Latin hypercube sampling with multidimensional uniformity

Jared L. Deutsch *, Clayton V. Deutsch
Centre for Computational Geostatistics (CCG), Department of Civil and Environmental Engineering, University of Alberta, 3-133 NREF Bldg. Edmonton, AB, Canada T6G 2W2

Abstract
Complex models can only be realized a limited number of times due to large computational requirements. Methods exist for generating input parameters for model realizations including Monte Carlo simulation (MCS) and Latin hypercube sampling (LHS). Recent algorithms such as maximinLHS seek to maximize the minimum distance between model inputs in the multivariate space. A novel extension of Latin hypercube sampling (LHSMDU) for multivariate models is developed here that increases the multidimensional uniformity of the input parameters through sequential realization elimination. Correlations are considered in the LHSMDU sampling matrix using a Cholesky decomposition of the correlation matrix. Computer code implementing the proposed algorithm supplements this article. A simulation study comparing MCS, LHS, maximinLHS and LHSMDU demonstrates that increased multidimensional uniformity can significantly improve realization efficiency and that LHSMDU is effective for large multivariate problems.

1. Introduction

Geoscientific models are often computationally expensive and involve numerous (i.e. 5–50) input variables. Each of the input variables follows a probability distribution from the sample data, physical models of the processes or expert judgement based on analogous settings. These models can only be realized a limited number of times due to constraints on computing time and power; even with computer clusters and ever-increasing CPU speed, the numerical models can be of higher resolution and include more details of the physical processes. Any method of generating input variables for these models must also account for correlations between variables; geoscientific variables are rarely independent of each other. Limited time and computing resources motivates the practitioner to use a method for generating input variables, which will provide an unbiased, accurate depiction of possible model outcomes.

The traditional technique for selecting input variables is Monte Carlo simulation (MCS) that randomly samples the cumulative distributions to obtain variable inputs. Another technique that has gained popularity is the Latin hypercube sampling (LHS), a technique emphasizing uniform sampling of the univariate distributions. LHS accomplishes this by stratifying the cumulative distribution function and randomly sampling within the strata. Uniform sampling increases realization efficiency while randomizing within the strata prevents the introduction of a bias and avoids the extreme value effect associated with regular stratified sampling. It has been demonstrated for many applications that LHS is a more efficient sampling method compared to MCS (McKay, 1979, 1992). The most recent methods seek to extend the notion of univariate uniformity to the multivariate. One method of increasing the multidimensional uniformity is through the use of minimax and maximin distance designs demonstrated by Johnson et al. (1990). The maximin designs seek to maximize the...
minimum statistical distance between model inputs, which are post-processed using the LHS algorithm for an efficient design with increased multidimensional uniformity. There are many other variants of Latin hypercube designs. Notable examples include orthogonal-array-based LHS designs (Leary et al., 2003), orthogonal-maximin LHS designs (Joseph and Hung, 2008) and orthogonal and nearly orthogonal designs (Bingham et al., 2009). The algorithm developed here is compared with MCS and LHS due to their historical significance and maximin-LHS due to its modern applicability.

Consider a model requiring \( N \) input variables that can be realized \( L \) times. The Monte Carlo approach is to generate \( L \times N \) uniform random numbers from \([0,1]\) and sample the cumulative distribution functions (CDFs) for each of the \( N \) variables using these random numbers. These sampled values would then be used as inputs for the realizations. From these \( L \) realizations, the probability distribution functions of the outputs will be estimated. This technique has been widely employed in all areas of science and engineering and forms the basis for many statistical methods.

The Latin hypercube approach, developed by McKay et al. (1979), for input generation is to stratify the CDF into \( L \) strata and draw uniform random numbers from each of the strata for the inputs. The reasoning for this stratification is that true randomness is not perfectly uniform; it is entirely possible that nearly identical values could be generated twice with MCS, which would be considered a waste of computing power. Moreover, with MCS it is possible that significant regions of the cumulative distribution function would not be sampled under the constraint of a limited number of realizations. Stratification of the univariate cumulative distribution functions with LHS enforces a degree of univariate uniformity but does not account for multivariate relationships.

Numerous modifications have been proposed for Latin hypercube sampling including those suggested by Iman and Conover (1982), Stein (1987), Johnson et al. (1990), Owen (1994), Cioppa and Lucas (2007), Lin et al. (2009) and others. The discussions by Stein, Iman, Conover and Owen provided methods for imposing correlations in a Latin hypercube matrix are very similar to the LU decomposition method used here. Lin et al. introduced a method for creating small orthogonal or nearly-orthogonal Latin hypercube matrices. These are perfectly uniform in the multivariate space, but are applicable only for small sets of variables and realizations and can be complicated to calculate. Cioppa and Lucas (2007) introduced a computationally expensive method for calculating nearly orthogonal Latin hypercubes. Their algorithm can calculate the best space-filling designs but requires long run times. The algorithm presented here does not necessarily calculate the best space-filling designs but is applicable to high dimensional problems, executes quickly and preserves the integrity of the LHS estimator.

The algorithm presented here increases the multidimensional uniformity of a sampling matrix by increasing the statistical distance between realizations. The most closely related modification of Latin hypercube sampling to the proposed algorithm is the maximinLHS algorithm proposed by Johnson et al. (1990). The maximinLHS implementation for \( R \) by Carnell (2009) was used for the comparative study in this article. This algorithm begins with a random starting point in the design matrix and a matrix of randomly generated locations in the Latin hypercube. The maximinLHS design is built by choosing the next point from the matrix of available locations with the maximum minimum inter-point distance from the points already included in the design matrix. The algorithm proceeds adding one point at a time until the design matrix has been generated. The resulting maximinLHS design is a Latin hypercube sampling matrix with increased multi-dimensional uniformity.

This article documents a modified LHS method emphasizing multidimensional uniformity, LHSMDU, which extends the idea of univariate uniformity used by LHS to a multivariate situation. This is accomplished using MCS to generate a large number of realization inputs and sequentially eliminating realizations that are near to each other in the multidimensional space. The distributed realizations are then post-processed to enforce univariate uniformity. However, emphasizing multidimensional uniformity is only desirable if the realization inputs respect correlations between the variables. This problem of conforming inputs to a correlation matrix is addressed with a Cholesky LU post-processing approach described by Tanabe and Sage (1992), Deutsch and Journel (1998) and others. Sampling matrices generated by MCS, LHS, maximinLHS and LHSMDU are compared to illustrate the differences between these methods. After the development of the LHSMDU algorithm, a synthetic case study is used to compare the improvements of LHS and LHSMDU over MCS. Finally, LHSMDU is compared with maximinLHS in a case study to demonstrate the effects of increasing dimensionality on the performance of these algorithms.

2. Algorithm for multidimensional uniformity

Consider a computer model requiring \( N \) variables that can be realized \( L \) times. The model inputs can be viewed as an input matrix with \( L \) rows and \( N \) columns (1). The Monte Carlo approach is to generate \( NL \) uniform random numbers in \([0,1]\) and arrange these values into an \( N \times L \) matrix. The cumulative distribution function for each of the \( N \) variables is then sampled using the uniform random numbers to form the \( L \times N \) sampling matrix in real variable units. For the duration of this article, the sampling matrix will only consist of numbers in the range of \([0,1]\). The transformation from \([0,1]\) to variable units by mapping with the cumulative distribution function of each variable is considered to come after the creation of the sampling matrix.

\[
\text{Sampling Matrix} = \begin{bmatrix}
F(x)_{11} & \cdots & F(x)_{1N} \\
\vdots & \ddots & \vdots \\
F(x)_{L1} & \cdots & F(x)_{LN}
\end{bmatrix}, \quad F(x)_{ij} \in [0,1]
\] (1)

The Latin hypercube approach is to generate \( NL \) uniform random numbers within specified ranges (2), which form the sampling strata. This ensures that each variable is uniformly sampled. The ordering of the \( L \) realizations is then randomized.
The cumulative distribution can then be sampled using these values to determine the input values:
\[ F(x) \in \left[ \frac{l-1}{L}, \frac{l}{L} \right], \quad l = 1, 2, \ldots, L \]  
(2)

Neither MCS nor LHS imposes a constraint on the multidimensional uniformity on the sampling matrix. To increase the multidimensional uniformity of the sampling matrix, a realization elimination algorithm is proposed. Using an elimination algorithm leads to a practical and robust algorithm that works for any dimension. Optimizing the space becomes computationally difficult for high dimensions; however, the distance calculation used to eliminate realizations in this algorithm scales linearly with dimension. The algorithm relies on calculating the Euclidean distance between realizations as a measure of how similar realizations are. This method is commonly used and can be found in many multivariate statistics textbooks (i.e. Johnson and Wichern, 2002). This distance \( D_{ij} \) is a measure of the redundancy or closeness of realizations \( i \) and \( j \) given by
\[
D_{ij} = \sqrt{\sum_{n=1}^{N} (F(x_{n,i}) - F(x_{n,j}))^2}
\]  
(3)

If the \( N \) variables are correlated with a known correlation matrix, the values can be correlated using the LU approach to enforce a correlation structure (Tanabe and Saga, 1992). This procedure is presented in brief here. To prevent artifacts from using linear combinations of variables, the input matrix is transformed to Gaussian units using a Gaussian inverse function. This is equivalent to sampling the CDF of the standard Gaussian distribution using the random numbers in the range of \([0,1]\). After transformation, the variables are correlated by multiplying each realization in \( L \) by the \( L \) matrix from the Cholesky LU decomposition of the correlation matrix. The correlated Gaussian variables are then normal score transformed to return them to the range of \([0,1]\). This is implemented in the proposed algorithm prior to applying LHS. The ability to generate designs that conform to a pre-specified correlation matrix is central to the applicability of this algorithm for the earth sciences. For example, oil content and water saturation often have a moderate negative correlation. The proposed algorithm for realization elimination and input correlation is given as follows:

Step 1: Generate \( N \times ML \) uniform random numbers in \([0,1]\) where \( M \) is a small number greater than 1 (\( M=5 \) was used in the case study presented; see Section 3). This is equivalent to considering \( I=ML \) realizations with \( N \) input variables.

Step 2: For each realization \( i \) in \( L \), calculate the Euclidean distance \( D_{ij} \) to all other realizations \( j \) in \( I \) with \( j \neq i \). Take the two smallest calculated distances (corresponding to the two nearest neighbors; see below for why two are used) and average them.

Step 3: Save the average distance corresponding to realization \( i \) and increment \( i \). Return to Step 2 until the average distance has been calculated for all \( I \) realizations.

Step 4: Eliminate the realization with the smallest average distance calculated in Step 2 (hence \( I=I-1 \)).

Step 5: Return to Step 2 until there are only \( I=L \) realizations.

Step 6: If the variables are dependent, then proceed with Step 6a, otherwise proceed to Step 7.

Step 6a: Transform \( N \times L \) values to Gaussian units using the standard Gaussian CDF.

Step 6b: Multiply each realization \( i \) in \( L \) by the \( L \) matrix from the Cholesky LU decomposition of the correlation matrix.

Step 6c: Back-transform each of the \( N \times L \) values to the \([0,1]\) uniform distribution using the standard Gaussian CDF.

Step 7: For variable \( n \) in \( N \), rank the \( L \) inputs and use these rankings as the strata \( l \).

Step 8: Generate uniform random numbers for the \( L \) strata for variable \( n \) according to (2).

Step 9: Sample the cumulative distribution function of \( n \) with the uniform random numbers generated in Step 8. Increment \( n \) and return to Step 7 until this ranking and drawing has been accomplished for each of the \( N \) variables. This is equivalent to applying Latin hypercube for each of the \( n \) variables with the ordering determined by Steps 2–5 instead of randomly.

The algorithm proposed uses the LHS approach with the goal of improving the multivariate space sampled by eliminating redundant realizations from the initial pool of \( ML \) realizations. Imposing LHS on the marginal distributions of the multivariate enforces univariate uniformity without significantly modifying the multivariate distribution of the realizations. As shown by McKay et al. (1979), LHS is an unbiased estimator; this approach preserves the integrity of the LHSMDU estimator.

The two nearest neighbors are considered (Step 2) because this is the minimum number of neighbors that can be considered. The redundancy of a realization can be defined by only considering the distance of a realization to its nearest neighbor. However, if only the nearest neighbor was considered then this would result in realization pairs being generated; the decision of which realization to eliminate could not be easily made. The second neighbor is considered only to break the tie between two close realizations. The possibility to consider greater than two realizations could be implemented, however the authors believe that no extra information on the redundancy of realizations would be gained by considering more than two nearest neighbors.

A graphical comparison of MCS, LHS, maximinLHS and LHSMDU for two independent variables is presented in Fig. 1. A similar plot is shown for two variables correlated with \( \rho=0.85 \) in Fig. 2 (maximinLHS was not used for this plot as a
correlation scheme for maximinLHS was not implemented in the R package used. Correlations were imposed using the LU method described. Both Figs. 1 and 2 are plots of four input matrices of twenty realizations each for a bivariate problem. For this illustration, the variables are assumed independent. It can be seen that MCS does not uniformly sample either the

Fig. 1. Comparison of realization sets generated by MCS, LHS and LHSMDU algorithms.

Fig. 2. Comparison of realization sets using post-processing to enforce a correlation of $\rho = 0.85$. 
marginal or multivariate space. LHS and LHSMDU both uniformly sample the univariate distributions but only maximinLHS and LHSMDU approximately sample the multivariate distribution uniformly.

This algorithm is implemented in the Fortran program lhsmdu (Supplementary materials). Given values for \( N \), \( L \), \( M \) and a correlation matrix, it will generate sampling matrices (1) using MCS, LHS and LHSMDU. The pseudo-random number generator used is the ACORNI generator described by Wikramaratna (1990). Any reputable random number generator could be used, but Ortiz and Deutsch (2001) tested the ACORNI pseudo-random number generator and found that it acceptably passed a number of tests for randomness.

3. Case study

The described algorithm for enhancing multidimensional uniformity was developed for geoscientific computer models that generally have a large number of variables and can only be realized a limited number of times due to model complexity. Testing the relative power of MCS, LHS, maximinLHS and LHSMDU using a realistically complex model would be unreasonable as the true probability distribution of outputs could not be easily calculated. For this reason, a simplified original oil in place problem was chosen as the primary case study. The problem is calculating the original oil in place, \( \text{OOIP} \), for a reservoir with the given thickness of the deposit \( T \), deposit area \( A \), net oil to gross volume NTG, net porosity \( \Phi_{\text{net}} \) and water saturation \( S_w \). Given these values, the OOIP is given by

\[
\text{OOIP} = CAT \times \text{NTG} \times \Phi_{\text{net}} (1 - S_w)
\]

(4)

where \( C \) is a constant to account for units (for this case study, \( C = 1 \)). Simple parametric distributions were chosen for the OOIP parameters. These are summarized in Table 1.

To judge the relative power of the algorithms for generating a sampling matrix, a statistic similar to the Kolmogorov–Smirnov \( D \) statistic was used. This value, \( e \), is illustrated in Fig. 3 and given by

\[
e = \max | F_{\text{est}}(p) - F_{\text{ref}}(p) |, \quad \text{for } p = 0.1, 0.2, \ldots, 0.9
\]

(5)

Using the Kolmogorov–Smirnov \( D \) statistic would require interpolating or fitting sample CDFs, which would be difficult for low realization numbers. Using the nine deciles to compare values forgoes any interpolation while allowing comparison of the deviations from the true distribution for MCS, LHS, maximinLHS and LHSMDU. As no method used showed a systematic bias, the maximum deviation from the true distribution was chosen for the value of \( e \). The value \( e \) is not unitless like the \( D \) statistic, but is useful for comparing the four sampling methods.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution</th>
<th>Distribution parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>Triangular</td>
<td>( a = 2, b = 4, c = 6 )</td>
</tr>
<tr>
<td>( T )</td>
<td>Gaussian</td>
<td>( m = 10, \sigma = 1 )</td>
</tr>
<tr>
<td>NTG</td>
<td>Uniform</td>
<td>( a = 0.6, b = 0.8 )</td>
</tr>
<tr>
<td>( \Phi_{\text{net}} )</td>
<td>Truncated Triangular</td>
<td>( a = 0.15, b = 0.25, c = 0.35 )</td>
</tr>
<tr>
<td>( S_w )</td>
<td>Triangular</td>
<td>( a = 0.15, b = 0.2, c = 0.3 )</td>
</tr>
</tbody>
</table>

Fig. 3. Illustration of \( e \) value calculation. The calculation is based on the maximum deviation of the sample CDF from the true CDF at each of the 9 deciles (0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9).
3.1. Choice of M

Consider first the situation where the variables are independent. The true distribution of OOIP can be calculated using a very large Monte Carlo simulation study. This was done using 10 million realizations for a mean OOIP value of 5.90 and standard deviation of 1.73. Using the LHSMDU algorithm requires the choice of a value for $M$. While a very large $M$ value could be used, calculating the inter-realization distances would be time consuming so values between 1 and 15 were tested. Note that using an $M$ value of unity would correspond to strictly using LHS with no increase in multidimensional uniformity from the MDU algorithm as no realizations would be eliminated. Median $e$ values were calculated as a function of $M$ using 5000 sets of 100 realizations (Fig. 4). Improvements from the MDU algorithm showed an asymptotic relationship with no increase in effectiveness for $M$ values greater than 5. No increase in effectiveness for $M$ greater than 5 implies that at $M=5$, the variables were very well distributed with a high degree of multidimensional uniformity. With this choice of $M$, the realizations were distributed with a high degree of multidimensional uniformity as illustrated by the performance of the LHSMDU algorithm in this case study. The algorithm performs well independently of the number of realizations and dimension of the sampling matrix. For the duration of this article, the initialization factor $M$ was chosen to be 5 as no improvement was seen for greater values.

3.2. Comparison when considering independent variables

Consider using only 100 realizations and calculating the sample CDF of OOIP from these realizations for each of MCS, LHS and LHSMDU. A cumulative probability plot of the $e$ value obtained using the three techniques under these conditions is plotted in Fig. 5. Enforcing univariate uniformity resulted in significantly improved reproduction of the true OOIP probability distribution (decreasing $e$). By increasing the multidimensional uniformity with the LHSMDU algorithm, reproduction of the true OOIP distribution was further increased as LHSMDU had the lowest median $e$ value.

The median $e$ value was calculated for different numbers of realizations ranging from $L=10$ to $L=10,000$ (Fig. 6). As expected, reproduction of the true OOIP CDF improved for all techniques as the number of realizations increased. The curves fit to each series have the following form:

$$e_{median} = at^b, \quad b \approx -0.5$$

(6)

The coefficient of $b$ in Eq. (6) was approximately $-0.5$ for all techniques, a relationship frequently found in statistics. A more intuitive way to understand the improvements by LHS and LHSMDU is shown in Fig. 7. Here, the equivalent
number of Monte Carlo realizations that would be required to have the same median $e$ value as the technique in question is plotted as a function of the number of realizations. The MCS line (a 1:1 relationship) is plotted for reference.

3.3. Comparison when considering correlated variables

The same OOIP simulation study was repeated, only this time a specified correlation matrix (7) was used. For this correlation matrix, variable 1 is $A$, variable 2 is $T$ and so on. The correlations were enforced on all sampling matrices using the LU method previously described. The median $e$ value as a function of the number of realizations is again plotted for the three sampling methods (Fig. 8) and equivalent Monte Carlo realizations (Fig. 9). The equations fit to the realized values have the same form as Eq. 6 with $b \approx 0.5$ for each method.

$$
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0.3 & 0.25 & -0.4 \\
0 & 0.3 & 1 & 0.4 & -0.5 \\
0 & 0.25 & 0.4 & 1 & -0.6 \\
0 & -0.4 & -0.5 & -0.6 & 1 \\
\end{bmatrix}
$$

(7)

The correlated simulation study showed that LHS and LHSMDU improved on MCS comparable to the improvements for the uncorrelated OOIP study. Imposing a correlation structure did not significantly affect the relative performance of either LHS or LHSMDU relative to MCS.
4. Case study with maximinLHS

The OOIP case study discussed compared the improvements of LHS and LHSMDU over MCS, but did not include recent modifications of the LHS algorithm such as maximinLHS (Johnson et al., 1990) to simplify the original case study. Both maximinLHS and LHSMDU improve on the multivariate uniformity of a sampling matrix, so any comparison between the two algorithms must consider both low and high dimensional multivariate cases. In this section, maximinLHS and LHSMDU are compared using a low dimensional (bivariate) and high dimensional (5-variate) case study.

4.1. Comparison of maximinLHS and LHSMDU for a bivariate case study

Consider modeling only the volume of the deposit for the previous OOIP example. This volume is a simple bivariate function of the area and thickness of the deposit, \( V = AT \). For this case study, parametric distributions similar to the previously used \( A \) and \( T \) variables were used (Table 2). The only change was the shift in \( A \) from a symmetric triangular distribution to an asymmetric triangular distribution.

The relative power of the four algorithms for generating a sampling matrix was compared using the median \( e \) value as a function of the number of realizations \( L \) (Fig. 10). As before, an initialization factor of \( M = 5 \) was used for LHSMDU. As observed in the prior OOIP case study, increasing the multidimensional uniformity of the input sampling matrix improved reproduction of the true output distribution. The performance of maximinLHS is comparable to LHSMDU; both methods improve on MCS and LHS as methods for generating bivariate sampling matrices. For larger numbers of realizations, maximinLHS outperforms LHSMDU, although both methods are efficient at reproducing the true volume CDF compared to LHS and MCS.
4.2. Comparison of maximinLHS and LHSMDU for a 5-variate case study

The same 5-variate OOIP simulation study used to compare MCS, LHS and LHSMDU was repeated including maximinLHS as one of the methods to generate a sampling matrix. The resulting plot of median $e$ value against the number of realizations is shown below (Fig. 11). For this high dimensional case study, there was no demonstrated improvement of maximinLHS over LHS. This observation contrasts with the low dimensional case study in which maximinLHS significantly improved over LHS and improved over LHSMDU for higher values of $L$. This indicates that increasing the number of variables in a problem can reduce the effectiveness of maximinLHS relative to LHSMDU; with the 5 dimensional problem presented, maximinLHS did not greatly improve on LHS.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution</th>
<th>Distribution parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Triangular</td>
<td>$a = 2$, $b = 3$, $c = 6$</td>
</tr>
<tr>
<td>$T$</td>
<td>Gaussian</td>
<td>$m = 10$, $\sigma = 1$</td>
</tr>
</tbody>
</table>

Fig. 10. Median $e$ value as a function of the number of realizations for the bivariate volume case study.

Fig. 11. Median $e$ value as a function of the number of realizations for the OOIP case study including maximinLHS.

4.2. Comparison of maximinLHS and LHSMDU for a 5-variate case study

The same 5-variate OOIP simulation study used to compare MCS, LHS and LHSMDU was repeated including maximinLHS as one of the methods to generate a sampling matrix. The resulting plot of median $e$ value against the number of realizations is shown below (Fig. 11). For this high dimensional case study, there was no demonstrated improvement of maximinLHS over LHS. This observation contrasts with the low dimensional case study in which maximinLHS significantly improved over LHS and improved over LHSMDU for higher values of $L$. This indicates that increasing the number of variables in a problem can reduce the effectiveness of maximinLHS relative to LHSMDU; with the 5 dimensional problem presented, maximinLHS did not greatly improve on LHS.
4.3. Impact of dimensionality on performance of maximinLHS and LHSMDU

The improvements of maximinLHS over MCS and LHS in the bivariate case were significant; however these improvements were not realized for the higher dimensional OOIP case study. We suggest that the decrease in efficiency of maximinLHS over LHSMDU in the higher dimensional OOIP case study is related to the exponential number of points that are required to effectively sample highly multidimensional space (the curse of dimensionality). The maximinLHS algorithm sequentially adds realization to a Latin hypercube design beginning from a single random input. The number of candidate points considered for a given iteration is equal to the difference between the specified number of realizations \( L \) and the realizations that have already been included in the design matrix. LHSMDU sequentially eliminates realizations from a larger pool of starting points (equal to \( ML \)) that were generated randomly. After eliminating realizations so that only \( L \) remain, the points are post-processed to conform them to a Latin hypercube design. For a high dimensional problem the realization addition approach implemented by maximinLHS imposes a significant limitation on the volume of an \( N \)-dimensional unit hypercube that the remaining candidate points can sample. This differs from LHSMDU, which can cover a larger volume of the highly multidimensional space by starting with a larger pool of realizations. The LHSMDU algorithm can be extended to any dimension; the calculation of distances in the determination of the optimal design scales linearly with dimension. The number of candidate designs also scales linearly with a slope of \( M \) and the problem dimension.

5. Conclusions

Four sampling techniques have been discussed for generating an \( L \times N \) input matrix for computationally expensive models: MCS, LHS, maximinLHS and LHSMDU. MCS is the classical random method for sampling while LHS, maximinLHS and LHSMDU all produce uniform univariate inputs for a better than random input model. Using multidimensional uniform inputs, which is an extension of the univariate uniformity enforced by LHS, shows great potential for improving model inputs as implemented in the LHSMDU algorithm.

For the OOIP simulation study considered, LHSMDU demonstrated significant improvements over maximinLHS, LHS and MCS. The maximinLHS algorithm greatly improved on LHS and MCS in low dimensions with results comparable to LHSMDU, however in the higher dimension case the study performed similar to LHS. The improvement is attributed to the greater degree of multidimensional uniformity achieved by the LHSMDU sampling matrix. Correlation structures were imposed using a \( LU \) decomposition of the correlation matrix. The improvements of LHSMDU were not adversely affected by imposing a correlation structure on the inputs. The elimination algorithm presented is robust in that the calculations necessary are simple to implement and scale linearly with dimension. Improvements of LHSMDU over other algorithms do not diminish with increasing dimension or by imposing a correlation matrix on the design as demonstrated in the case studies presented. Further research possibilities include the possibility of inserting realizations into a design and investigating the non-uniqueness of the results due to the randomness of the initial realizations.

Appendix A. Supplementary materials

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jspi.2011.09.016.

References


