{k - 0.5}{N} \tag{18}$$

The samples are selected using the inverse transformation of the probabilities in \mathbf{p} .

$$x_k = F^{-1}(p_k) \tag{19}$$

The mean in each interval is selected for *mean* LHS. It makes a numerical integration of the PDF $f(X)$ necessary. The samples are selected using

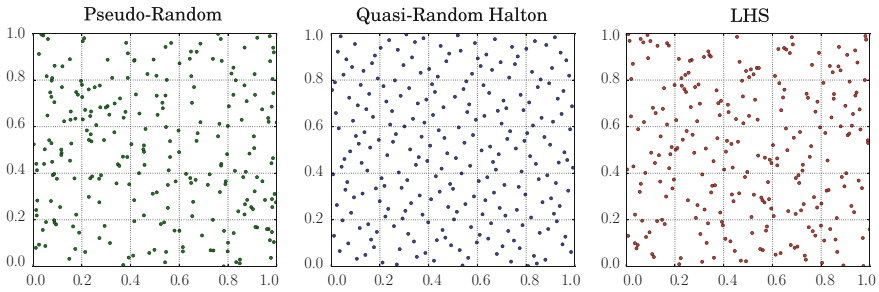


Fig. 1 Visual comparison of 250 pseudorandom, Halton quasi-random and *random* Latin Hypercube samples

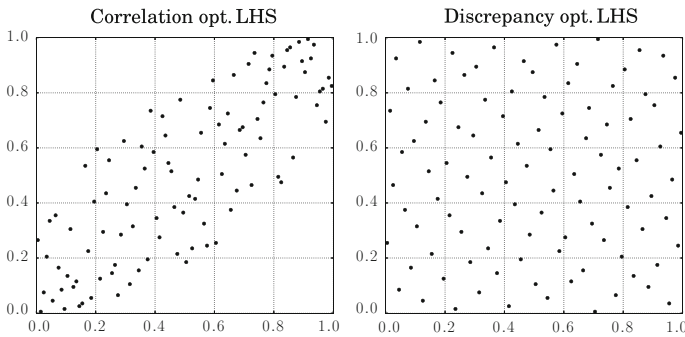


Fig. 2 Visual comparison of correlation optimized and discrepancy optimized *median* LHS; $N = 100$

$$x_k = \frac{\int_{\xi_{k-1}}^{\xi_k} x f(X) dx}{\int_{\xi_{k-1}}^{\xi_k} f(X) dx} . \tag{20}$$

Figure 1 shows a visual comparison of different sampling techniques for $N = 250$ realizations. Figure 2 extends the visual comparison and shows correlation and discrepancy control for a *median* LHS with $N = 100$.

Multi-level Monte Carlo

As previously stated, Crude Monte Carlo (CMC) sampling has a dimension independent convergence rate which is not affected by the presence of possible discontinuities in the parameter space. However, the CMC approach converges very slowly and is impractical in complex applications that require accurate solutions. The Multi-Level Monte Carlo (MLMC) method has been introduced by Heinrich [21, 22] in the context of parametric integration and extended by Giles [23] to approximate stochastic

differential equations (SDEs) in financial mathematics as a way to improve the efficiency of MC simulations. Applications to PDE models with random parameters can be found in [24–28].

Algorithm 5: Multi-Level Monte Carlo Algorithm

```

MLMC( $L, \{N_l\}_{l=0}^L$ )
  for  $l = 0 : L$  do
    for  $i = 0 : N_l$  do
      Generate random input:  $\omega^{(i,l)}$ 
       $Q_{M_l}^{(i)} \leftarrow \text{PROBLEM}_l(\omega^{(i,l)})$ 
       $Q_{M_{l-1}}^{(i)} \leftarrow \text{PROBLEM}_{l-1}(\omega^{(i,l)})$ 
       $Y_l^{(i)} = Q_{M_l} - Q_{M_{l-1}}$ 
    compute  $\mathbb{E}^{\text{MC}}[Y_l]$ 
  return  $\mathbb{E}^{\text{MLMC}}[Q_M] = \sum_{l=0}^L \mathbb{E}^{\text{MC}}[Y_l]$ 

```

The key idea of MLMC is to simultaneously draw MC samples on several approximations Q_{M_l} of Q built on a hierarchy of nested computational grids (with discretization parameters $M_0 < M_1 < \dots < M_L = M$) thanks to the linearity propriety of the expectation operator. Indeed the expectation of a QoI computed on the finest level can be written as a telescopic sum of the expectation of the QoI on the coarsest level plus a sum of correction terms adding the difference in expectation between evaluations on consecutive levels:

$$\mathbb{E}[Q_{M_L}] = \mathbb{E}[Q_{M_0}] + \sum_{l=1}^L \mathbb{E}[Q_{M_l} - Q_{M_{l-1}}] = \sum_{l=0}^L \mathbb{E}[Y_l] \tag{21}$$

with $Y_l = Q_{M_l} - Q_{M_{l-1}}$ and $Y_0 = Q_{M_0}$.

The MLMC estimator for $\mathbb{E}[Q]$ can be written as:

$$\mathbb{E}^{\text{MLMC}}[Q_M] := \sum_{l=0}^L \frac{1}{N_l} \sum_{i=1}^{N_l} Y_l(\omega^{(i,l)}) = \sum_{l=0}^L \mathbb{E}^{\text{MC}}[Q_{M_l} - Q_{M_{l-1}}], \tag{22}$$

where the same realization $\omega^{(i,l)}$ is used to compute the correction $Y_l(\omega^{(i,l)}) = Q_{M_l}(\omega^{(i,l)}) - Q_{M_{l-1}}(\omega^{(i,l)})$ on both levels whereas corrections on different levels should be sampled independently.

The accuracy in estimating $\mathbb{E}[Q]$ by $\mathbb{E}^{\text{MLMC}}[Q_M]$ can be quantified by considering the mean square error (MSE) of the estimator:

$$e(\mathbb{E}^{\text{MLMC}}[Q_M])^2 := \mathbb{E}[(\mathbb{E}^{\text{MLMC}}[Q_M] - \mathbb{E}[Q])^2] = \underbrace{(\mathbb{E}[Q_M - Q])^2}_{(\text{B-}\mathbb{E}^{\text{MLMC}})} + \underbrace{\sum_{l=0}^L \frac{\text{Var}[Y_l]}{N_l}}_{(\text{SE-}\mathbb{E}^{\text{MLMC}})}. \quad (23)$$

The standard MLMC algorithm is summarized in Algorithm 5. The notation PROBLEM_l denotes a general ‘black-box’ CFD solver that computes the QoI of the problem under investigation given a set of input values at the grid discretization level l . The description of the treatment of specific geometric or operating input random parameters will be provided in the following chapters.

References

1. Kroese, D.P., Taimre, T., Botev, Z.I.: Handbook of Monte Carlo Methods. Wiley, Inc. (2011)
2. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul. (TOMACS)* **8**(1), 3–30 (1998)
3. L’Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. *Oper. Res.* **47**(1), 159–164 (1999)
4. Niederreiter, H.: Random number generation and quasi-Monte Carlo methods. In: Society for Industrial and Applied Mathematics, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 63, Pa (1992)
5. Morokoff, W.J., Caflisch, R.E.: Quasi-Monte Carlo integration. *J. Comput. Phys.* **122**(2), 218–230 (1995)
6. Kocis, L., Whiten, W.J.: Computational investigations of low-discrepancy sequences. *ACM Trans. Math. Softw. (TOMS)* **23**(2), 266–294 (1997)
7. Hong, H.S., Hickernell, F.J.: Algorithm 823: implementing scrambled digital sequences. *ACM Trans. Math. Softw. (TOMS)* **29**(2), 95–109 (2003)
8. Cochran, W.G.: Sampling Techniques, 3rd edn. Wiley Series in Probability and Mathematical Statistics. Wiley, New York (1977)
9. Nataf, A.: Dtermination des distributions de probabilités dont les marges sont donnés. *Comptes Rendus de l’Académie des Sciences* **225**, 42–43 (1962)
10. Ditlevsen, O., Madsen, H.O.: Structural Reliability Methods. Wiley, Chichester, New York (1996)
11. Iman, R.L., Conover, W.J.: A distribution-free approach to inducing rank correlation among input variables. *Commun. Stat. Simul. Comput.* **11**(3), 311–334 (1982)
12. Dandekar, R.A., Cohen, M., Kirkendall, N.: Sensitive micro data protection using Latin hypercube sampling technique. In: Inference Control in Statistical Databases, Springer, pp. 117–125 (2002)
13. Vořechovský, M., Novák, D.: Correlation control in small-sample Monte Carlo type simulations I: a simulated annealing approach. *Probab. Eng. Mech.* **24**(3), 452–462 (2009)
14. Liefvendahl, M., Stocki, R.: A study on algorithms for optimization of Latin hypercubes. *J. Stat. Plan. Inference* **136**(9), 3231–3247 (2006)
15. Dambin, G., Couplet, M., Iooss, B.: Numerical studies of space-filling designs: optimization of Latin hypercube samples and subprojection properties. *J. Simul.* **7**(4), 276–289 (2013)
16. Hickernell, F.: A generalized discrepancy and quadrature error bound. *Math. Comput. Am. Math. Soc.* **67**(221), 299–322 (1998)
17. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **42**(1), 55–61 (1979)

18. Iman, R.L., Conover, W.: Small sample sensitivity analysis techniques for computer models with an application to risk assessment. *Commun. Stat. Theory Methods* **9**(17), 1749–1842 (1980)
19. Stein, M.: Large sample properties of simulations using Latin hypercube sampling. *Technometrics* **29**(2), 143–151 (1987)
20. Manteufel, R.D.: Evaluating the convergence of Latin hypercube sampling. In: 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (2000)
21. Heinrich, S.: Monte Carlo complexity of global solution of integral equations. *J. Complex.* **14**(2), 151–175 (1998)
22. Heinrich, S., Sindambiwe, E.: Monte Carlo complexity of parametric integration. *J. Complex.* **15**(3), 317–341 (1999)
23. Giles, M.B.: Multilevel Monte Carlo path simulation. *Oper. Res.* **56**(3), 607–617 (2008)
24. Barth, A., Lang, A., Schwab, C.: Multilevel Monte Carlo method for parabolic stochastic partial differential equations. *BIT Numer. Math.* **53**(1), 3–27 (2013)
25. Barth, A., Schwab, C., Zollinger, N.: Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numer. Math.* **119**(1), 123–161 (2011)
26. Charrier, J., Scheichl, R., Teckentrup, A.L.: Finite element error analysis of elliptic PDEs with random coefficients and its application to multi level Monte Carlo methods. *SIAM J. Numer. Anal.* **51**(1), 322–352 (2013)
27. Cliffe, K., Giles, M., Scheichl, R., Teckentrup, A.L.: Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.* **14**(1), 3–15 (2011)
28. Teckentrup, A., Scheichl, R., Giles, M., Ullmann, E.: Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numer. Math.* **125**(3), 569–600 (2013)