Latin Hypercube Sampling with Multidimensional Uniformity

Jared L. Deutsch and Clayton V. Deutsch

Complex geostatistical models can only be realized a limited number of times due to large computational requirements. Many methods exist for generating input parameters for model realizations including Monte Carlo simulation (MCS) and Latin hypercube sampling (LHS). An extension of Latin hypercube sampling (LHSMDU) for multivariate models with limited realizations is presented which enforces multidimensional uniformity of the input parameters through sequential realization elimination. A lhsmdu program is shown that will calculate an L x N input matrix for N variables and L realizations. A simulation study comparing MCS, LHS and LHSMDU demonstrates that LHSMDU can significantly improve realization efficiency. Correlations are imposed on the sampling matrix using a Cholesky LU decomposition of the correlation matrix.

Introduction

Geostatisticians frequently deal with computationally expensive models involving numerous input variables each distributed according to a calculated distribution. The traditional technique for selecting realization input variables is Monte Carlo simulation (MCS) which randomly samples the cumulative distributions to obtain variable inputs. Another technique which has gained popularity is Latin hypercube sampling (LHS), a technique that emphasizes uniformly sampling the variables by stratifying the distribution function. It has been demonstrated for many applications that LHS is a much more efficient method compared to MCS (McKay 1979, 1992). We propose a modified LHS method, LHSMDU, that enforces multidimensional uniformity. This method extends the goal of univariate uniformity used by LHS to a multivariate situation. A method for conforming the inputs to a correlation matrix using LU post-processing is also implemented.

Background

Consider a computer model requiring N input variables that can be realized L times. The Monte Carlo approach would be to generate LxN 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. MCS was first named and used extensively by von Neumann and Ulam in their work at Los Alamos Labs during the 1940s (Kochanski, 2005). This technique has been widely employed by 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 possible that significant regions of the cumulative distribution function would not be sampled if only a limited number of inputs could be realized. The stratification enforces univariate uniformity but does not account for multivariate relationships. LHS was demonstrated by McKay (1979, 1992) to, on average, be better than MCS for the selection of input variables and to be an unbiased estimator.

Modifications to the Latin hypercube presented by McKay *et al.* have been described by Iman and Conover (1982), Owen (1994), Lin *et al.* (2009) and others. The discussion by Iman, Conover and Owen providing a method for imposing correlations in a Latin hypercube matrix is very similar to the method used in this paper. Lin *et al.* introduced a method for creating small orthogonal or nearly-orthogonal Latin hypercube matrices. Their idea of enforcing multivariate uniformity is key.

Methods

Suppose a geostatistical model requires N variables and can be realized L times. The model inputs can be viewed as an input matrix with L rows and N columns (Equation 1). The Monte Carlo approach would be to generate NL uniform random numbers in the range of 0 to 1 and arrange these values into an N by L

matrix. The cumulative distribution function for each of the *N* variables is then sampled using the random numbers.

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

The Latin hypercube approach would be to generate *NL* uniform random numbers with a range given by Equation 2. This ensures that each variable is uniformly sampled. The ordering of the *L* realizations is then randomized. The cumulative distribution is sampled using these values to determine the input variables.

$$F(x) \in \left[\frac{l-1}{L}, \frac{l}{L}\right], l = 1, 2, ..., L \tag{2}$$

Neither MCS nor LHS impose a degree of multidimensional uniformity on the sampling matrix. To enforce a degree multidimensional uniformity, a realization elimination algorithm is proposed. Initially, *ML* realizations are generated where *M* is a small number greater than 1 making the initial sampling matrix *M* times larger than the final. Each of the *ML* realizations is composed of *N* uniform random numbers in the range of 0 to 1. To eliminate realizations, the average Euclidean distance of each realization to its two nearest neighbors is calculated (Equation 3). The two nearest neighbors instead of the single nearest neighbor are taken since this prevents realization pairs from being generated.

$$D_{i,j} = \sqrt{\sum_{n=1}^{N} \left(F(x)_{n,i} - F(x)_{n,j}\right)^2} \tag{3}$$

The realization with the smallest average distance to its two nearest neighbors is eliminated. This distance calculation and elimination scheme is repeated until only *L* realizations remain. If the variables are assumed independent, then the realizations are ranked and the same stratification scheme implemented by LHS is imposed on the samples. This enforces univariate uniformity without significantly modifying the multivariate distribution of the realizations.

If necessary, a correlation structure can be imposed before applying Latin hypercube using a method similar to that of Iman, Conover and Owen. The LU approach to enforce a correlation structure is detailed by Deutsch (1998) but 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. After transformation, the variables are correlated by multiplying each realization 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].

A comparison of the three techniques discussed, MCS, LHS and LHSMDU is presented in Figure 1. This figure is a plot 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 marginal or multivariate space. LHS and LHSMDU both uniformly sample the marginal distributions but only LHSMDU approximately samples the multivariate uniformly.

Implementation

The GSLIB compatible program *lhsmdu* implements all of the techniques discussed. Given values for *N, L, M* and a correlation matrix, it will generate sampling matrices in the form of Equation 1 using MCS, LHS and LHSMDU. The pseudo-random number generator used is the ACORNI generator described by Wikramaratna (1990). Ortiz and Deutsch (2001) tested the ACORNI pseudo-random number generator and found that it acceptably passed all tests for randomness. The parameter file and specific details on the *lhsmdu* program are contained in Appendix 1.

A small simulation study to test the relative power of the MCS, LHS and LHSMDU algorithms was set up using an original oil in place (OOIP) example. For this 5 variable problem, the variables were assumed to be independent. Probability distribution functions for each variable and a full description are given in Appendix 2. The OOIP was calculated using Equation 4 letting C=1 for simplicity. Unless otherwise noted, the default values used in this study were *L*=100 realizations and an initialization factor of *M*=5.

$$\mathrm{OOIP} = \mathrm{C}\bullet\mathrm{A}\bullet\mathrm{T}\bullet\mathrm{NTG}\bullet\Phi_{net}\bullet(1-\mathrm{S}_w) \tag{4}$$

The "true" OOIP probability distribution function was easily calculated using 10 million MCS realizations which was compared with the sample cumulative distribution functions (CDFs) generated in this study. The effectiveness of each technique was judged using a value similar to the Kolmogorov-Smirnov D statistic. This value, e, is given by Equation 5 and illustrated in Figure 2.

$$e = \max\left|F_{ref}^{-1}(p) - F_{est}^{-1}(p)\right|, \text{ for } p = 0.1, 0.2, ..., 0.9 \tag{5}$$

A cumulative histogram of e values for MCS, LHS and LHSMDU techniques was generated using 5000 sets of 100 realizations (Figure 3). From this plot it can be seen that LHSMDU significantly improves over LHS which in turn is a significant improvement over MCS. All of the histograms are bounded on the lower end by 0 resulting in skewed distributions.

The median e value was calculated for different numbers of realizations ranging from *L*=10 to *L*=10000 (Figure 4). As expected, all techniques improved as the number of realizations increased. The curves fit to each series have the form of Equation 6.

$$e_{median} = a\bullet L^b, \;\; b \cong -0.5 \tag{6}$$

The coefficient of *b* in Equation 6 was approximately -0.5 for all techniques. MCS had a value slightly lower, LHS was almost exactly -0.5 and LHSMDU had a slightly higher coefficient. This relationship frequently occurs in statistics, such as the standard deviation for a Gaussian variable. A more intuitive way to understand the improvements by LHS and LHSMDU is shown in Figure 5. Here, the equivalent number of Monte Carlo realizations that would be required to have the same median e value is plotted as a function of the number of realizations if MCS, LHS or LHSMDU was used. The MCS line (a 1:1 relationship) is plotted for reference.

To determine the optimal initialization factor, *M*, for the MDU algorithm, a simulation of e values as a function of *M* was conducted (Figure 6). This was found to have an asymptotic relationship as the e value for *M*=5 was almost the same as for *M*=10 and *M*=15. As such, the default setting recommended for the *lhsmdu* program is *M*=5. Note that if *M*=1 then no realizations are eliminated so the LHSMDU algorithm is the same as LHS.

Correlated variables are handled by LHSMDU using the LU method previously described. After correlating, the Latin hypercube stratification scheme is applied to enforce univariate uniformity. A comparison of realization sets generated by MCS, LHS and LHSMDU for ρ = 0.85 in the same form as Figure 1 is shown below (Figure 7).

The same OOIP simulation study was repeated, only this time a specified correlation matrix (Equation 7) was used. For this correlation matrix, variable 1 is A, variable 2 is T and so on. The median e value as a function of the number of realizations is again plotted for the three sampling methods (Figure 8). The equations fit to the realized values have the same form as Equation 6 with b ≈ 0.5 for each method.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.3 & 0.25 & -0.4 \\ 0 & 0.3 & 1 & 0.4 & -0.5 \\ 0 & 0.25 & 0.4 & 1 & -0.6 \\ 0 & -0.4 & -0.5 & -0.6 & 1 \end{bmatrix} \tag{7}$$

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 affect the relative performance of either LHS or LHSMDU relative to MCS.

Conclusions

Three sampling techniques have been discussed for generating an *L* x *N* input matrix for a computationally expensive geostatistical model: MCS, LHS and LHSMDU. MCS is the classical random method for sampling the model while LHS and LHSMDU are both marginally stratified to achieve univariate uniformity in an attempt to produce a better than random input model. The concept of multidimensional uniformity, which is an extension of the univariate uniformity enforced by LHS, shows potential for improving model inputs as implemented in the LHSMDU algorithm. The program *lhsmdu* implements a realization elimination scheme to impose a degree of multidimensional uniformity.

For the small OOIP simulation study in this note, LHSMDU demonstrated significant improvements over LHS and MCS. 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 affected by imposing a correlation structure.

Limitations and Future Work

LHSMDU suffers from the same principle problem as LHS. Both techniques require the user to know the number of realizations they want to run prior to running the realizations. This prevents the implementation of an early stopping criterion and does not make allowances for running additional realizations with ease. MCS does not have these conditions, however, as suggested by the small study conducted, will require many more total realizations compared to LHS or LHSMDU to achieve the same accuracy.

Future work would see the implementation of all three techniques into a larger scale study with many input variables. Ideally this study would be non-trivial but not so computationally demanding that only a small number of realizations could be run. LHS and LHSMDU could then be graded based on their relative efficiency compared to MCS.

References

Deutsch, C.V. and Journel, A.G., 1998, *GSLIB: Geostatistical Software Library and User's Guide,* Oxford University Press, New York, 2nd Ed., pp 142-143.

Iman, R.L. and Conover, W.J., 1982, A distribution-free approach to inducing rank correlation among input variables, *Communications in Statistics: Simulation and Computation*, 11(3), pp 311-334.

Johnson, R.A. and Wichern, D.W., 2002, *Applied Multivariate Statistical Analysis,* Prentice-Hall, New Jersey, 5th Ed., pp 183-190.

Kochanski, G., Monte Carlo Simulation, 2005, http://kochanski.org/gpk/teaching/0401Oxford/, accessed July 14, 2009

Lin, C.D., Mukerjee, R. and Tang, B., 2009, Construction of orthogonal and nearly orthogonal Latin hypercubes, *Biometrika*, 96(1), pp 243-247.

McKay, M.D., Beckman, R.J. and Conover, W.J., 1979, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, 21, pp 239-245.

McKay, M.D., 1992, Latin hypercube sampling as a tool in uncertainty analysis of computer models, *Proceedings of the 1992 Winter Simulation Conference*, pp 557-564.

Ortiz, J.C. and Deutsch, C.V., 2001, Testing Pseudo-Random Number Generators, *Third Annual Report of the Centre for Computational Geostatistics*.

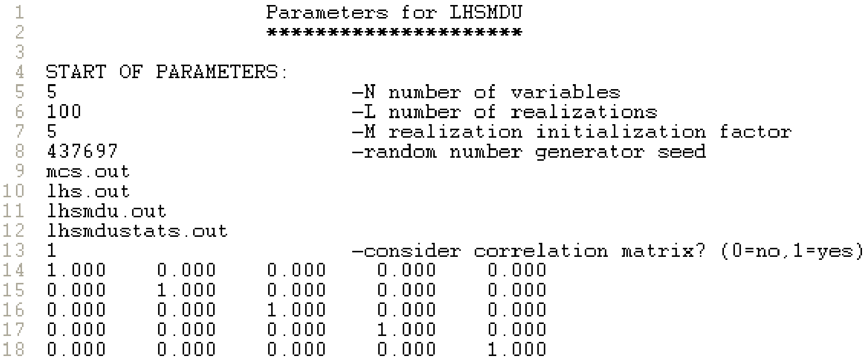
Owevn, A.B., 1994, Controlling Correlations in Latin Hypercube Samples, *Journal of the American Statistical Association*, 89(428), pp 1517-1522.

Sprent, P. and Smeeton, N.C., 2007, *Applied Nonparametric Statistical Methods*, Chapman & Hall/CRC texts in statistical science series, 4th ed., pp 83-90.

Wikramaratna, R.S., 1990, Theoretical analysis of the acorn random number generator, *SIAM Conference on Applied Probability in Science and Engineering*.

Appendix 1

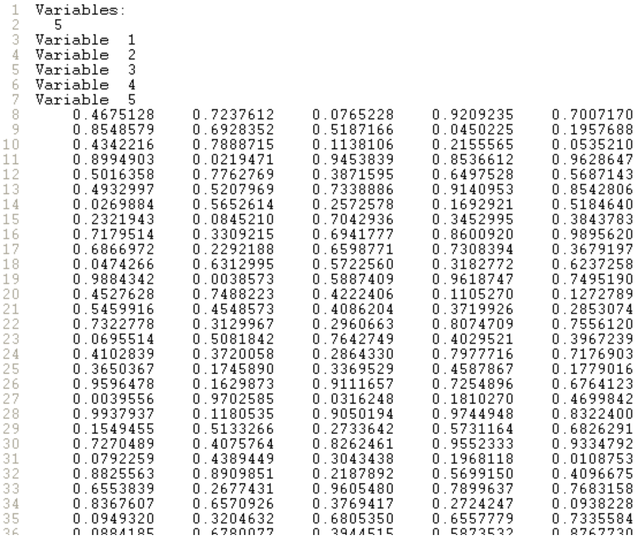
The *lhsmdu* program follows standard GSLIB convention in that it requires no direct user input but references a parameter file, an example of which is given below.



```
 1                     Parameters for LHSMDU
 2                     *********************
 3
 4  START OF PARAMETERS:
 5  5                           -N number of variables
 6  100                         -L number of realizations
 7  5                           -M realization initialization factor
 8  437697                      -random number generator seed
 9  mcs.out
10  lhs.out
11  lhsmdu.out
12  lhsmdustats.out
13  1                           -consider correlation matrix? (0=no,1=yes)
14  1.000    0.000    0.000    0.000    0.000
15  0.000    1.000    0.000    0.000    0.000
16  0.000    0.000    1.000    0.000    0.000
17  0.000    0.000    0.000    1.000    0.000
18  0.000    0.000    0.000    0.000    1.000
```

The user specifies the number of variables, number of realizations and *M* discussed in this paper in lines 5 through 7. A random number generator seed for the ACORNI generator and output file names for the matrices are specified in lines 8-11. Line 12 allows the user to specify a file for output of the entropy of the *lhsmdu* design matrix. To enable this output, the user should consult the documentation in the Fortran code of *lhsmdu* and recompile as instructed. If the user wishes to consider correlated variables, line 13 is set to 1, otherwise the variables are assumed independent. The correlation matrix for the *N* variables is then specified if desired.

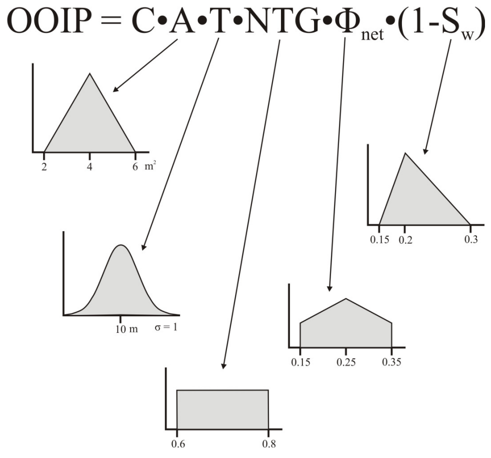
Each of the MCS, LHS and LHSMDU sampling matrices are output in the format shown below which can easily be read by any GSLIB compatible software.



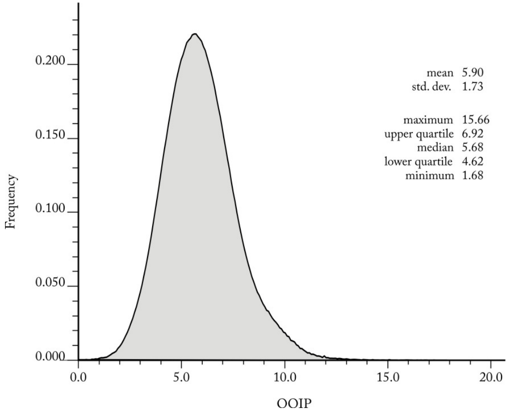
```
Variables:
  5
Variable  1
Variable  2
Variable  3
Variable  4
Variable  5
     0.4675128     0.7237612     0.0765228     0.9209235     0.7007170
     0.8548579     0.6928352     0.5187166     0.0450225     0.1957688
     0.4342216     0.7888715     0.1138106     0.2155565     0.0535210
     0.8994903     0.0219471     0.9453839     0.8536612     0.9628647
     0.5016358     0.7762769     0.3871595     0.6497528     0.5687143
     0.4932997     0.5207969     0.7338886     0.9140953     0.8542806
     0.0269884     0.5652614     0.2572578     0.1692921     0.5184640
     0.2321943     0.0845210     0.7042936     0.3452995     0.3843783
     0.7179514     0.3309215     0.6941777     0.8600920     0.9895620
     0.6866972     0.2292188     0.6598771     0.7308394     0.3679197
     0.0474266     0.6312995     0.5722560     0.3182772     0.6237258
     0.9884342     0.0038573     0.5887409     0.9618747     0.7495190
     0.4527628     0.7488223     0.4222406     0.1105270     0.1272789
     0.5459916     0.4548573     0.4086204     0.3719926     0.2853074
     0.7322778     0.3129967     0.2960663     0.8074709     0.7556120
     0.0695514     0.5081842     0.7642749     0.4029521     0.3967239
     0.4102839     0.3720058     0.2864330     0.7977716     0.7176903
     0.3650367     0.1745890     0.3369529     0.4587867     0.1779016
     0.9596478     0.1629873     0.9111657     0.7254896     0.6764123
     0.0039556     0.9702585     0.0316248     0.1810270     0.4699842
     0.9937937     0.1180535     0.9050194     0.9744948     0.8322400
     0.1549455     0.5133266     0.2733642     0.5731164     0.6826291
     0.7270489     0.4075764     0.8262461     0.9552333     0.9334792
     0.0792259     0.4389449     0.3043438     0.1968118     0.0108753
     0.8825563     0.8909851     0.2187892     0.5699150     0.4096675
     0.6553839     0.2677431     0.9605480     0.7899637     0.7683158
     0.8367607     0.6570926     0.3769417     0.2724247     0.0938228
     0.0949320     0.3204632     0.6805350     0.6557779     0.7335584
     0.0884185     0.6780077     0.3944515     0.5873532     0.8767730
```

Appendix 2

The original oil in place (OOIP) simulation shown in this note used the following reference histograms for the drawing of input values.



Letting C=1 m^{-3} gives the OOIP distribution obtained below using 10 million realizations.



The deciles used in the calculation of e values were drawn from this distribution. The correlated OOIP distribution was also calculated (not shown) and the deciles drawn in an identical manner.

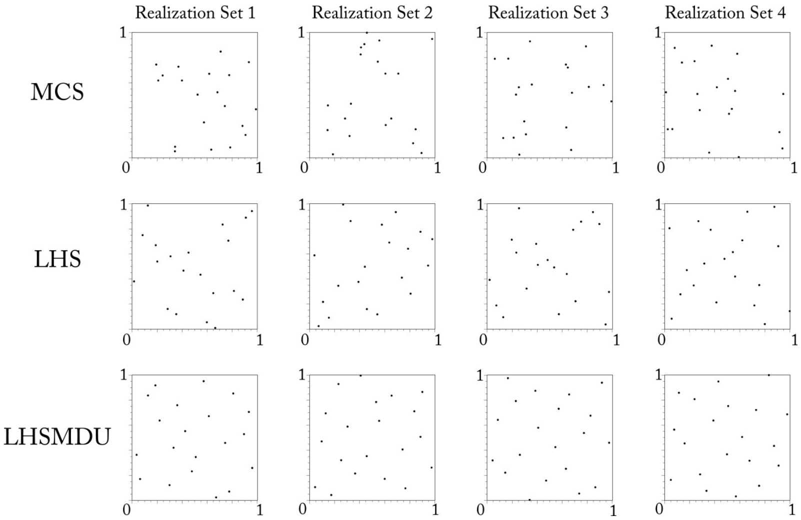


Figure 1: Comparison of realization sets generated by MCS, LHS and LHSMDU algorithms.

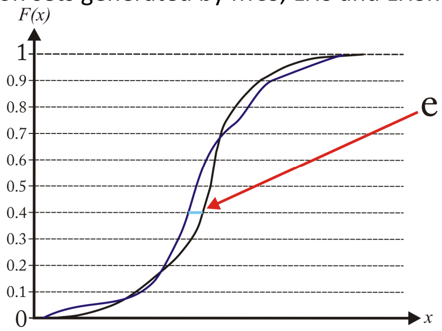


Figure 2: Illustration of e value calculated based on maximum deviation of the sample CDF from the true CDF at each of the 9 deciles.

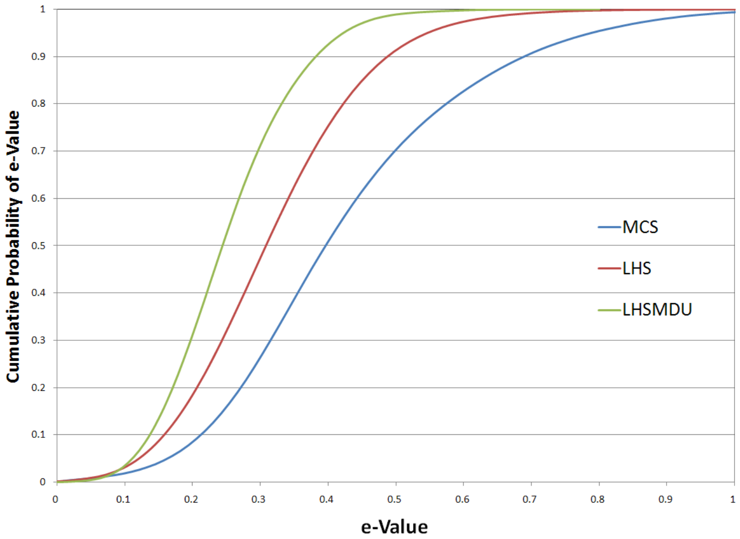


Figure 3: Cumulative probability plots of e-values from OOIP study for different sampling techniques.

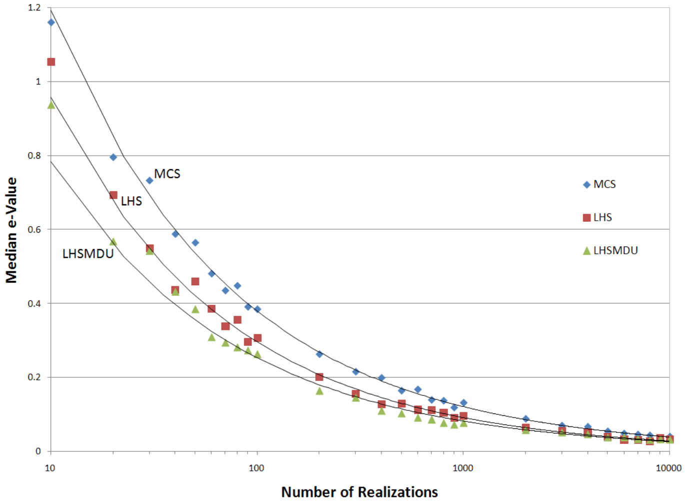


Figure 4: Median simulated e values as a function of the number of realizations

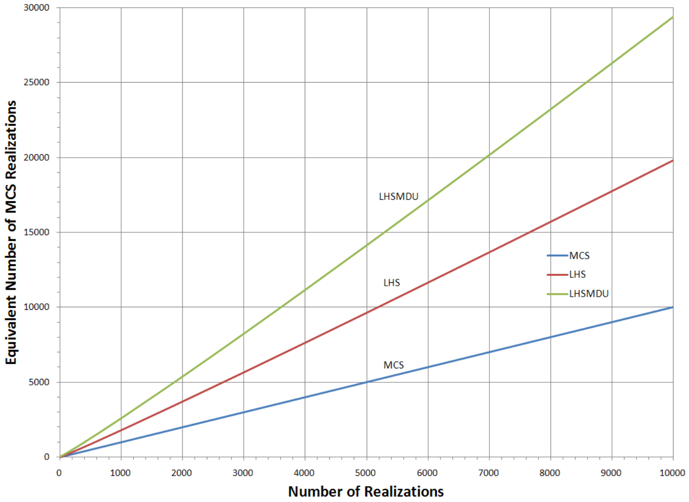


Figure 5: Equivalent number of MCS realizations to have the same median e value as a number of realizations using one of the techniques studied

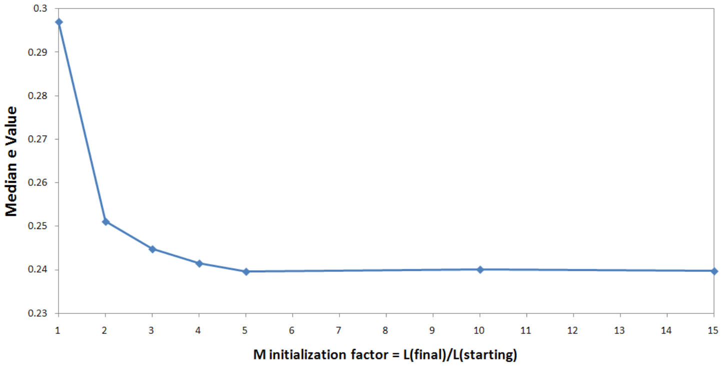


Figure 6: Median e value from LHSMDU as a function of different M values.

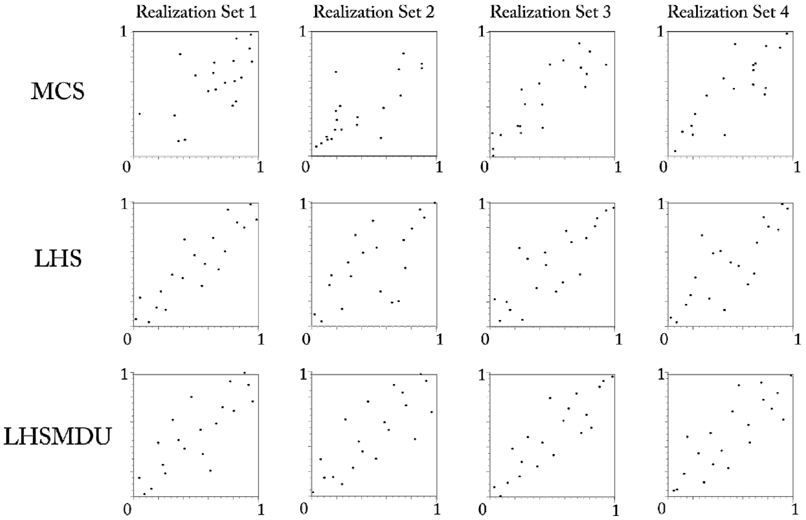


Figure 7: Comparison of realization sets generated by MCS, LHS and LHSMDU with ρ = 0.85

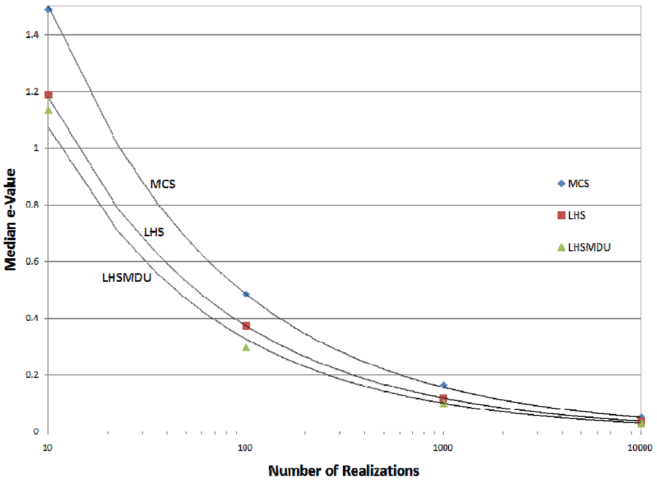


Figure 8: Median e values as a function of the number of realizations for correlated OOIP simulation