Probabilistic Engineering Mechanics 24 (2009) 452–462

Contents lists available at ScienceDirect

Probabilistic Engineering Mechanics

journal homepage: www.elsevier.com/locate/probengmech

Correlation control in small-sample Monte Carlo type simulations I: A simulated annealing approach

M. Vořechovský *, D. Novák

Institute of Structural Mechanics, Faculty of Civil Engineering, Brno University of Technology, Veveří 95, 602 00 Brno, Czech Republic

ARTICLE INFO

Article history:
Received 2 November 2008
Accepted 21 January 2009
Available online 1 February 2009

Keywords:
Monte Carlo simulation
Covariances
Latin Hypercube Sampling
Statistical correlation
Combinatorial optimization
Simulated Annealing

ABSTRACT

The objective of this paper is to propose an effective procedure for sampling from a multivariate population within the framework of Monte Carlo simulations. The typical application of the proposed approach involves a computer-based model, featuring random variables, in which it is impossible to find a way (closed form or numerical) to carry out the necessary transformation of the variables, and where simulation is expensive in terms of computing resources and time. Other applications of the proposed method can be seen in random field simulations, optimum learning sets for neural networks and response surfaces, and in the design of experiments.

The paper presents a technique for efficient Monte Carlo type simulation of samples of random vectors with prescribed marginals and a correlation structure. It is shown that if the technique is applied for small-sample simulation with a variance reduction technique called Latin Hypercube Sampling, the outcome is a set of samples that match user-defined marginals and covariances. Such a sample is expected to lead to stable estimates of the statistics of the analyzed function, with low variability. The method is very flexible in terms of the allowable combination of marginal distributions and correlation structures. The efficiency of the technique is documented using simple numerical examples. The advantages of the presented method are its simplicity and clarity; the method has proven itself to be simple to use, fast, robust and efficient, especially for very small sample sizes.

1. Introduction

The aim of statistical and reliability analyses of any computational problem which can be numerically simulated is mainly the estimation of statistical parameters of response variables and/or theoretical failure probability. Pure Monte Carlo simulation cannot be applied to time-consuming problems, as it requires a large number of simulations (repetitive calculation of responses). A small number of simulations can be used to gain an acceptable level of accuracy for the statistical characteristics of the response using the stratified sampling technique Latin Hypercube Sampling (LHS) first developed by Conover [1] and later elaborated mainly in [2–4]. Stratified sampling is revised here, and this is followed by the main problem of treating/imposing statistical correlations among input basic random variables.

It is known that the output response variables of some systems, represented by their response functions, are sensitive to changes in correlations among the input variables. Therefore, it is essential to precisely capture the input correlations in the simulated values. Thus, Monte Carlo type simulation approaches require sampling of correlated data from Gaussian and frequently also non-Gaussian distributions.

Other than the multivariate normal distribution, few random-vector models are tractable and general, though many multivariate distributions are well documented [5]. The available techniques for simulation of (generally non-Gaussian) correlated vectors are listed in Section 2.2.

In the present paper, the task of correlation control in sampling is treated as a combinatorial optimization problem. Examples of such a class of problems and approaches can be found in graph theory and integer programming, such as the traveling salesman problem, the network design problem, the problem of optimal chip placement in computer processors, the knapsack problem or the decision tree design problem [6]. This class of problems has received a great deal of attention in the literature due to the large number of practical problems that it includes. In the 1980s Kirkpatrics et al. [7] and Černý [8,9] argued that all combinatorial problems possess a common structure, namely, they are all multivariate discrete systems with many degrees of freedom. An analogy is made between the behavior of large physical systems and that of combinatorial problems, with the result that one could apply results from classical statistical mechanics to combinatorial optimization. In this paper, an analogy between the statistical mechanics of large multivariate physical systems

* Corresponding author. Tel.: +420 5 4114 7370; fax: +420 5 41240994.
E-mail addresses: vorechovsky.m@fce.vutbr.cz (M. Vořechovský), novak.d@fce.vutbr.cz (D. Novák).
URL: http://www.fce.vutbr.cz/STM/vorechovsky.m/ (M. Vořechovský).

0266-8920/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.probengmech.2009.01.004

Notations

N_{sim}	number of simulations;
N_{var}	number of variables (marginals);
N_{trials}	number of trials at a constant temperature;
$\boldsymbol{A}$	actual correlation matrix (estimated from sample) with entries $A_{i,j}$;
$\boldsymbol{T}$	target correlation matrix (user-defined) with entries $T_{i,j}$;
$\boldsymbol{W}$	weight correlation matrix (user-defined) with entries $W_{i,j}$;
$\boldsymbol{E}$	error correlation matrix (computed);
$\hat{\boldsymbol{E}}$	weighted error correlation matrix;
E_{parent}, $[E_{off}, E_{best}]$	cost function (error) of the parent [offspring, currently best] configuration (state);
C_{parent}, $[C_{off}, C_{best}]$	parent [offspring, current best] configuration (state);
$\boldsymbol{r}$	realization of rank matrix corresponding to $\boldsymbol{x}$;
ΔE	error difference before and after random change in configuration;
t_0, t_{min}	maximum and minimum temperatures in Simulated Annealing;
$\boldsymbol{I}$	unit matrix;
$\boldsymbol{X}$	vector of random variables;
$\boldsymbol{x}$	realization of the vector $\boldsymbol{X}$ ($N_{sim} \times N_{var}$ matrix);
$\boldsymbol{U}$ $[\boldsymbol{Z}]$	vector of uncorrelated [correlated] standardized Gaussian variables;
ρ_{rms}	weighted root mean square error (correlation);
ρ_{max}	weighted maximum absolute error (correlation);
$\pi_i(j)$	random permutation of rank number j for ith variable;
$\xi_{i,j}$	upper bound for jth realization $x_{i,j}$ in LHS;
Φ $[\Lambda]$	eigenvectors [eigenvalues] of $\boldsymbol{T}^Z$;

and combinatorial optimization is presented and used to develop a strategy for the optimal ordering of samples to control the correlation structure. The problem of optimal sample ordering is solved by the so-called Simulated Annealing method using a Monte Carlo procedure similar to the one developed by Metropolis et al. [10]. The present paper is an extended version of conference papers published by the authors in 2002 [11–14]. A paper describing a nearly identical technique appeared in the literature two years later [15].

The paper is organized as follows. After a review of small-sample simulation techniques in Section 2 we proceed to univariate sampling (Section 2.1) and a formulation of the problem of correlation control (Section 2.2). Next, we turn our attention to general combinatorial optimization and the Simulated Annealing technique in Section 3. The application of the method to correlation control is thoroughly described in Section 3.2 and numerical examples follow.

2. Review of small-sample Monte Carlo LHS

Let us consider the deterministic function $Y = g(\boldsymbol{X})$ (e.g. a computational model), where $Y \in \mathbb{R}$, $\boldsymbol{X} \in \mathbb{R}^{N_{var}}$ is a random vector of N_{var} marginals (input random variables describing uncertainties) and $g(\cdot)$ can be expensive to evaluate. The information on the random vector is limited to marginal probability distributions and the target correlation matrix $\boldsymbol{T}$:

$$\boldsymbol{T} = \begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_{N_{var}} \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & \dots & x_{N_{var}} \end{array} \\ \begin{pmatrix} 1 & T_{1,2} & \cdots & T_{1,N_{var}} \\ \vdots & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \text{sym.} & \cdots & \cdots & 1 \end{pmatrix} \end{array}. \tag{1}$$

The task is to perform statistical, sensitivity and possibly reliability analyses of Y. Suppose the analytical solution of the transformation of input variables to Y is not possible. The most prevalent technique for solving the task is Monte Carlo simulation (MCS). MCS is popular for its simplicity and transparency and is also often used in the benchmarking of other (advanced) simulation techniques. The procedure is to draw N_{sim} realizations of $\boldsymbol{X}$ and compute the same number of output realizations of Y using the model $g(\cdot)$. Because $g(\cdot)$ can be expensive to compute it pays to use a more advanced sampling scheme. A good choice is one of the "variance reduction techniques" called Latin Hypercube Sampling (LHS).

LHS was first suggested by W.J. Conover, whose work was motivated by the time-consuming nature of simulations connected with the safety of nuclear power plants. Conover's original unpublished report [1] is reproduced as Appendix A of [16] together with a description of the evolution of Latin Hypercube Sampling as an unpublished text by R.L. Iman (1980). LHS was formally published for the first time in conjunction of Conover's colleagues in 1979 [2].

LHS is a special type of Monte Carlo numerical simulation which uses the stratification of the theoretical probability distribution functions of input random variables. For statistical analysis (aiming at the estimation of low statistical moments of response) it requires a relatively small number of simulations (from tens to hundreds)—repetitive evaluations of the response function $g(.)$. LHS strategy has been used by many authors in different fields of engineering and with both simple and very complicated computational models. Its role in reliability engineering is described in [17]. LHS is suitable for statistical and sensitivity calculations. There is also the possibility of using it for probabilistic assessment within the framework of curve fitting. LHS has also been successfully combined with the Importance Sampling technique [18] to minimize the variance of estimates of failure probability by sampling importance density around the design point. Optimal coverage of a space with many variables with a minimum number of samples is also an issue in the design of experiments, and LHS, along with related sampling techniques, has its place in that field. In the next two subsections, we will discuss (i) univariate sample selection in LHS and (ii) the correlation estimation of samples.

Although the text in following chapters will deal with LHS, the methods described in this work can readily be generalized to any Monte Carlo sampling type method.

2.1. Univariate sampling

Latin Hypercube Sampling is a form of simultaneous stratification for all N_{var} variables of the unit cube $[0;\ 1]^{N_{var}}$. There are several alternative forms of LHS. In the centered version (called lattice sampling by Patterson [19]) the jth realization of ith random variable X_i ($i = 1, \dots, N_{var}$) is denoted $x_{i,j}$ and generated as:

$$x_{i,j} = F_i^{-1}\left(\frac{\pi_i(j) - 0.5}{N_{sim}}\right), \tag{2}$$

where $\pi_i(1), \dots, \pi_i(N_{sim})$ is a random permutation of $1, \dots, N_{sim}$; F_i^{-1} is the inverse of the cumulative distribution function of this random variable and N_{sim} is the number of simulations, i.e. the

Table 1
Sampling scheme for N_{sim} evaluations of $g(\boldsymbol{X})$.

Var	Sim:			
	1	2	...	N_{sim}
X_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,N_{\text{sim}}}$
X_2	$x_{2,1}$	$x_{2,2}$	...	...
...	...	...	...	...
$X_{N_{\text{var}}}$	$x_{N_{\text{var}},1}$	...	...	$x_{N_{\text{var}},N_{\text{sim}}}$

number of realizations for each random variable. If F_i is continuous, then each of the N_{sim} equiprobable subintervals $j = 1, \ldots, N_{\text{sim}}$ for X_i is represented by one value $x_{i,j}$. McKay et al. [2] showed that such a sample selection reduces the sampling variance of the statistics of $g(\boldsymbol{X})$ when $g(\cdot)$ is monotone in each of the inputs. In the unbiased version, from McKay et al. [2], the Latin Hypercube Sample is generated by replacing the number 0.5 in Eq. (2) by U_j^i, where U_j^i is a uniformly distributed random variable over the interval [0, 1), independent of the permutations π_i (this sampling selection is called LHS-random from here on). The centered version in Eq. (2) was originated by Patterson [19] in the setting up of agricultural experiments, whereas the version by McKay et al. [2] was motivated by computer experiments.

The midpoint rule (Eq. (2)) is very often used for various problems in the literature (we denote the technique as LHS-median). However, one can criticize such a reduction of the sample selection to the midpoints within intervals (interval medians). This objection deals mainly with samples of the tails of PDF, which mostly influence the sample variance, skewness and kurtosis. This elementary simple approach has already been overcome by the sampling of interval mean values, e.g. [20,21]:

$$x_{i,j} = \frac{\int_{\xi_{i,j-1}}^{\xi_{i,j}} x\, f_i(x)\, \mathrm{d}x}{\int_{\xi_{i,j-1}}^{\xi_{i,j}} f_i(x)\, \mathrm{d}x} = N_{\text{sim}} \int_{\xi_{i,j-1}}^{\xi_{i,j}} x\, f_i(x)\, \mathrm{d}x \tag{3}$$

where f_i is the probability density function (PDF) of variable X_i and the integration limits (right bounds for jth realizations) are $\xi_{i,j} = F_i^{-1}(j/N_{\text{sim}}), j = 1, \ldots, N_{\text{sim}}$, see Fig. 1. By using this scheme (LHS-mean), samples represent one-dimensional marginal PDF better in terms of the distance of the point estimators from the exact statistics. In particular, the mean value is achieved exactly (the analytical expression preserves the mean) and estimated variance is much closer to that of the target. For some PDFs (including Gaussian, Exponential, Laplace, Rayleigh, Logistic, Pareto, and others) the integral Eq. (3) can be solved analytically. In the case that solution of the primitive is impossible or difficult, it is necessary to use an additional effort: numerical solution of the integral. However, such an increase in computational effort is definitely worthwhile especially when N_{sim} is very small. Samples selected by both Eqs. (2) and (3) are almost identical except for the values in the tails of PDFs (this will be shown also using numerical examples, Fig. 7). Therefore one can use the more advanced scheme Eq. (3) only for the tails, considering the fact that tail samples mostly influence the estimated variance of the sample set. Generally, in all three cases, regularity of sampling (the range of distribution function is stratified) ensures good sampling and consequently good estimation of statistical parameters of response using a small number of simulations.

Stratification with proportional allocation never increases variance compared to crude Monte Carlo sampling, and can reduce it. Indeed, Stein [22] has shown that LHS reduces the variance compared to simple random sampling (crude Monte Carlo). The amount of variance reduction increases with the degree of additivity in the random quantities on which the function $g(\boldsymbol{X})$ depends.

The sampling scheme of any Monte Carlo type technique is represented by Table 1, where simulation numbers are in columns

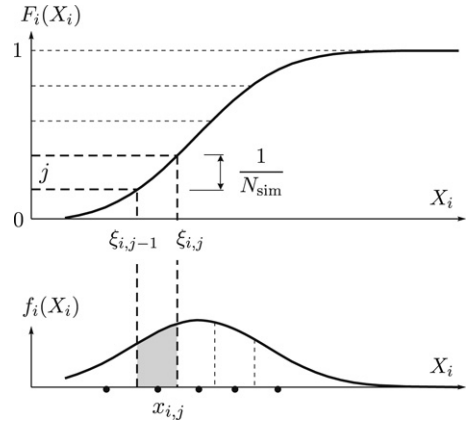

Fig. 1. Samples as the probabilistic means of intervals: LHS-mean scheme.

while rows are related to random variables (N_{var} is the number of input variables). Note that Table 1 can be obtained either by sampling from a parametric distribution or from a set of raw data, bounded or unbounded, continuous or discrete, empirical histogram, etc. The only requirement is that the sample size N_{sim} is identical for all sampled variables. From here on, we assume that the values representing each variable from Table 1 have already been selected, and that we want to pay attention to the correlation structure among the variables.

2.2. Statistical correlation of samples

There are generally two problems related to LHS concerning statistical correlation: First, during sampling an undesired correlation can be introduced between random variables (rows in Table 1). For example, instead of a correlation coefficient of zero for uncorrelated random variables, an undesired (spurious) correlation, e.g. 0.4 can be randomly generated. It can be significant, especially in the case of a very small number of simulations (tens), where the number of interval combinations is rather limited. The second task is to determine how to introduce prescribed statistical correlations between pairs of random variables defined by the target correlation matrix $\boldsymbol{C}_{\text{x}} \equiv \boldsymbol{T}$. The samples in each row of Table 1 should be rearranged in such a way as to fulfill the following two requirements: to diminish spurious random correlation, and to introduce the prescribed correlation given by $\boldsymbol{T}$.

Two widely used possibilities exist for the point estimation of correlation between two variables: the Pearson correlation coefficient (PCC) and the Spearman rank order correlation coefficient (SRCC). The PCC takes values between -1 and 1 and provides a measure of the strength of the linear relationship between two variables. For samples in the form of rows as presented in Table 1, the sample PCC A_{ij} between two variables, say x_i and x_j, is defined by

$$A_{ij} = \frac{\sum_{k=1}^{N_{\text{sim}}} \left(x_{i,k} - \overline{x_i}\right)\left(x_{j,k} - \overline{x_j}\right)}{\sqrt{\sum_{k=1}^{N_{\text{sim}}} \left(x_{i,k} - \overline{x_i}\right)^2 \sum_{k=1}^{N_{\text{sim}}} \left(x_{j,k} - \overline{x_j}\right)^2}},$$

$$\overline{x_i} = \frac{1}{N_{\text{sim}}} \sum_{k=1}^{N_{\text{sim}}} x_{i,k} \qquad \overline{x_j} = \frac{1}{N_{\text{sim}}} \sum_{k=1}^{N_{\text{sim}}} x_{j,k}. \tag{4}$$

The SRCC is defined similarly to the PCC but with rank-transformed data. Let us define a matrix $\boldsymbol{r}$ in which each row/column is filled with rank numbers corresponding to a matrix $\boldsymbol{x}$. Specifically, the smallest value $x_{i,j}$ of a variable i is given a rank $r_{i,j} = 1$; the next

largest value is given a rank of 2; and so on up to the largest value, which is given a rank equal to sample size N_{sim}. In the event of ties, average ranks are assigned. Note that when LHS is applied to continuous parametric distributions no ties can occur in the generated data. The SRCC is then calculated in the same manner as the PCC except in the use of rank-transformed data. Specifically, $x_{i,j}$ must be replaced by rank number $r_{i,j}$ in Eq. (4). In the formula $\overline{x_i}$ simplifies to the average rank of $(N_{\text{sim}}+1)/2$.

Note that there are more possible choices for the correlation estimator in $\boldsymbol{A}$, e.g. Kendall's *tau*.

2.3. Available sampling techniques for correlated random vectors

There are special analytical results (e.g. for multivariate Gaussian distribution) and algorithms available for simulating samples of multivariate random variables, e.g. Parrish's method [23] for sampling from the multivariate Pearson family of distributions with known product moments up to the fourth order. There are also works available in the literature on the generation of random vectors based on a sample of the target vector $\boldsymbol{X}$ (data-based algorithms, see e.g. an algorithm in [24] that may also yield a simple bootstrap).

The majority of available methods developed so far for the simulation of correlated random vectors with arbitrary given marginals and correlations within the context of Monte Carlo sampling can be divided into two fundamental groups: (i) methods that transform an underlying correlated Gaussian vector $\boldsymbol{Z}$ into the target non-Gaussian vector $\boldsymbol{X}$ and, (ii) methods that perform rank optimizations of samples generated without taking account of intercorrelations. We proceed with a revision of the available methods now.

Multivariate extensions of the Fleishman power method [25] for simulation of a random variable X belong to the first group. Generation of each random variable X_i from the target vector $\boldsymbol{X}$ is based on the knowledge of the first four moments (i.e. specified means, variances, skewnesses and kurtoses) and uses the polynomial transformation of an underlying Gaussian random variable Z_i: $X_i = a + bZ_i + cZ_i^2 + dZ_i^3$. Given the target statistical moments, one has to solve a nonlinear system of equations for the coefficients a, b, c, d. This expression of the target non-Gaussian random variable has been extended to the simulation of random vectors $\boldsymbol{X}$ with Pearson's correlation coefficients by Vale and Maurelli [26]. In their method, a Gaussian random vector $\boldsymbol{Z}$ with an *intermediate correlation matrix* $\boldsymbol{C}_z$ is first generated via principal component factorization of $\boldsymbol{C}_z$ (or any other factorization) and then each target variable X_i is obtained by the aforementioned Fleishman's polynomial transformation. The method has a serious drawback: the two processes interact, i.e. $\boldsymbol{C}_x \neq \boldsymbol{C}_z$, and there is a need to find proper intermediate correlations $\boldsymbol{C}_z$ depending on the coefficients a, b, c, d and the desired intercorrelations $\boldsymbol{C}_x$. A variation of this approach avoiding the factorization procedure has been published in [27]. Another variation is the approach in [28] where it is proposed that the Cholesky decomposition be combined with the cubic transformation in heuristic iterative modifications of the distribution of $\boldsymbol{Z}$.

The most widespread representative of group (i) for simulation from random vectors with specified marginals and correlations is the NORTA (NORmal To Anything) model, often also called the Nataf model [29]. In this model the underlying Gaussian random vector $\boldsymbol{Z}$ with the intermediate correlation matrix $\boldsymbol{C}_z$ is transformed into the desired vector $\boldsymbol{X}$ component by component via the equality $F_i(X_i) = \Phi(Z_i)$, $i = 1, \ldots, N_{\text{var}}$, i.e.:

$$X_i = F_i^{-1}\left[\Phi(Z_i)\right] \Leftrightarrow Z_i = \Phi^{-1}\left[F_i(X_i)\right] \tag{5}$$

where Φ is the standard normal cumulative distribution function (CDF) and F_i is the ith marginal CDF. The approach was founded by Mardia [30] who described the transformation of bivariate normal random vectors. Li and Hammond [31] extended the concept to random vectors of arbitrary finite dimension with continuous non-Gaussian marginals. The continuity of variables X_i is helpful because $F(X)$ then has a uniform distribution on (0,1), and so one can obtain a normally distributed random variable using the second part of Eq. (5). The model has been developed to match the desired linear Pearson correlation $\boldsymbol{C}_x$. Linear correlation has a serious deficiency in that it is not invariant under nonlinear strictly increasing transformations such as the one in Eq. (5) and thus $\boldsymbol{C}_x \neq \boldsymbol{C}_z$. In the model, every entry $\rho_{z_i z_j}$ of $\boldsymbol{C}_z$ must be computed by inversely solving the equality for the desired correlations $\rho_{x_i x_j}$:

$$\rho_{x_i x_j} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{(x_i-\mu_i)}{\sigma_{z_i}} \frac{\left(x_j-\mu_j\right)}{\sigma_{z_j}} \quad \varphi_2\left(z_i, z_j, \rho_{z_i z_j}\right) \mathrm{d}z_i \mathrm{d}z_j \tag{6}$$

where $\varphi_2\left(z_i, z_j, \rho_{z_i z_j}\right)$ is the bivariate standard Gaussian PDF with the correlation $\rho_{z_i z_j}$. In total one must solve $N_{\text{var}}\left(N_{\text{var}}-1\right)/2$ inversions of the double integral. Liu and Der Kiureghian [32] and later Cario and Nelson [33] have stated several important properties of the transformation in Eq. (6). Liu and Der Kiureghian have [32] found simple regression formulas that can assist in approximating the double integral associated with transformation for several of the most frequent distributions.

There are two problems associated with the intermediate correlations $\boldsymbol{C}_z$ of the underlying Gaussian vector obtained as a solution of Eq. (6). First, for some combinations of marginals of vector $\boldsymbol{X}$ and target correlations $\rho_{x_i x_j}$, a feasible solution may not exist i.e., some correlations $\rho_{z_i z_j}$ lie outside the interval -1, 1 and thus such a vector $\boldsymbol{X}$ cannot be represented by the underlying Gaussian vector $\boldsymbol{Z}$. As will be shown in the companion part III [34], our algorithm is able to construct such vectors $\boldsymbol{X}$ and therefore we conclude that NORTA cannot be used for some vectors even if they exist. The second problem is that even if all entries of the correlation matrix $\boldsymbol{C}_z$ can be solved, the matrix as a whole may become negative definite and thus NORTA-infeasible. The reason is that $|\rho_{x_i x_j}| \leq |\rho_{z_i z_j}|$ (see Eq. (6)), and therefore the positive definite matrix $\boldsymbol{C}_x$ may yield the negative definite $\boldsymbol{C}_z$. The probability of receiving the NORTA-infeasible correlation matrix $\boldsymbol{C}_z$ increases fast with the dimension of the problem N_{var}, see [35]. Usually in such cases, however, the matrix $\boldsymbol{C}_z$ is "close" to a positive semidefinite matrix, and therefore it is possible to find the nearest feasible matrix in order to use the model. Ghosh and Henderson [35] proposed fixing the matrix via a semidefinite programming approach while Lurie and Goldberg [36] proposed finding the nearest positive semidefinite matrix by adjusting the lower triangular Cholesky matrix $\boldsymbol{L}$ via Gauss–Newton iterations to make $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ close to $\boldsymbol{C}_z$. Another approach may be to perform spectral decomposition of $\boldsymbol{C}_z$ and change negative eigenvalues into small positive ones (and decrease the rest to keep the matrix trace). As will become clear later, our proposed approach automatically performs the search for the nearest positive definite matrix as well.

Biller and Nelson [37] proposed a combination of the NORTA technique together with the Johnson translation system of distributions [38] for cases when modeling data with unknown PDFs. The Johnson system is reasonably flexible (it can match any of the feasible first four moments) and its application allows one to avoid evaluation of $\Phi(\cdot)$ in the numerical integration in Eq. (6). A very recent paper by Headrick and Mugdadi [39] proposes connection of the NORTA approach with the generalized lambda distribution as a tool for modeling partially defined random vectors. An alternative approach named DIRTA (DIRichlet To Anything) with a similar basis to the NORTA approach has recently been published by Stanhope in [40].

We note that sampling of the underlying $\boldsymbol{C}_z$-correlated Gaussian vector $\boldsymbol{Z}$ is a well-documented and widely used task (see e.g. [41]).

A direct approach is to generate uncorrelated standard Gaussian vector $\boldsymbol{Y}$ and transform it via Cholesky linear transformation $\boldsymbol{YL}^{\mathrm{T}}$ $\left(\boldsymbol{C}_{\mathrm{z}} = \boldsymbol{LL}^{\mathrm{T}}\right)$ or via spectral decomposition of the correlation matrix.

This idea of transforming independent random variables into correlated ones using orthogonal transformation has been exploited in what is probably the most widespread technique for correlation control in simulations, developed by Iman and Conover [42] (later published and numerically verified by Florian [43] under the name "Updated Latin Hypercube Sampling" (ULHS)). They used the Spearman rank order correlation coefficient, which is invariant under monotone transformations of marginals, and therefore, they removed the problems with the intermediate Pearson's correlation matrix present in the NORTA technique. The method changes the sample ordering of $\boldsymbol{X}$ while leaving the representative values sampled directly from non-Gaussian marginals of each variable untouched. The ordering of samples of $\boldsymbol{X}$ is equal to the ordering of van der Waerden scores set for each variable as $a(i) = \Phi^{-1}(i/(i+1))$ for $i = 1, \ldots, N_{\mathrm{sim}}$, which initially form a matrix $\boldsymbol{A}$ of independent variables and are transformed into correlated ones via $\boldsymbol{AL}^{\mathrm{T}}$.

Iman and Conover used the technique in connection with LHS, the efficiency of which was first shown in the work of McKay et al. [2], but only for uncorrelated random variables. Iman and Conover [42] perturbed Latin Hypercube Samples in a way that reduces off-diagonal correlation—they diminished undesired random correlation. The technique is based on iterative updating of the sampling matrix, and Cholesky decomposition of the actual current correlation matrix of vector $\boldsymbol{X}$. This is simply an application of the method published by Scheuer and Stoller for normal vectors [44]. The second step (the procedure of correlating independent variates) can be performed only once; there is no way to iterate it and to improve the result, which is a disadvantage of the technique.

The above described iterative technique formulated by Iman and Conover [42] can result in a very low correlation coefficient if generating uncorrelated random variables. However, Huntington and Lyrintzis [21] have found that the approach tends to converge to an ordering which still gives significant correlation errors between some variables. Huntington and Lyrintzis have proposed a so-called single-switch-optimized sample ordering scheme. The approach is based on iterative switching of a pair of samples of Table 1, which gives the greatest reduction in correlation error. Huntington and Lyrintzis have shown that their technique clearly performs well enough. However, it may still converge to a non-optimum ordering. A different method is needed for simulation of both uncorrelated and correlated random variables. Such a method should be adequately efficient: reliable, robust and fast.

As will be discussed in the companion Part III [34], the transformation approaches are unable to exhaust all possible dependency patterns between marginal random variables. Therefore, they may turn out to be useless in practical applications with estimated marginals and covariances incompatible with the multivariate Gaussian models after transformations. The presented work is motivated mainly by the work in [21] and proposes a direct approach involving the induction of target correlation into samples of random variables, which avoids transformations based on multivariate Gaussian or other distributions.

3. Correlation control by simulated annealing combinatorial optimization

The correlation control problem is treated as a combinatorial optimization problem in this paper; we attempt optimization of the ranks of sample values. It is well known that deterministic optimization techniques and simple stochastic optimization approaches can very often fail to find the global minimum [45,46].

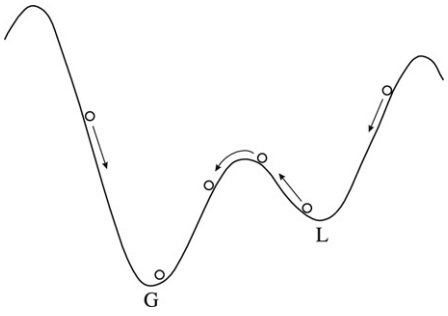

Fig. 2. Energy landscape—the problem of local and global minima.

They are generally strongly dependent on the starting point (as will become clear later, the starting point corresponds to the initial configuration of the sampling scheme, see Table 1). Such techniques fail and finish with a local minimum such that there is no chance to escape from it and find the global minimum (Fig. 2). The ball in the illustrative figure jumps from one minimum to another minimum when the energy landscape has high energy (for understanding let us imagine the shaking of the landscape). If the energy is low, the ball will remain in one of the minima—the local or global one. It is obvious that the best procedure for finding the global minimum is to start with high energy (temperature, excitation) and then step by step decrease this temperature to almost zero (*cooling*). During such a process the lowest position of the ball has to be monitored: at the end it corresponds to the global minimum (or at least to a "very good" local one). It can be intuitively predicted that in our case we are definitely facing a problem with multiple local minima. Therefore, the use of a stochastic optimization method (which works with a nonzero probability of escaping from local minima) seems to be a rational solution. The simplest form of the stochastic optimization algorithm is the evolution strategy working in two steps: *mutation* and *selection*. During the mutation step, an "offspring" configuration is generated from a current "parent" configuration. The selection step chooses the best configuration between the "parent" and "offspring" to survive. The simplest form of the algorithm selects offspring giving a smaller value of a certain *objective function E*. However, such a rule can lead to a local minimum solution. The selection step can be improved by the Simulated Annealing approach, a technique which is very robust concerning the starting point (initial arrangement of the sampling scheme in Table 1).

3.1. General remarks on the Simulated Annealing technique

The Simulated Annealing (SA) method originated in the early 1980s when Kirkpatrick et al. [7] and Černý [8,9] independently explored an analogy between the physical annealing process in solids and the task of solving large combinatorial optimization problems. A good coverage of the topic is provided by [45]. A quite complete bibliography list is provided in [47].

Annealing, in metallurgy, refers to a heat treatment that alters the microstructure of a material causing changes in properties such as strength and hardness. It causes a solid in a heat bath to enter low energy states. In this process, the solid is first heated to melting point and then slowly cooled until the low energy ground state is reached. If the initial temperature is not sufficiently high or the cooling is carried out too quickly, the solid will freeze into a metastable state rather than the ground state. Physical annealing can be modeled successfully using computer simulation methods. In one of these approaches, Metropolis et al. [10] in 1953 introduced a simple Monte Carlo algorithm which generates a sequence of states in the solid in the following way. Given a current state i of the solid, characterized by the positions of its particles, with energy

$E_i = E_{\text{parent}}$, a subsequent state $i+1$ (called "offspring" here), with energy $E_{i+1} = E_{\text{off}}$, is generated by applying a small distortion to the current state (mutation), for example by the displacement of a particle. The acceptance rule for the offspring configuration $(i+1)$ depends on the energy difference $\Delta E = E_{\text{off}} - E_{\text{parent}} = E_{i+1} - E_i$:

$$P(\Delta E) = \begin{cases} P_-(\Delta E) = 1, & \Delta E \le 0 \\ P_+(\Delta E), & \Delta E > 0. \end{cases} \tag{7}$$

In the so-called Boltzmann annealing, the acceptance probability for an energetically higher configuration is related to the Boltzmann distribution:

$$P_+(\Delta E) = \frac{\exp\left(-E_{\text{parent}}/t\right)}{\exp\left(-E_{\text{off}}/t\right) + \exp\left(-E_{\text{parent}}/t\right)} = \frac{1}{1+\exp\left(\Delta E/t\right)} \approx \exp\left(\frac{-\Delta E}{t}\right) \tag{8}$$

where t denotes the temperature of the heat bath. This acceptance rule is known as the *Metropolis criterion* and the associated algorithm is known as the *Metropolis algorithm*. The Metropolis algorithm was generalized by the Kirkpatrick algorithm to include a temperature schedule for efficient searching [7]. Therefore, the Simulated Annealing algorithm can now be viewed as an iteration of Metropolis algorithms, evaluated at decreasing values of the control parameter t. Initially, the control parameter is given a large value, and starting with an initial randomly chosen feasible solution, a sequence (a random walk through the solution space or Markov chain) of *trials* is generated. In each trial, an offspring configuration $i+1$ is generated by applying a random elementary change to the current configuration i (the set of configurations attainable by such changes is called the neighborhood of i).

The control parameter (temperature) t is lowered in steps with the system being allowed to approach equilibrium through the sequence of N_{trials} trials at each step. After an appropriate *stopping condition* is met, the final configuration may be taken as the solution of the problem at hand. Suppose t_i is the value of the control parameter (temperature) and N_{trials} is the length of the Markov chain generated at the ith iteration of the Metropolis algorithm (number of loops or trials at the temperature t_i. Suppose u is a realization of the random variable $U_{0,1}$ uniformly distributed over [0; 1). Then, the Simulated Annealing algorithm may be expressed by a flowchart in Fig. 4. Even though in the figure we sketched the exact implementation as proposed and used for correlation control, the algorithm is general. The performance of the algorithm will be influenced by the following factors (termed the *cooling schedule*):

(1) The initial value of temperature t_0;
(2) The method of reducing the value of t after a sequence of trials;
(3) The minimum temperature $t_{\min}$;
(4) The length of the Markov chain for each value of t (number of loops, trials N_{trials});
(5) The choice of stopping condition(s).

3.2. Application of annealing to correlation control in Monte Carlo sampling

The imposition of a prescribed correlation matrix into a sampling scheme can be understood as an optimization problem: we want to *minimize the difference* between the target correlation matrix (e.g. user-defined, prescribed) $\boldsymbol{T}$ and the actual correlation matrix $\boldsymbol{A}$ (estimated from samples via a suitable statistic point estimator such as Pearson's correlation coefficient, Spearman's rho or Kendall's tau). Let us denote the difference matrix (error matrix) $\boldsymbol{E}$:

$$\boldsymbol{E} = \boldsymbol{T} - \boldsymbol{A}. \tag{9}$$

Moreover, we may, for a variety of reasons, want to highlight the importance of arbitrary entries in the matrix $\boldsymbol{T}$ (e.g. user-defined weights based on better knowledge of correlation e.g. from measurements). Such an accentuation can be introduced by a weight matrix $\boldsymbol{W}$ (a square and symmetric matrix of the same order as $\boldsymbol{T}$ and $\boldsymbol{A}$). The (symmetric) error matrix $\hat{\boldsymbol{E}}$, taking into account the weights of entries, is constructed as:

$$\widehat{E}_{i,j} = W_{i,j} E_{i,j}, \quad i,j = 1,\ldots,N_{\text{var}}. \tag{10}$$

If there is no accentuation of any particular correlation coefficient, the matrix $\boldsymbol{W}$ is filled with unit weights and $\boldsymbol{E} = \hat{\boldsymbol{E}}$.

To have a scalar measure of the error we introduce a suitable matrix norm for $\hat{\boldsymbol{E}}$. In particular, a good and conservative measure of the distance between $\boldsymbol{T}$ and $\boldsymbol{A}$ can be the norm defined as:

$$\rho_{\max} = \frac{\max\limits_{1\le i<j\le N_{\text{var}}} \left|\widehat{E}_{i,j}\right|}{W_{i,j}} \tag{11}$$

where indices i,j in the denominator are those maximizing the nominator. We simply find the maximum entry of $\hat{\boldsymbol{E}}$ and divide it by its weight to get the error in correlation. Even though this norm has a clear meaning and known "units" (correlation) and can be used as a good stopping condition in the iterative algorithm, it is not a suitable objective function to be subjected to direct minimization, see part III [34]. A better choice is a norm taking into the account deviations of all correlation coefficients:

$$E = \sum_{i=1}^{N_{\text{var}}-1} \sum_{j=i+1}^{N_{\text{var}}} W_{i,j} \left(E_{i,j}\right)^2 \tag{12}$$

where we used the symmetry of the correlation matrices by summing up the squares of the upper triangle off-diagonal terms only. This norm proved itself to be a good objective function for the optimization algorithm described below. The objective function E can be further normalized by the sum of the weights of the considered correlation coefficients (entries of the one off-diagonal triangle of $\boldsymbol{T}$), and taking the square root yields a measure in units of correlation:

$$\rho_{\text{rms}} = \sqrt{\frac{E}{\sum\limits_{1\le i<j}^{N_{\text{var}}} W_{i,j}}} \tag{13}$$

which represents the normalized weighted error per entry and is, therefore, suitable for comparison when examples of a different number of variables N_{var} are involved. Note that if equal unit weights are used ($W_{i,j} = 1$ for $i,j = 1,\ldots,N_{\text{var}}$) this norm simplifies to

$$\rho_{\text{rms}} = \sqrt{\frac{2E}{N_{\text{var}}(N_{\text{var}}-1)}} \tag{14}$$

which is called the *root mean square correlation* by Owen [48].

Note that for computer implementation of minimization it is not necessary to compute the square root (Eq. (13) or (14)), as only the differences between the norms matter. Eq. (12) suffices for the objective function and also the normalization can be done after the iterative algorithm has been completed. However, the initial and terminating temperature (see later) must be redefined accordingly.

The norm E (Eq. (12)) has to be minimized; from the point of view of definition of the optimization problem, the *objective function* is E and the *design variables* are related to *ordering* in the sampling scheme (Table 1). Clearly, in real applications the space of the possible actual correlation matrices $\boldsymbol{A}$ is extremely large and we want to find an efficient near-optimal solution.

With a combinatorial optimization problem we can simulate the annealing process by making the following correspondences: (i) A feasible solution during the process (the ordering of samples in Table 1) corresponds to states of the solid; (ii) the cost E is a function defined in the combinatorial problem and it corresponds to the energy of the actual state (configuration); (iii) a randomly chosen feasible initial state (configuration of sample, Table 1) corresponds to the state of the solid after it has been heated to melting point (iv) an elementary change to a solution corresponds to a slight distortion in the state of the solid; (v) the control parameter t corresponds to the temperature of the heat bath.

To review the two step optimization strategy as introduced in Section 3; in the proposed algorithm, the first step **(mutation)** is performed by a transition called a *swap* from the parent configuration i (with error E_{parent}) to the offspring configuration (E_{off}). A swap (or a *trial*) is a small change to the arrangement of Table 1. It is done by randomly interchanging a pair of two values $x_{i,j}$ and $x_{i,k}$. In other words one needs to randomly generate $i \in \langle 2, N_{\text{var}} \rangle$ (choose the variable), and a pair $j, k \in \langle 1, N_{\text{sim}} \rangle, j \neq k$ (choose the pair of realizations to interchange), see Fig. 5. Such a change to the arrangement of samples requires the recalculation of $N_{\text{var}} - 1$ correlation coefficients in $\boldsymbol{A}$ (associated with variable i) to update the objective function $E_{\text{parent}} \rightarrow E_{\text{off}}$. Naturally, the "parent norm" E_{parent} is calculated using the parent configuration and the "offspring norm" E_{off} is calculated using the newly obtained offspring configuration. One swap may or may not lead to a decrease (improvement) in the norm.

Note that a version of swapping may also be an interchange of more than one pair of values of one variable or more variables (multiple swaps as defined above in one transition between configurations). The experience is that swapping more than a pair of values at a time does not seem to improve convergence of the algorithm. Therefore, from here on, we will not consider such a vector type of swap.

In the second step **(selection)** one configuration between the "parent" and "offspring" is selected to survive. In the simplest version a configuration (sampling scheme arrangement) giving a smaller value of the objective function (norm E) is selected. Such an approach has been intensively tested using numerous examples. It has been observed that the method, in most cases, could not capture the global minimum. It failed in a local minimum as only the improvement of the norm E resulted in acceptance of "offspring". Note that a similar approach, in a way, was applied in [49] for random autocorrelated sequences.

An improvement is here proposed by employing the Simulated Annealing technique (selection rule from Eq. (7)). As a result there is a much higher probability that the global minimum will be found in comparison with deterministic methods and the simple evolution strategies. The advantage of this compared to the simple evolution strategy described above (corresponding to $t = 0$ in Eqs. (7) and (8)) is that there is a nonzero probability of accepting an offspring configuration with higher error than its parent (hill climbing). The acceptance rule with the nonzero current temperature t in Eq. (7) gives us a mechanism for accepting increases in a controlled fashion. It is possible that accepting an increase in the penalty function (E) will reveal a new configuration that will avoid a local minimum or at least a bad local minimum.

The probability of accepting an increase (Eq. (7)) is driven by the difference of norms ΔE and the excitation (temperature t). This probability distribution expresses the concept that a system in thermal equilibrium at temperature t has its energy probabilistically distributed among all different energy states ΔE. Note that "offspring" leading to a decrease in the objective function is naturally accepted for the new generation (see Eq. (7)). At high temperatures (the beginning of the optimization process), the quantity $P(\Delta E)$ is usually close to one for $\Delta E > 0$. Thus, a state

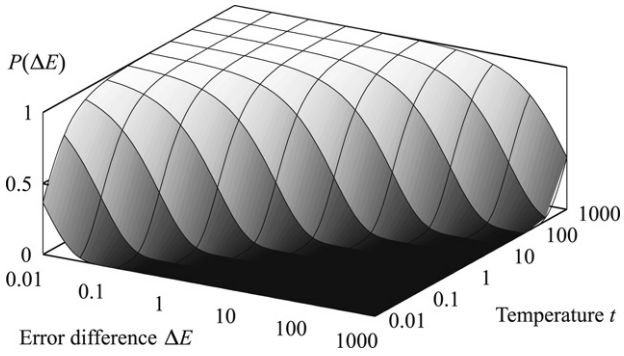

Fig. 3. Acceptance probability for "hill climbing" (ΔE and t in a logarithmic scale).

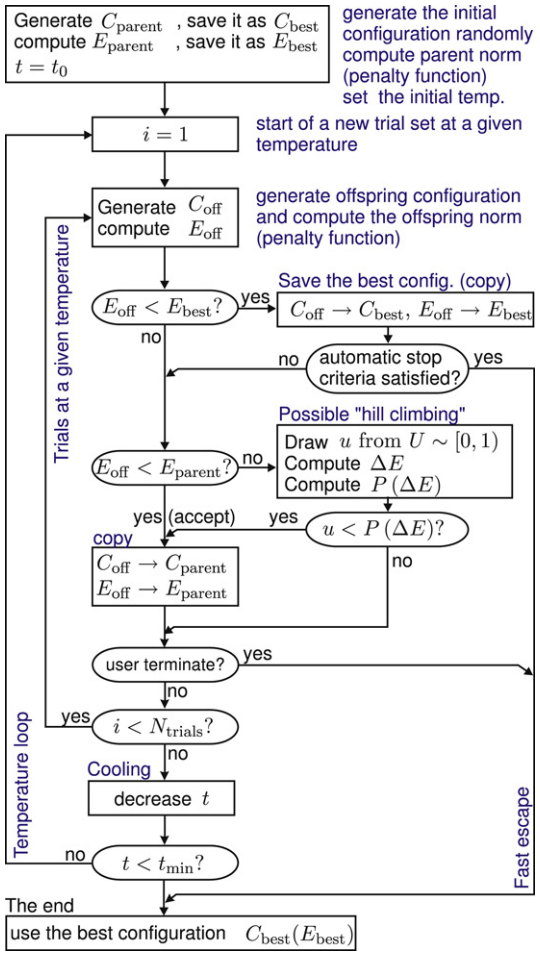

Fig. 4. Flowchart of Simulated Annealing algorithm implementation.

with greater E is usually accepted. As the temperature lowers, the system freezes, and configurations with increases of E are accepted with diminishing probability, see Fig. 3. Also as the temperature falls, the algorithm converges to the optimum or a state which is close to optimal, see Fig. 6. Even at low temperatures, there is a chance (although very small) of a system being locally in a high energy state.

At the end of an SA algorithm we accept the best configuration obtained throughout the whole process and perform other N_{trials} random changes at zero temperature (i.e. accepting only improving configurations). This enables us to search the surroundings of the last obtained minima without the chance of hill climbing. Our experience is that sometimes the solution can be slightly improved.

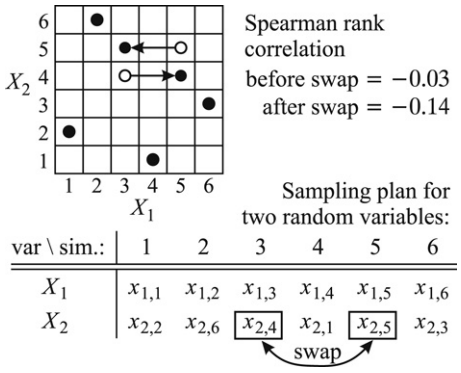

Fig. 5. Illustration of a random trial—swap of samples i and j of variable X_2.

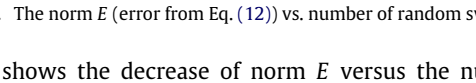

Fig. 6. The norm E (error from Eq. (12)) vs. number of random swaps.

Fig. 6 shows the decrease of norm E versus the number of swaps during the SA process. Such a figure is typical and should be monitored. In the figure we also plot the temperature and the current best value of the minimized norm E.

3.3. Cooling schedule

As already mentioned in the classical application of the SA approach for the optimization of unexplored functions there is the problem of finding the optimal cooling schedule. In particular, the five parameters listed at the end of Section 3.1 are generally unknown. Usually they must be set heuristically. Fortunately, the correlation control problem is constrained in the sense that all possible elements of the correlation matrix are always within the interval $\langle -1; 1\rangle$. This makes the application of the method to correlation control very straightforward and easily automated.

The *maximum* of the norms (11) and/or (12) can be estimated using the prescribed matrix $\boldsymbol{T}$ and hypothetically the "most remote" matrix $\boldsymbol{A}$ (filled with unit correlation coefficients, plus or minus). For small to moderate correlations our experience with countless numerical examples has shown that a good starting temperature is: $t_0 = 50/N_{\text{sim}}$ when the sample ordering table is randomly rearranged before the start of the Simulated Annealing. Also, we know that the *minimum* of the objective function is zero ($\boldsymbol{T}$ and $\boldsymbol{A}$ match). These facts represent a significant advantage: the heuristic estimation of initial temperature is not necessary—the initial setting of parameters can be performed automatically without estimation by the user and the "trial and error" procedure. The minimum temperature can be set to be equal (in order) to the required value of error function E. Concerning the number of trials at a given temperature, Kirpatrick et al. [7] give the following guidance: at each temperature, the simulation must proceed long enough for the system to reach a steady state. Our experience is that an optimal number of trials is related to the size of the searched space, i.e. to both N_{var} and N_{sim}. We recommend using $N_{\text{trials}} = NN_{\text{var}}N_{\text{sim}}$ (the value N varies between 1 and 10 and its choice is clarified in the companion part III paper).

The initial temperature has to be decreased gradually, e.g. using the reduction function $f(t_j)$ after a constant number of trials N_{trials} applied at temperature t_j. A sufficiency proof was shown to put a lower bound on the temperature schedule as $t_j = t_0/\ln(j)$ [50]. A logarithmic temperature schedule is consistent with the Boltzmann algorithm, e.g. the temperature schedule can be taken to be $t_j = t_0 \ln(j_0)/\ln(j)$, where j_0 is a starting index. Some researchers using the Boltzmann algorithm use exponential schedules, e.g. $t_j = t_0 \exp((c-1)j)$, where $0 < c < 1$. An exponential temperature schedule:

$$t_{j+1} = ct_j, \tag{15}$$

has been widely used with great benefit in combinatorics with $0.7 \leq c \leq 0.99$. We use this simple rule for the jth temperature with $c = 0.95$, i.e. in each cooling step j after N_{trials} trials the temperature is obtained as $t_j = c^j t_0$. The number of temperature steps from t_0 till $t_{\min}$ is known in advance and can be easily computed as $\ln(t_0/t_{\min})/\ln(c)$ rounded down to an integer number. The result can be multiplied with N_{trials} to get the maximum total number of swaps (tested configurations) during one run of the algorithm (reached if no stopping condition is used). To conclude, we refer to the above-cited references or publications [45,46,51] for more sophisticated cooling schedules known in Simulated Annealing theory.

4. Numerical examples

4.1. Univariate sampling

This section presents a comparison of two sampling schemes introduced in Section 2.1. The compared sampling schemes are denoted as LHS-median (Eq. (2)) and LHS-mean (Eq. (3)). For comparison, sample sets from two different distribution functions were used, namely Gaussian distribution (symmetric around the mean value) and the exponential distribution (skewed). Convergence of the estimated statistics from N_{sim} values are plotted against the number of samples N_{sim}. In particular, convergence of four point estimations of the mean value, standard deviation, skewness and kurtosis excess to the target values is studied. The target values are {1, 1, 0, 0} in the case of Gaussian and {1, 1, 2, 6} in the case of exponential distributed variables.

The results are plotted in the middle and right columns of Fig. 7. Since the Gaussian distribution is symmetric the average and sample skewness always match the mean and skewness values irrespective of N_{sim}. In the case of standard deviation and kurtosis, we see the slower convergence of the LHS-median samples. Using the non-symmetric exponential distribution, we highlight the fact that LHS-mean samples always match the mean value (it obeys its definition, Eq. (3)) while LHS-median samples may suffer from considerable errors in the mean for very small samples. Overall, the convergence of statistics to the target moments is faster for the LHS-mean scheme.

To illustrate where the differences come from, a comparison of samples selected for $N_{\text{sim}} = 1 \ldots 10$ is made in Fig. 7 left. From the figure, it is clear that the differences between samples selected mainly concern the samples in the tails while samples in the core region are nearly identical for both schemes.