ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

LATIN HYPERCUBE SAMPLING BASED ON ADAPTIVE ORTHOGONAL DECOMPOSITION

Dirk Roos

Institute of Modelling and High-Performance Computing
Niederrhein University of Applied Sciences, Germany
dirk.roos@hs-niederrhein.de

Keywords: Experimental and Simulation Design, Latin Hypercube Sampling, Proper Orthogonal Decomposition, Singular Value Decomposition.

Abstract. *In the field of real world or simulation experiments, latin hypercube designs are very useful to approximate implicit given functions based on experimental designs of rectangular grid of points used e.g. for an efficient optimization of black-box experiments. The so called standard latin hypercube sampling is based on an idea of [8] to simulate correlated multivariate relationships based on uncorrelated variables. Hereby, the linear or rank correlations between the random variables can be reduced using a Cholesky decomposition of the covariance matrix. However, this algorithm requires that the number of samples is higher than the number of input variables to ensure a positive definite and non-singular covariance matrix. Furthermore, a rearrangement of the latin hypercubes so that they will have the same ordering to increase the multidimensional uniformity introduces an unwanted pseudo linear correlation of the simulated variables.*

In this paper, the new method of proper orthogonal decomposition methods in adaptive combination with re-sorting the rank-order of the original marginal distributions not only eliminates the drawbacks of the standard latin hypercube sampling, but also reduces the number of experimental designs necessary. In contrast to the standard latin hypercube sampling, the adaptive approach results in a further improvement of the independency and the multidimensional uniformity of the variables and shows an improved efficiency in comparison to optimization-based methods. Furthermore, a lower-rank approximation of the design matrix based on singular value decomposition ensures a positive definite and non-singular correlation matrix to reduce the correlation error in a stepwise manner during the adaptive procedure.

1 INTRODUCTION

Physical or computer experiments are widely used for the design and optimization of products and processes. In simulation experiments, rather than carrying out physical experiments, numerical models describing the performance of the product or the process are only implicitly given through numerical methods. When modelling complicated physical phenomena we are searching for a mathematical model

$$y = f(x_1, ..., x_i, ..., x_n) = f(\mathbf{x}), \quad \mathbf{x} = (x_1, ..., x_i, ..., x_n)^T \in \mathbb{D}^n \subset \mathbb{R}^n$$

where the vector $\mathbf{x}$ consists of the n design (input) variables or factors and y defines a specific output variable (response). The function $f(\mathbf{x})$ is defined within the design space $\mathbb{D}$ with the lower and upper bounds x_i^l, x_i^u of the variables.

In general, efficient design exploration techniques, such as sensitivity analysis and optimization are based on surrogate (or meta) models $\tilde{y}(x_i)$ of response values using evaluation points or the so-called designs. A design of experiments with N points and n variables is usually written as a $N \times n$ design matrix

$$\mathbf{X} = \left[\mathbf{x}^1, ..., \mathbf{x}^k, ..., \mathbf{x}^N\right]^T \in \mathbb{D}^{N,n} \subset \mathbb{R}^{N,n}$$

where each row

$$\mathbf{x}^k = \left[x_1^k, ..., x_i^k, ..., x_n^k\right]$$

represents a sample k and each column represents a variable i. Obviously, in a real world design of experiment, it is not possible to enlarge the necessary number of experimental designs n to obtain acceptable space-filling and independency of the simulated design space which is important for an accurate sensitivity analysis and for design space exploration. So one of the main aims in the field of experiments is therefore to obtain efficient designs of experiments.

A commonly used stratified sampling technique is the latin hypercube sampling, especially if the number of design variables is greater than $n = 5$ and general nonlinearities of the model responses are possible. Numerous modifications have been proposed for latin hypercube sampling to minimize correlation error and to ensure the unidimensional uniformity. The pseudo linear correlations can be minimized by an internal optimization procedure using a column pairwise switch algorithm to increase uniformity or space-fillingness of a sampling plan. Several methods are proposed in the literature, e.g.[7, 13] use a simulated annealing and in [9] an enhanced stochastic evolutionary is presented. Commonly used optimization criteria are the maximization of the minimum distance (maximin metric) according to [13] introduced by [10, 5] and mean of the square of the pairwise correlations, proposed by [8]. A Matlab implementation of an optimization algorithm to construct these maximin latin hypercubes is given in [5].

Other methods are developed to create small orthogonal or nearly-orthogonal latin hypercube matrices [12], which are perfectly uniform in the multivariate space. These methods are applicable only for small sets of variables and design evaluations. In [2] a method is presented for calculating nearly orthogonal latin hypercubes and calculating the best space-filling designs which is computationally expensive. The method presented in [4] does not necessarily calculate the best space-filling designs but is applicable to high dimensional problems, preserving the integrity of the latin hypercube sampling estimator by increasing the statistical distance between designs using sequentially eliminating design evaluations that are near to each other in the multidimensional space. In [14] another method for reducing correlations of a latin hypercube using the Gram-Schmidt orthogonalization is proposed. An extended overview of space-filling designs can be found in [3].

Because of the huge combinatorial effort of the assigned optimization problem, finding an optimized latin hypercube sampling is a very difficult and time-consuming task and restricted to a relative small number of variables. Otherwise, several decomposition methods of the covariance matrix exist to reduce the unwanted pseudo correlation of the variables in a high dimensional space, which can be enhanced to improve unidimensional uniformity and independency of the design points, as shown in the following.

2 STANDARD LATIN HYPERCUBE SAMPLING

2.1 Plain latin hypercube sampling

A plain Monte Carlo simulation can be used to create $i = 1, ..., n$ nearly-uniformly distributed samples $U_i^k(0.5; 1/(2\sqrt{3}))$ with $k = 1, ..., N$ designs with normalization into a $[0, 1]^n$ box. Unfortunately, a plain Monte Carlo simulation does not consider that every class of all variables should be simulated with the same probability and results in non-uniformity of the marginal distributions.

The basic idea of [15] assumes that each sample out of N is unique with high probability. Of course, sorting these samples in ascending order gives random numbers too, and returns a row vector of $\mathbf{u}$ for every variable $i = 1, ..., n$ containing a random permutation of the integers from $k = 1, .., N$. The permutation matrix $\pi_{U_i}^k$ of U_i^k ensures that the bins have their centers in the center of the latin hypercubes with

$$H_i^k = \frac{\pi_{U_i}^k - 0.5}{N}$$

The splitting of the design space $\mathbb{D}^n$ into N classes $\mathbb{D}_k^i$ with the same probability

$$P\left[h_i^k \in \mathbb{D}_k^i\right] = \frac{1}{N}, \qquad i = 1, ..., n, \quad k = 1, ..., N$$

gives N^n hypercubes (bins) with the probability N^{-n}. Hereby, all portions of the random variables' range are represented and therefore we obtain a univariate uniformity of the marginal distribution of the variables. Unfortunately, this procedure leads to an unwanted correlated sample matrix

$$\mathbf{H} = \begin{bmatrix} h_1^1 & h_2^1 & \cdots & h_n^1 \\ h_1^2 & h_2^2 & ... & h_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^N & h_2^N & \cdots & h_n^N \end{bmatrix}$$

especially for a small number of samples N. In some cases the maximal error of the correlation matrix of a plain latin hypercube sampling

$$\epsilon_{\boldsymbol{\rho}_{\mathbf{HH}}} = \max_{1 \le i,j \le n} \left(|\boldsymbol{\rho}_{\mathbf{HH}} - \mathbf{I}|\right)$$

can be larger than the maximal correlation error $\epsilon_{\boldsymbol{\rho}_{\mathbf{UU}}}$ of a plain Monte Carlo sampling.

2.2 Correlation transformation using factorization of the covariance matrix

According to [8], the unwanted linear or rank correlations between the random variables can be eliminated using a Cholesky decomposition of the covariance matrix

$$\mathbf{C}_{\mathbf{HH}} = E\left[\left(\mathbf{H} - \bar{\mathbf{H}}\right)\left(\mathbf{H} - \bar{\mathbf{H}}\right)^T\right]$$

whereby $E[.]$ denotes the expectation value and $\bar{\mathbf{H}}$ the vector of the means of the latin hypercube samples. Using the standardization $Y_i(0;1) \sim H_i$ by

$$Y_i = \frac{H_i - E[H_i]}{\sigma_{H_i}} \tag{1}$$

the covariance matrix of Y_i is equal to the assigned correlation matrix

$$\boldsymbol{\rho}_{\mathbf{YY}} = \mathbf{C}_{\mathbf{YY}} = E\left[\mathbf{Y}\mathbf{Y}^T\right] \tag{2}$$

In the general case a variable transformation $Z_i(0;1) \sim Y_i$ is to find so that the correlation matrix of all variables Z_i represents the target correlation in terms of the identity matrix, thus

$$\boldsymbol{\rho}_{\mathbf{ZZ}} = E\left[\mathbf{Z}\mathbf{Z}^T\right] \stackrel{!}{=} \mathbf{I} \tag{3}$$

Inserting Eq. (2) and Eq. (3) in

$$\boldsymbol{\rho}_{\mathbf{ZZ}}\, \boldsymbol{\rho}_{\mathbf{ZZ}}^{-1} = \mathbf{I} = \boldsymbol{\rho}_{\mathbf{YY}}\, \boldsymbol{\rho}_{\mathbf{YY}}^{-1} \tag{4}$$

and using the Cholesky factorization of the matrix of the pseudo correlations

$$\boldsymbol{\rho}_{\mathbf{YY}} = \mathbf{L}\mathbf{L}^T \tag{5}$$

we obtain

$$E\left[\mathbf{Z}\mathbf{Z}^T\right] = \mathbf{I}\,\mathbf{I}^T = \left(\mathbf{L}^{-1}\mathbf{L}\right)\left(\mathbf{L}^{-1}\mathbf{L}\right)^T = \mathbf{L}^{-1}\mathbf{L}\,\mathbf{L}^T\mathbf{L}^{-1^T} = \mathbf{L}^{-1}\boldsymbol{\rho}_{\mathbf{YY}}\mathbf{L}^{-1^T}$$

and furthermore

$$E\left[\mathbf{Z}\mathbf{Z}^T\right] = \mathbf{L}^{-1}E\left[\mathbf{Y}\mathbf{Y}^T\right]\mathbf{L}^{-1^T} = E\left[\mathbf{L}^{-1}\mathbf{Y}\;\mathbf{Y}^T\mathbf{L}^{-1^T}\right] = E\left[\underbrace{\left(\mathbf{L}^{-1}\mathbf{Y}\right)}_{\mathbf{Z}}\underbrace{\left(\mathbf{L}^{-1}\mathbf{Y}\right)^T}_{\mathbf{Z}^T}\right]$$

In the case of the non-singular lower triangle matrix $\mathbf{L}$ the transformation results in

$$\mathbf{Z} = \mathbf{L}^{-1}\mathbf{Y} \tag{6}$$

and deals in this way numerically without correlation error. The back substitution of Eq. (1)

$$U_i = Z_i \cdot \sigma_{H_i} + E[H_i]$$

gives normalized, nearly-uniformly distributed variables U_i. Finally, the uniform distributed variables $X_i \sim U_i$ with respect to the vectors of lower and upper bounds $\mathbf{x}^l$ and $\mathbf{x}^u$ are simply given by

$$\mathbf{X} = \mathbf{x}^l + \mathbf{U} \circ \left(\mathbf{x}^u - \mathbf{x}^l\right) = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_n^N \end{bmatrix} \in \mathbb{D}^{N,n} \subset \mathbb{R}^{N,n} \tag{7}$$

whereby, the symbol $\circ$ denotes the entrywise product.

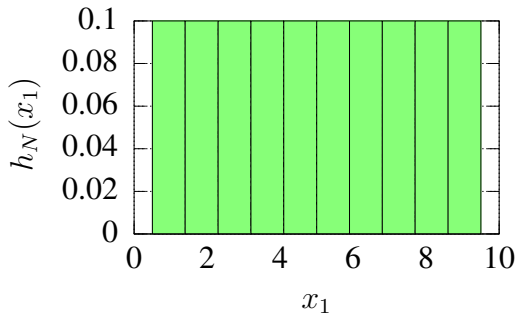

Figure 1: Histogram of the first variable x_1, simulated by Eq. (7).

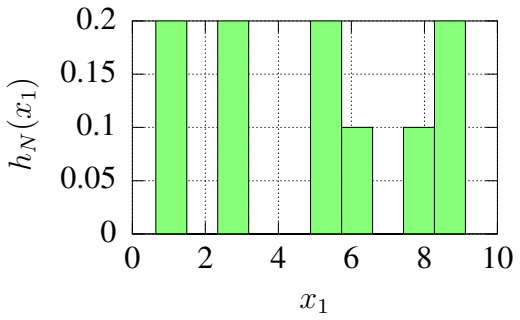

Figure 2: Histogram of the second variable x_2, simulated by Eq. (7).

2.3 Rearrangement of the latin hypercubes

On the other hand, the linear transformation in Eq. (6) moves the sample points coordinates for all variables $i > 1$. And as a result the samples may be moved beyond the desired bounds and marginal distributions may be effected from the change of correlation, as shown in Fig. (1) and Fig. (2). To solve this problem, in [8] a resorting of the design matrix $\mathbf{Z}$ according to the rank correlation of the exact marginal distributions of $\mathbf{H}$

$$H_i^k = \frac{\pi_{Z_i}^k[\pi_{Z_i}^k] - 0.5}{N}$$

is presented. Whereby, the inner permutation matrix describes the rank-order sorting of the N samples of n variables of the design matrix $\mathbf{Z}$ and the outer permutation matrix describes the rank-order sorting of the sorted indices to create modified center-based latin hypercubes $\mathbf{H}$ with accurate marginal distributions to ensure univariate uniformity. Unfortunately, this rearrangement within the so-called standard latin hypercube sampling achieves the multidimensional uniformity but introduces an unwanted pseudo linear correlation of the simulated variables

$$\mathbf{X} = \mathbf{x}^l + \mathbf{H} \circ \left(\mathbf{x}^u - \mathbf{x}^l\right)$$

again. But in summary, the maximal correlation error of the simulated variables

$$\epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}} = \max_{1 \le i,j \le n} \left(|\boldsymbol{\rho}_{\mathbf{XX}} - \mathbf{I}|\right) \tag{8}$$

is significantly smaller than the maximal correlation error $\epsilon_{\boldsymbol{\rho}_{\mathbf{HH}}}$ of the first latin hypercube sampling.

3 LATIN HYPERCUBE SAMPLING BASED ON ADAPTIVE ORTHOGONAL DECOMPOSITION

3.1 Adaptive rearrangement of the latin hypercubes

Particularly, in high-dimensional design spaces and with a reduced number of design evaluations $n \lesssim N$ the correlation error in Eq. (8) increases as a result of the rank-order sorting. Otherwise, the linear transformation in Eq. (6) achieves an exact correlation and maintain the uniformity of the first variable. So, an essential enhancement of the standard latin hypercube sampling is an adaptive rearrangement of the columns of the current given latin hypercube matrix $\mathbf{H}$ in each iteration step, as shown in Fig. (3).

$$[|\boldsymbol{\rho}_{\mathbf{XX}} - \mathbf{I}|] = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & \dots & \dots & j & \dots & n \end{array} \\ \begin{pmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{pmatrix} \end{array} \quad \mathbf{H} = \begin{bmatrix} h_1^1 & \cdots & h_i^1 & \cdots & h_j^1 & \cdots & h_n^1 \\ h_1^2 & \cdots & h_i^2 & \cdots & h_j^2 & ... & h_n^2 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ h_1^N & \cdots & h_i^N & \cdots & h_j^N & \cdots & h_n^N \end{bmatrix}$$

Figure 3: Iterative rearrangement of the columns of the latin hypercube matrix $\mathbf{H}$, according to the rows number i and the columns number j of the maximal correlation error and the first column with accurate uniformity.

3.2 Orthogonal transformation using proper orthogonal decomposition

The Cholesky factorization Eq. (5) requires that the number of samples is greater than the number of input variables, thus $N > n$ to ensure a positive definite pseudo correlation matrix. Otherwise, this kind of decomposition is not applicable. Eq. (5) can be replaced by principal component analysis, e.g. using the eigenvalue decomposition

$$\boldsymbol{\rho}_{\mathbf{YY}}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}$$

whereby $\mathbf{V}$ denotes the matrix of the eigenvectors and $\boldsymbol{\Lambda}$ the diagonal matrix of the eigenvalues. Inserting the orthogonality condition $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ we obtain

$$\mathbf{V}^T\boldsymbol{\rho}_{\mathbf{YY}}\mathbf{V} = \boldsymbol{\Lambda}$$

and furthermore the spectral decomposition of the pseudo correlation matrix

$$\boldsymbol{\rho}_{\mathbf{YY}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}{\boldsymbol{\Lambda}^{\frac{1}{2}}}^T\mathbf{V}^T$$

which is a symmetric decomposition and can be used instead of Eq. (6) for the transformation of the design matrix

$$\mathbf{Z} = \left(\mathbf{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)^{-1}\mathbf{Y} \tag{9}$$

This transformation is recommended in case of a non-positive definite pseudo correlation matrix. However, in the next step, the case $N \leq n$ results in a singular correlation matrix within Eq. (9), again. Indeed, this is solvable using the Moore-Penrose pseudo inverse but this procedure would occur with the same order of error magnitude as using plain Monte Carlo simulation.

The proper orthogonal decomposition, also known as Karhunen-Loève decomposition, may serve to ensure the correlation transformation in case of $N < n$ within two steps, namely the order reduction of the design matrix by projecting high-dimensional data into a lower-dimensional space and orthogonal transformation of the sample covariance or correlation matrix to the basis of the largest eigenvalues. A bibliographical review and applications of the proper orthogonal decomposition is given in [11]. The discrete modelling of the proper orthogonal decomposition is achieved by the singular value decomposition. The singular value decomposition of the correlation matrix can be written as

$$\boldsymbol{\rho}_{\mathbf{YY}} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{U}\mathbf{S}^{\frac{1}{2}}\mathbf{S}^{\frac{1}{2}}\mathbf{V}^T$$

where $\mathbf{U}$ is an $n \times n$ orthonormal matrix containing the left singular vectors of the correlation matrix; $\mathbf{S}$ is an $n \times n$ diagonal and semi-positive definite matrix with diagonal entries containing

the singular values σ_i and $\mathbf{V}$ is an $n \times n$ orthonormal matrix containing the right singular vectors. Since, the diagonal matrix $\mathbf{S}$ contains non-negative real numbers only, we obtain a non-negative definite transformation matrix in Eq. (6) using

$$\mathbf{Z} = \left(\mathbf{U}\mathbf{S}^{\frac{1}{2}}\right)^{-1}\mathbf{Y} \tag{10}$$

There are several numerical methods for calculating the singular value decomposition, e.g. presented in [6].

3.3 Low-rank approximation of the design matrix

Unfortunately, the application of Eq. (10) for the singular case $N \leq n$ results in a numerically singular transformation matrix $\mathbf{U}\mathbf{S}^{\frac{1}{2}}$. This problem can be eliminated by projecting the design matrix $\mathbf{Y}$ from the n-dimensional variable space into a lower-dimensional space, see e.g. [1]. For the non-singular case $N > n$ a singular value decomposition of the design matrix is given by

$$\mathbf{Y} = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Phi}^T$$

where $\mathbf{\Psi}$ is an $N \times N$ unitary matrix, $\mathbf{\Sigma}$ is a $N \times n$ pseudodiagonal matrix with non-negative numbers on the diagonal also known as the singular values of the design matrix $\mathbf{Y}$ and $\mathbf{\Phi}$ is an $n \times n$ unitary matrix. The columns of $\mathbf{\Psi}$ and the columns of $\mathbf{\Phi}$ are called the left-singular vectors and right-singular vectors of $\mathbf{Y}$, respectively.

For the singular case we use any $m < N < n$ column-reduced and row-reduced matrices to calculate an optimal rank m approximation of the data matrix $\mathbf{Y}$ as follow

$$\mathbf{Y}_m = \mathbf{\Psi}_m\mathbf{\Sigma}_m\mathbf{\Phi}_m^T; \quad \mathbf{Y}_m \in \mathbb{R}^{N,n}, \mathbf{\Psi}_m \in \mathbb{R}^{N,m}, \mathbf{\Sigma}_m \in \mathbb{R}^{m,m}, \mathbf{\Phi}_m \in \mathbb{R}^{n,m}$$

with the leading $m \times m$ principal minor of $\mathbf{\Sigma}_m$ and the m proper orthogonal modes of the m columns of $\mathbf{\Phi}_m$. The rank m can be chosen stepwise afterwards to reach the optimality of the approximation in every iteration step of the adaptive orthogonal decomposition and rearrangement of the latin hypercubes.

4 SIMULATION EXAMPLES OF UNCORRELATED VARIABLES WITH MULTIDIMENSIONAL UNIFORMITY

4.1 Non-singular, positive definite covariance matrix

In the following example the convergence of the proposed adaptive procedure in comparison with other methods is presented to simulate uniform distributed variables $X_i^k(5; 10/(2\sqrt{3}))$ with $i = 1, ..., n = 100$ variables and $k = 1, ..., N = 120$ latin hypercube samples.

The initial error $\epsilon_{\boldsymbol{\rho}_{\mathbf{HH}}}$ of plain latin hypercube sampling of 0.323 is near the error of the Monte Carlo simulation $\epsilon_{\boldsymbol{\rho}_{\mathbf{UU}}}$ of 0.3228. The adaptive procedure converged continuously to an error of $\epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}} = 0.0045$ after 100 iterations. For comparison the errors of other methods are given in the Fig. (4). In particular, optimized latin hypercube sampling[1], according to the Matlab implementation of [5], Matlab implementation lhsdesign, based on the Gram-Schmidt orthogonalization[2], as proposed in [14] and standard latin hypercube sampling algorithm[3], as shown in section (2) with rearrangement of the latin hypercubes, introduced in [8].

4.2 Singular covariance matrix and low-rank approximation of the design matrix

For the singular case $N < n$ of the simulation of $i = 1, ..., n = 100$ uniform distributed variables $X_i^k(5; 10/(2\sqrt{3}))$ and $k = 1, ..., N = 80$ latin hypercube samples it is essential to

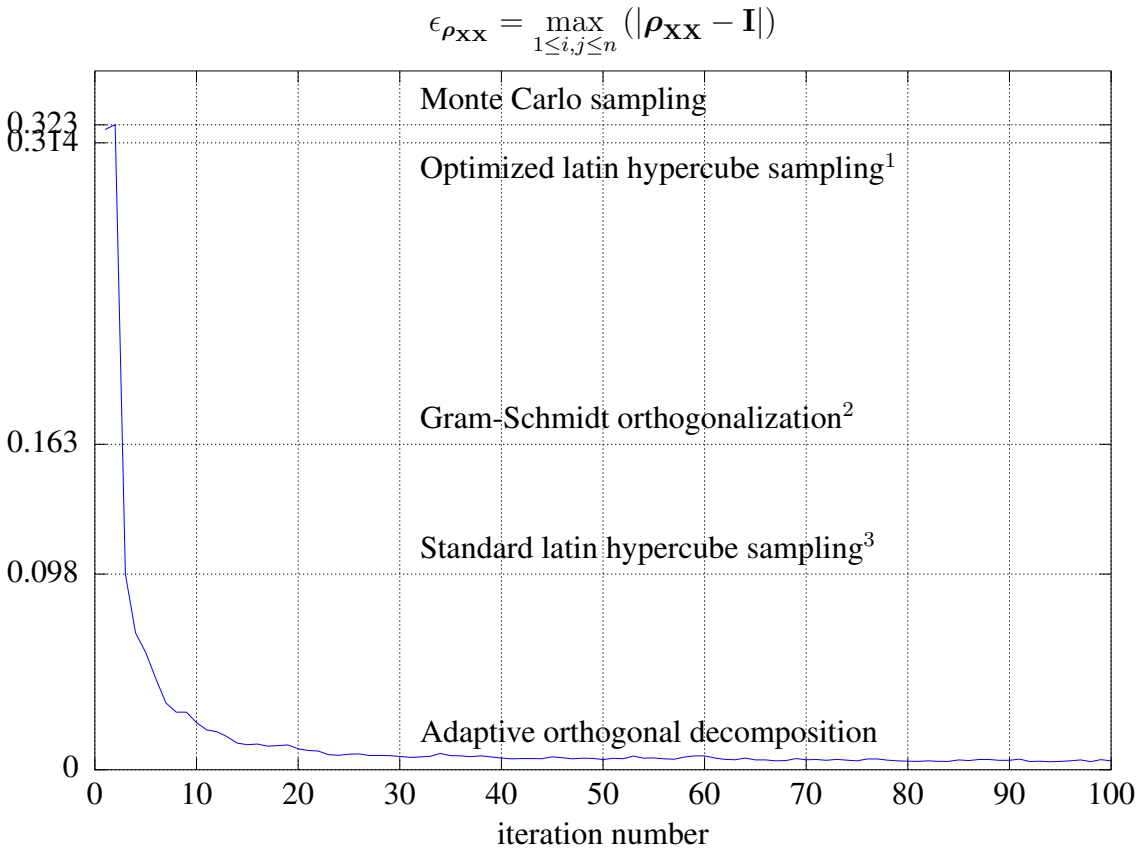

Figure 4: Convergence of the maximal correlation error $\epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}}$ in regular case of $n = 100$ and $N = 120$.

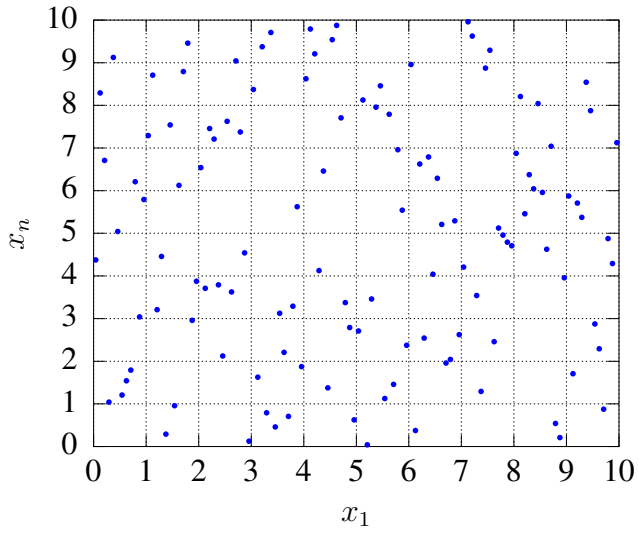

Figure 5: Anthill plot of the simulated variables x_1 and x_n in singular case of $n = 100$ and $N = 80$

calculate a low-rank approximation of the design matrix and perform the proper orthogonal decomposition in every iteration step. The initial error of plain latin hypercube sampling of $\epsilon_{\boldsymbol{\rho}_{\mathbf{HH}}} = 0.422$ is near the error of the Monte Carlo simulation. The adaptive procedure con-

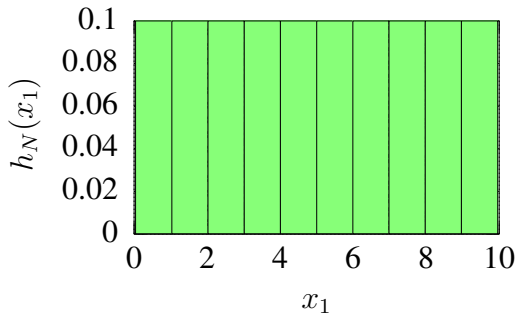

Figure 6: Histogram of the first variable x_1 in singular case of $n = 100$ and $N = 80$.

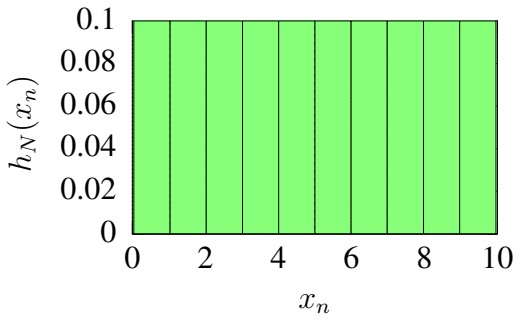

Figure 7: Histogram of the last variable x_n in singular case of $n = 100$ and $N = 80$.

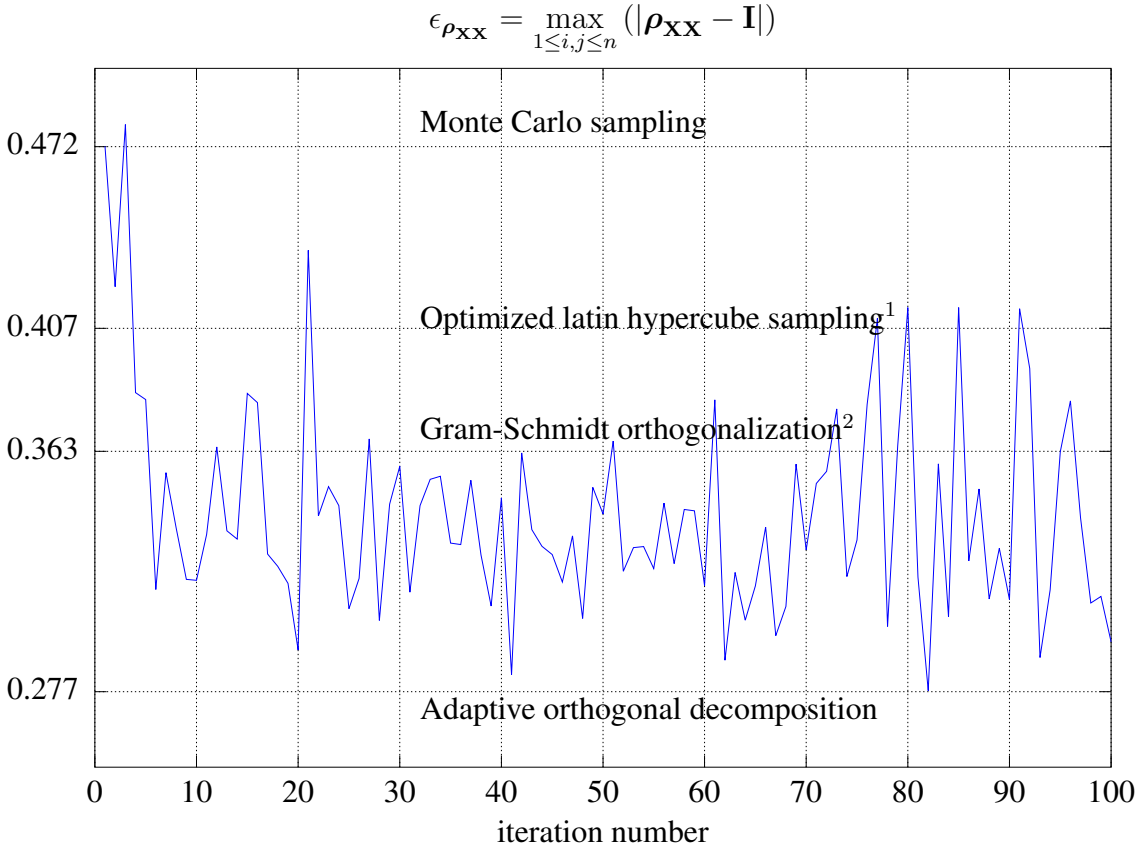

Figure 8: Convergence of the maximal correlation error $\epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}}$ in singular case of $n = 100$ and $N = 80$.

verged continuously to an error of $\epsilon_{\boldsymbol{\rho}_{\mathbf{XX}}} = 0.277$ after 100 iterations, as shown in Fig. (8). The anthill plot Fig. (5) and the histograms Fig. (6) and Fig. (7) of the first and last simulated variable shows the applicability of the proposed method.

5 SUMMARY

The proposed method of proper orthogonal decomposition of the covariance matrix and low-rank approximation of the design matrix using the singular-value decomposition in combination with an enhanced adaptive rearrangement of the columns or variables of the latin hypercube de-

sign matrix and an adaptive rank-order sample resorting of the variables is a fast an accurate alternative simulation procedure to sample uncorrelated variables with multidimensional uniformity. In contrast to the standard latin hypercube sampling, the Gram-Schmidt orthogonalization and a chosen optimization-based procedure we obtain a stable convergence of the correlation error to zero, even in high-dimensional problems. For the singular case, which means a smaller number of design evaluations as the number of variables, the propose method is more stable and efficient than other tested optimization-based methods or orthogonalization methods. This is very important for an efficient design of experiment in industrial applications of the variance-based sensitivity analysis and the metamodel-based optimization.

REFERENCES

[1] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(10):808–817, 2000.

[2] T.M. Cioppa and T.W. Lucas. Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics*, 49:45–55, 2007.

[3] K. Crombecq, E. Laermans, and T. Dhaene. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, 214(3):683 – 6965, 2011.

[4] J. L. Deutsch and C. V. Deutsch. Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference*, 142:763–772, 2012.

[5] Alexander I. J. Forrester, Andras Sobester, and Andy J. Keane. *Engineering Design via Surrogate Modelling - A Practical Guide.* Wiley, 2008.

[6] G. H. Golub and C. F Van Loan. *Matrix Computations*. The Johns Hopkins University Press, London, 1996.

[7] D. E. Huntington and C. S. Lyrintzis. Improvements to and limitations of latin hypercube sampling. *Probabilistic Engineering Mechanics*, 13(4):245 – 253, 1998.

[8] R. L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communication in Statistics*, B11(3):311–334, 1982.

[9] R. Jin, W. Chen, and A. Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134(1):268–287, 2005.

[10] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26:131–148, 1990.

[11] G. Kerschen, J. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear Dynamics*, 41(1):147–169, 2005.

[12] C.D. Lin, R. Mukerjee, and B. Tang. Construction of orthogonal and nearly orthogonal latin hypercubes. *Biometrika*, 96:243–247, 2009.

[13] Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381 – 402, 1995.

[14] A.B. Owen. Controlling correlations in latin hypercube samples. *Journal of the American Statistical Association*, 89(428):1517–1522, 1994.

[15] M.L. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.