Improvements to and Limitations of Latin Hypercube Sampling
Seminar: Uncertainty Quantification in Machine Learning

Name: Mridul Varghese Koshy
Enrolment No.: 230196
Faculty of Statistics
Technical University Dortmund
July 29, 2022 Summer Term 2022/23

Abstract

The report aims to discuss two methods that can enhance the performance of Latin Hypercube sampling. The conventional approach of taking samples for each variable from the cumulative distribution is replaced by finding the probabilistic means of equiprobable disjunct intervals in the variable's domain. Instead of doing matrix manipulation to reduce the correlation between variables(correlated and uncorrelated), a single-switch-optimized step for one variable at a time is proposed. Improvements and limitations achieved by the two new methods introduced, and experimental results from a Poisson process are further discussed.

1. Introduction

Numerous statistical studies have utilized and continue to employ the Monte Carlo method or Monte Carlo simulation as an analytical tool. Monte Carlo simulation is a powerful tool to analyze random phenomena using computers. It relies on repeated random sampling to produce numerical results. The idea is to use randomness to solve problems that, in theory, may be deterministic. Monte Carlo simulation is simple and easy to use. It is most useful when other approaches are difficult or impossible to use. The random inputs variables in Monte Carlo are expressed in several deterministic set of numbers known as samples, realization, or observation. The random problem is now divided into smaller deterministic problems which are easy to solve. The samples generated by Monte Carlo as a result, will provide accurate statistical and probabilistic information about the desired variable.

The crux of Monte Carlo simulation is the sampling scheme. The simplest sampling scheme technique uses a pseudorandom generator which generates random samples based on a known transformation. The samples are generally drawn randomly from the cumulative distribution of the input variables. This straightforward sampling scheme has a significant flaw as it requires large samples to represent the input distribution precisely. When simulating a single Gaussian variable using this sampling scheme, 133,000 samples are needed to guarantee a 99% probability that the sample variance is within 1% of the true value.

Other sampling schemes such as Shinozuka or auto-regressive moving average(or ARMA for short) have been developed to tackle the large sample size requirement for specific cases. Although these techniques provide good samples which match the target mean and correlation precisely when temporal averaging of the simulated process is used, they cannot be used for non-Gaussian or non-stationary, or several independent non-Gaussian variables[5].

Latin hypercube sampling is considered as one of the best sampling techniques for Monte Carlo simulation. Latin Hypercube sampling is a stratified random process that provides an efficient way of sampling variables from their distributions[4]. Unlike simple random sampling scheme, this method ensures full preservation of marginal probability of the simulated variables and the correlation between each variable by maximally stratifying each marginal distribution. Only a small number of samples are required for good accuracy in the response parameter. This is achieved by creating a highly joint probability density function for each of the variables in the random problem.

Latin Hypercube sampling has two stages. First, the values are chosen from a cumulative distribution to represent the variables' probability density function. The samples of the variables are then ordered in such a way that they match the target correlation between the variables. The marginal probability of the variables remain unaltered as ordering is done for matching correlation rather than changing the values. The improvement suggested by authors (D. E. Huntington and C. S. Lyrintzis) in these two steps improves the samples by ensuring a higher correlation between the variables to the target values and better representation of the probability distribution of each variable. This method improves the result especially when the correlation of the variable is nonzero.

The improvement suggested in the first stage of generating sample is to take the probabilistic mean of each section or intervals from the variables' domain. This is in contrast to the normal approach of taking samples directly from the cumulative distribution function. The second improvement suggested is to reorder the samples generated by using a single-switch-optimizer rather than the conventional approach of using matrix manipulation to achieve target correlation between the variable. The single-switch-optimizer hence orders the variable one at a time rather than ordering all variables in one step in the conventional manner. The results obtained by the two improvements are discussed with respect to existing techniques. The strength and limitation of the improvements to Latin Hypercube sampling suggested here are analyzed using a simple Poisson process. Improvements related to the updated Latin Hypercube sampling to enable non-positive ordering matrix and achieving better statistical accuracy are mentioned in this report. Further, future improvements are also discussed in this report.

2. Sample Generation

The first stage of Latin Hypercube can be summarized as:

- divide the cumulative distribution of each variable into N equiprobable intervals
- from each interval, select a value randomly, for the ith interval

The samples for each variable generated will represent the variable's probability distribution.

$$x_{i,k} = {F_i}^{-1}\left(\frac{k-0.5}{N}\right), \tag{1}$$

where ${F_i}^{-1}$ is the inverse cumulative distribution function for variable X_i, N is the number of samples per variable, and $x_{i,k}$ is the kth sample of the ith variable of X_i. The major drawback of this approach is the lack of matching target correlation, although mean is fairly close to the desired one.

The authors suggest a method which would help improve the variables' marginal probability density function. This is achieved by taking the random mean of each section as the section's sample from the probability density function.

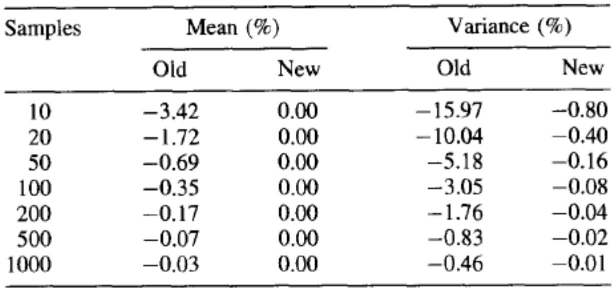
$$x_{i,k} = \frac{1}{N}\int_{y_{i,k-1}}^{y_{i,k}} x f_i(x)dx, \tag{2}$$

where f_i is the probability density function of the variable X_i. The limits of the integration can be found out by using eqn (3).

$$y_{i,k} = {F_i}^{-1}\left(\frac{k}{N}\right), \tag{3}$$

Using eqn (2), we might not be able to solve some probability distribution in its closed form. It is worthwhile to put forth the additional effort required to perform the necessary numerical integration.

Figure 1. Percentage error in mean and variance of an exponentially distributed variable for the current(old) and proposed(new) sampling schemes using different numbers of samples.



Samples	Mean (%)		Variance (%)	
	Old	New	Old	New
10	−3.42	0.00	−15.97	−0.80
20	−1.72	0.00	−10.04	−0.40
50	−0.69	0.00	−5.18	−0.16
100	−0.35	0.00	−3.05	−0.08
200	−0.17	0.00	−1.76	−0.04
500	−0.07	0.00	−0.83	−0.02
1000	−0.03	0.00	−0.46	−0.01

Fig (1) shows a table that compares the difference in statistical accuracy between the old and new sampling schemes. Various number of samples are simulated for an exponentially distributed variable using both the techniques. The probability density function of the variable is given by $f(x) = e^{-x}H(X)$, where $H(X)$ is the Heaviside unit step function, and e is the base of natural logarithms. Mean and variance of the random variable is 1. Comparing the values from the table shown in fig (1), we can see that the errors in the simulated mean for the old technique (taking directly from the cumulative distribution function) is low, whereas the error in the simulated mean while using the new or proposed technique is 0. There is a noticeable difference for the errors in simulated variance of the variable while using the proposed method. In order to get the simulated variance to be accurate within 1%, 500 samples were required when using the old technique where as only ten samples were required in the proposed new sampling scheme.

The new sampling scheme can be used along with the old scheme since the values generated by the samples are nearly identical *everywhere except at the tail* of the distribution. The old, simpler sampling scheme should be effective for replicating a distribution with finite domain. The new scheme only needs to generate the first and last several samples; the remaining samples can be obtained using the old method if the probability distribution for the relevant variable has an infinite or semi-infinite domain or if the numerical integration in the proposed scheme becomes challenging.

3. Sample Ordering

The correlations between variables, whether they are zero or not, must be taken into consideration after the samples for each variable have been generated. Samples for each variable generated are ordered using Latin Hypercube. The nth response value is produced when doing function evaluations using the nth ordered sample for each variable. In order to maintain the marginal probability density functions for each variable, the target correlations between variables can be matched with the appropriate ordering scheme.

Random ordering of generated samples for each variable was considered sufficient when the variables were uncorrelated. Florian, in his work has proved that this method would generate significant correlation between some of the variables. Hence to reduce the random correlation between the generated variables, Florian proposed an updated Latin Hypercube sampling scheme[1]. Florian proposed the following ideas in his updated Latin Hypercube sampling.

In the updated Latin Hypercube, instead of using the samples themselves for ordering, Florian proposed to use rank number. The value of k used in eqn (1) or in eqn (2) or in eqn (3) is taken as the rank number. An ordering matrix R of size $N \times M$ is formed with columns containing permutation of rank numbers for each variable. The simulation proceeds as the following once the ordering matrix R is formed: Samples for each variable are arranged on the same column of the ordering matrix based on the rank numbers for the variable. The nth reordered sample of each is then used to produce the nth required output.

It must be ensured that no two columns are alike in the ordering matrix, even though the permutation of rank numbers in it are generated randomly. Spearman correlation coefficients defines the correlation between columns in the ordering matrix.

$$T_{i,j} = 1 - \frac{6 \sum\limits_{k} (R_{ki} - R_{kj})^2}{N(N-1)(N+1)}, \tag{4}$$

where T_{ij} represents the Spearman correlation coefficient between variables i and j. The Spearman correlation coefficient ranges from -1 to 1.

Unless some columns in the matrix have identical ordering, the correlation matrix T will be symmetric and positive definite. To ease further calculations, Cholesky decomposition of matirx T can be performed:

$$T = Q^T . Q \tag{5}$$

A pseudo-ordering matrix R_B is generated at this point:

$$R_B = R . Q^{-1} \tag{6}$$

The rank numbers are then put in each column of the ordering matrix R in the same order as the corresponding column of RB. It has been demonstrated that applying this strategy may cause correlations between any two system variables to decline rather than increase. It should be noted that this method can be used repeatedly, possibly allowing for extremely low correlations. The authors have discovered that updated Latin hypercube sampling actually exhibits a tendency to converge to an ordering that nevertheless produces considerable correlation errors between some variables. Therefore, for uncorrelated variables, a different ordering strategy must be utilized.

The current technique has greater challenges while simulating correlated variables. To generate 'uncorrelated' variables, Latin Hypercube sampling must be done repeatedly. Cholesky decomposition is done on the matrix T of the target correlation coefficient as in eqn (5). Using eqn (6) we generate the new ordering matrix. There is no method to iterate the correlation procedure done using eqn (5) and eqn (6). Consequently, nothing can be done to solve the issue if the new ordering matrix does not enable the variables to match the intended correlation coefficients well. It is essential to devise a new ordering scheme that can accurately represent both correlated and uncorrelated data.

The authors propose a single-switch-optimized sample ordering scheme to better simulate both correlated and uncorrelated variables. The proposed scheme orders modified sample rather than using rank numbers or Spearmans coefficient. Spearmans coefficient correlation between two variables fails when the variables are not uniformly distributed. Hence the updated Latin Hypercube approach has to be modified. For this, a new matrix with unordered samples is formed:

$$R_{ij} = \frac{x_{j,i} - \mu_j}{\sigma_j} \tag{7}$$

where σ_j is the standard deviation for that variable and μ_j is the mean for the jth variable. Each column of the transformed matrix R now has a zero-mean and unit variance variable.

Each column of matrix R is subjected to the following method one at a time beginning with the second column. Let's say the first $m-1$ columns were subjected to the ordering technique. The actual correlation coefficients between the mth column and each of the preceding ones are first calculated into the vector T. This is formed using:

$$T_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}R_{im} \tag{8}$$

where $1 \leq j \leq m-1$. The correlation coefficient error E is defined as:

$$E = \sum_{j=1}^{m-1} (T_j - T'_{jm})^2 \tag{9}$$

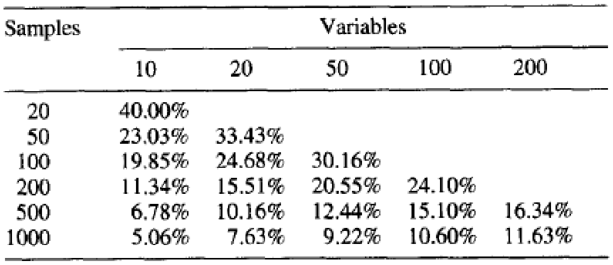
where T' is matrix containing target correlation coefficients. The change in E that would happen if a pair of samples in the mth variable were switched is computed for each pair of samples. The samples that resulted in the highest decrease in E are then exchanged. The mth variable is subjected to this process repeatedly until the point at which either there is no room for improvement or the correlation coefficients are all accurate to within a predetermined threshold. The method is then applied, one variable at a time, to all remaining variables. Since just one switch is used at a time to optimize sample ordering, the method is known as single-switch optimization[2].

Finally, the ordered samples can be written in a matrix S when the reordering method is finished, converting the original unordered modified sample matrix R into an ordered modified sample matrix R', as shown here:

$$S_{ij} = R'_{ij}\sigma_j + \mu_j \tag{10}$$

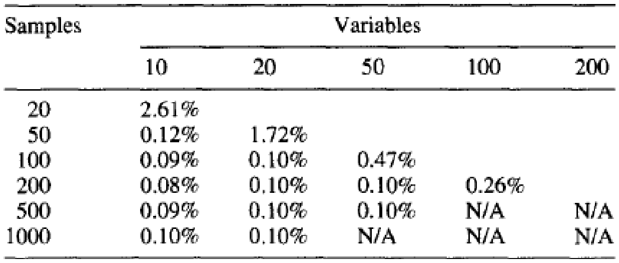
Figure 2 and 3 are tables that show the difference in performance between the updated Latin Hypercube sampling method and the proposed single-switch-optimized method. Uncorrelated exponential variables were simulated using varying amounts of samples and variables for both sampling schemes. Actual correlation coefficients were used for the proposed method while Spearman coefficient correlation was used for the updated Latin Hypercube sampling. The greatest Spearman coefficient(in absolute value) was recorded for each of the 50 runs of Florian's revised Latin hypercube sampling method. The procedure was iterated 50 times in total, with each run lasting until there was no longer any change in the ordering matrix. The minimum of these numbers was then kept, and is shown in Figure 2. Maximum correlation coefficients from the proposed technique with a threshold of 0.1% are shown in Figure 3.

Figure 2. Minimum over 50 runs of maximum Spearman coefficient using updated Latin hypercube ordering for various numbers of variables and samples.



Samples	Variables				
	10	20	50	100	200
20	40.00%				
50	23.03%	33.43%			
100	19.85%	24.68%	30.16%		
200	11.34%	15.51%	20.55%	24.10%	
500	6.78%	10.16%	12.44%	15.10%	16.34%
1000	5.06%	7.63%	9.22%	10.60%	11.63%

Figure 3. Maximum correlation coefficient using proposed single-switch optimization ordering for various numbers of variables and samples; variables are exponentially distributed. The ordering scheme was halted when the maximum correlation coefficient was 0.1% or lower, or when no further improvement was possible.



Samples	Variables				
	10	20	50	100	200
20	2.61%				
50	0.12%	1.72%			
100	0.09%	0.10%	0.47%		
200	0.08%	0.10%	0.10%	0.26%	
500	0.09%	0.10%	0.10%	N/A	N/A
1000	0.10%	0.10%	N/A	N/A	N/A

It is evident from the figures that the proposed technique provides much higher correlation accuracy across all the variables than the updated Latin hypercube sampling method. Both ordering techniques' accuracy are largely influenced by the N/M ratio, which measures the number of samples to variables. For updated Latin hypercube sampling, N/M must be greater than 100 to guarantee that the maximum

Figure 4. Average computation time (s) required for updated Latin hypercube ordering on an IBM 486DX4/120 desktop computer.

Samples	Variables				
	10	20	50	100	200
20	4.1832				
50	5.92E−01	6.25E−01			
100	3.98E−01	4.53E−01	1.7996		
200	4.89E−01	6.47E−01	1.7902	9.2092	
500	1.1328	1.9686	4.8796	15.8472	90.5822
1000	3.6842	7.125	18.3132	39.3108	163.2448

Figure 5. Computation time (s) required for proposed single-switch optimization ordering on an IBM 486DX4/120 desktop computer. The ordering scheme was halted when the maximum correlation coefficient was 0.1% or lower. For reference, 1 day is 86400s.

Samples	Variables				
	10	20	50	100	200
20	2.70E−01				
50	2.03	14.17			
100	6.26	41.96	954.82		
200	39.33	212.67	2543.87	29 900.73	
500	560.57	2848.6	22084.94	N/A	N/A
1000	4410.35	21292.2	N/A	N/A	N/A

Spearman coefficient is less than 5%. The proposed method yields a maximum correlation coefficient of less than 1% for a ratio N/M of 2 to 5. The reason for this, as previously explained, is that updated Latin hypercube sampling converges to an inappropriate ordering. The suggested method performs sufficiently, even if it may possibly converge to a non-optimal ordering.

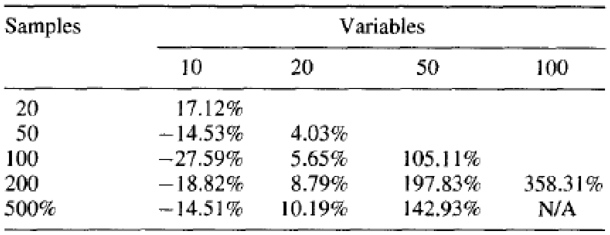
Additionally, computation times for the two reordering methods on an IBM 486 DX4, 120 desktop computer were determined. This is displayed as tables in fig (4) and Fig (5) respectively. Figure 4 shows the average computational time required by the updated Latin Hypercube for various number of variables and sample size. Likewise, Figure 5 shows the average computational time required by the proposed method for various number of variables and sample sizes. From the tables, it is evident that the updated Latin Hypercube technique is much faster compared to the proposed technique. For a simulation with 200-variables with 1000-samples, the updated Latin hypercube sampling took less than 3 minutes of time. The proposed technique is quite time consuming with larger cases taking several hours or days. The entry denoted by N/A on the table took more than a day.

The computation time required by the proposed method can be reduced roughly by approximately a factor of N, if there is enough memory to store the changes in each correlation coefficient change caused by each sample pair switch. This is because only a small subset of those numbers would need to be updated following a switch. This was not done for any of the simulation done by the authors. The computation time can further be decreased if the threshold value for maximum correlation coefficient error is set higher resulting in a lower accuracy statistical output.

Figure 6. Percentage error in skewness of sum of various number of simulated independent exponential variables, with various number of samples.

Samples	Variables				
	10	20	50	100	200
20	40.00%				
50	23.03%	33.43%			
100	19.85%	24.68%	30.16%		
200	11.34%	15.51%	20.55%	24.10%	
500	6.78%	10.16%	12.44%	15.10%	16.34%
1000	5.06%	7.63%	9.22%	10.60%	11.63%

Figure 7. Percentage error in kurtosis of sum of various number of simulated independent exponential variables, with various number of samples.



Samples	Variables			
	10	20	50	100
20	17.12%			
50	−14.53%	4.03%		
100	−27.59%	5.65%	105.11%	
200	−18.82%	8.79%	197.83%	358.31%
500%	−14.51%	10.19%	142.93%	N/A

4. Simulating a Poisson process

A Poisson process is used to test the powers of proposed Latin Hypercube sampling. The proposed Latin Hypercube sampling method was used to simulate various uncorrelated exponential variables with different sample size. The outcomes can then be used to simulate the sum of ”independent” identically-distributed exponential variables, which should be a gamma-distributed variable. The mean μ, variance σ^2, kurtosis k, and skewness s of the gamma distribution are given below :

$$\begin{aligned} \mu &= E[x] = n/\lambda \\ \sigma^2 &= E[(x-\mu)^2] = n/\lambda^2 \\ s &= E[(x-\mu)^3]/\sigma^3 = 2/\sqrt{n} \\ k &= E[(x-\mu)^4]/\sigma^4 = 3 + 6/n \end{aligned} \tag{11}$$

where λ represents the parameter in the exponential distribution for each variable which is taken to be unity in the computations here, x is a dummy variable denoting a value of the gamma-distributed variable, n denotes the number of exponential variables being added together, and $E[\ ]$ is the expected value.

As mentioned above, the simulated mean for each variable will have 0% error as simulated mean will be exactly the same as the actual mean. This can be verified from Figure 1. The simulated variance will also be close to the actual variance (again, see Figure 1). From Figure 3, we can see that the simulated correlation between the variables are near to zero. We look at the skewness and kurtosis of the sum of the simulated variables and compare them to the target values in eqn (11) to determine how successfully Latin hypercube sampling replicates independent variables. Figure 6 and 7 shows the percentage of error in skewness and kurtosis of sum of various numbers of simulated independent variables, with various number of samples. From Figure 6 and 7, we can see that the error in skewness and kurtosis of the simulated variables are high. The error is skewness tends to increase as the number of variables increases. Although less predictable, the error in kurtosis again seems to be mostly dependent on the number of variables. No definite relationship between the variable size and error in skewness or error in kurtosis can be established. It is to be noted, more samples are not always going to produce more accurate results in Latin hypercube sampling after a sufficient number of samples have been picked for accuracy in variable distributions and correlations.

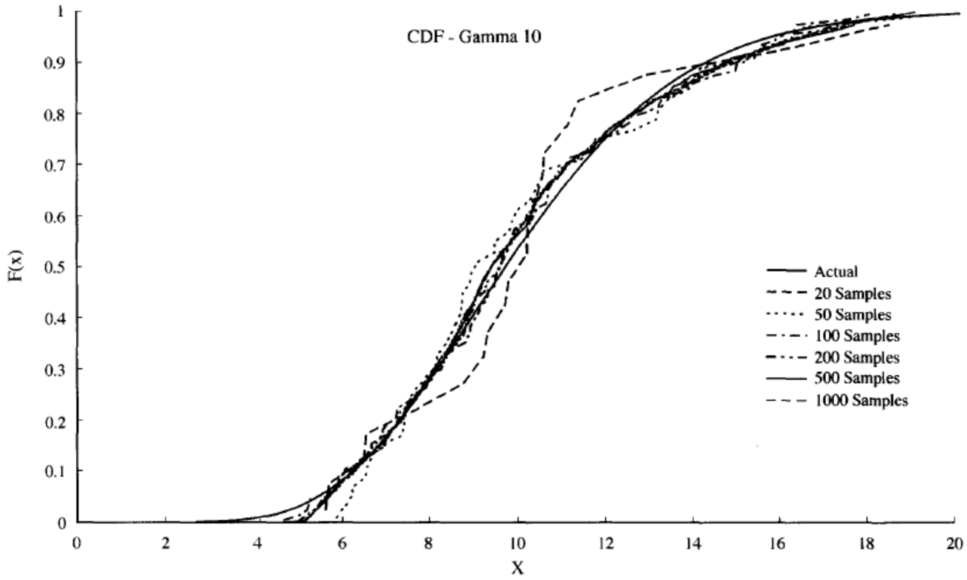
Latin hypercube sampling can match higher order moments inside each variable, which are represented by the variable’s marginal probability distribution, but it can only match second-order random moments(variances) between variables. Although independent variables cannot be simulated using Latin hypercube sampling, uncorrelated variables can be. Therefore, unless a small number of variables become dominating, which the Poisson process does not produce, the preservation of the marginal distributions for the variables is of limited use. Higher-order random moments between variables might be incorporated by modifying the error term in eqn (9), but doing so would need significantly more memory, computing time, and samples per variable than the suggested method for good accuracy.

Latin Hypercube can generate an approximate cumulative distribution function(or CDF) from the samples generated. For this, the outputs of the Latin Hypercube are sorted in ascending order. This can be used as the x-values and can be placed at equal intervals along the y-axis ranging from 0 to 1. Figure 8 and 9 show charts that compares actual CDF to the approximate ones derived from the outputs. The CDF generated from an output of 10 and 100 variables are seen in figure 8 and 9 respectively. From these figures, it is evident that the generated CDF closely matches to that of the actual CDF. Slight variation from the actual CDF can be seen at the tail of the distribution. Although the curve improves with the increase in samples, after a point more samples would not yield a better curve as the generated CDF converges to a single limiting curve. This can be predicted because of the error in skewness and kurtosis explained above.

5. Related Works

The updated Latin Hypercube sampling scheme uses Cholesky decomposition in its permutation as explained above in eqn (5). In this approach, the Latin Hypercube tends to converge to an ordering but will still produce

Figure 8. Actual CDF for the sum of 10 independent exponential variables, with parameter $\lambda = 1$, and approximate CDF for various numbers of samples.



significant correlation error between the variables. Also a positive definite correlation matrix is required for the approach. The single-switch-optimized sample ordering scheme proposed by the authors for better accuracy in correlation between the simulated variables comes with a lot of computational time. Although the proposed method achieves better statistical accuracy, compared to the updated Latin Hypercube sampling, it is far too slow. Hence a modification in updated Latin Hypercube step to incorporate for non-positive definite correlation matrix yielding higher statistical output and lower computational time can be used as alternative to the proposed single-switch-optimizer.

Three modified algorithms are proposed that can improve the stability of the Latin Hypercube sampling when a non-positive definite correlation matrix case arises[6].

5.1. Hypersphere Decomposition

The hypersphere decomposition(or Hd) can be used to convert the matrix into a semi-positive definite matrix. One way to think of hypersphere decomposition is as an iterative procedure to change the existing defined matrix to the desired correlation matrix. P is a given non-positive target matrix, $\hat{P}$ is the desired matrix that is closest to P. $\hat{P}$ can be constructed as follows:

$$\hat{P} = Q.Q^T \tag{12}$$

$$x_{i1} = \begin{cases} \cos\theta_{ij}.\prod_{t=1}^{j-1}\sin\theta_{it} & \text{for} \, j = \mathit{1.. .. .. n\text{-}1}, \\ \prod_{t=1}^{j-1}\sin\theta_{it} & \mathit{for\, j = n}\,, \end{cases} \tag{13}$$

θ_{ij} is an arbitrary set of $n \times (n-1)$ dimensional angles. A suitable error measure can help $\hat{P}$ to approach P and can be defined as follows:

$$\epsilon_a = \|P - \hat{P}\| \tag{14}$$

To identify the matrix that most closely matches the target matrix, optimization algorithms can be used based on the target eqn (14). The correctness of the output, admittedly, comes at the sacrifice of time.

5.2. Spectral Decomposition

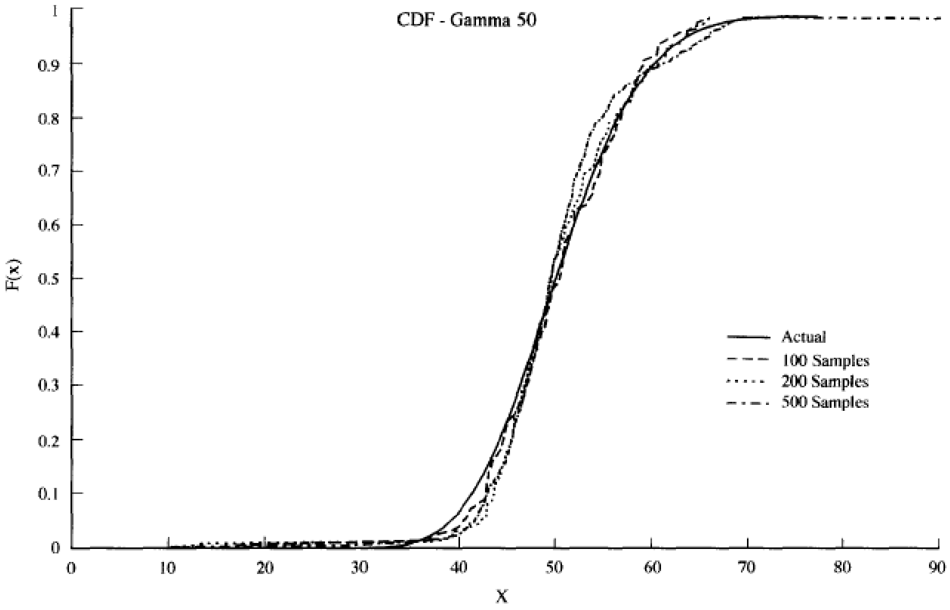
Spectral decomposition(or Sd) is an empirical method without iterating and we can always get a correlation matrix modified well. The modification step is:

1. P is a correlation matrix, λ_i, L_i are its corresponding eigenvalues and eigenvectors.

$$\begin{aligned} PL_i &= \lambda_i L_i \\ L_i &= [l_{i1}, l_{i2}, ..., l_{in}]^T \end{aligned} \tag{15}$$

2. A positive matrix Λ' is modified from Λ which is

Figure 9. Actual CDF for the sum of 50 independent exponential variables, with parameter $\lambda = 1$, and approximate CDF for various numbers of samples.



the diagonal matrix of eigenvalues.

$$\Lambda = diag\lambda_i$$

$$\Lambda' : \lambda'_i = \begin{cases} \lambda_i & \lambda_i > 0, \\ \epsilon_b & \lambda_i \leq 0, \end{cases} \tag{16}$$

3. A diagonal scaling matrix D can be formed from the eigenvectors Li as:

$$D : d_i = \left[\sum_{t=1}^{n} l_{it}^2 \lambda'_t \right]^{-1} \tag{17}$$

4. The columns of Q' is formed by multiplying the eigenvectors with their corresponding modified eigenvalues. Q is the normalized form of Q'. $\hat{P}$ is the corrected correlation matrix constructed by Q.

$$\begin{aligned} Q' &= L\sqrt{\Lambda'}, \quad Q = \sqrt{D}Q' \\ \hat{P} &= Q.Q^T \end{aligned} \tag{18}$$

Intuitively, the acquired matrix is exactly like the targeted one. However, the simplicity of computation and quick speed of this method are its advantages.

5.3. Modified alternating projections method

Modified alternating projections method(or Mapm) modifies the principle of the alternating projections method. It combines advantages of Hypersphere decomposition and Spectral decomposition, which possess high precision as well as fast speed. Alternating projections method is applied to find the nearest matrix. The alternating projections method can be summarized as eqn (19).

$$\begin{aligned} X &= P_U(P_S(P_U(P_S(P)))) \rightarrow P \\ P &= ZAZ^T, \quad A = diag(vi) \end{aligned} \tag{19}$$

where

$$P_U(P) = P - diag(diag(P - I))$$

$$P_S(P) = Z \times diag(max(v_i, 0)) \times Z^T$$

5.4. Experimental Results

Data simulation is performed in order to compare the effect of these three algorithms applied to Latin Hypercube sampling. Three algorithms are compared based on their correcting effect(Figure 10) and computational time(Figure 11). The correcting

Figure 10. Correcting error of three algorithms under different dimensions.

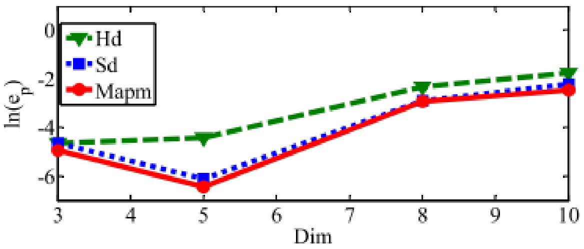
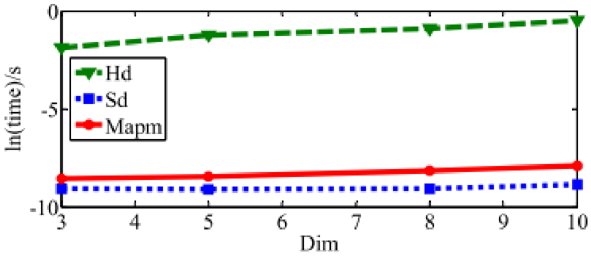


Figure 11. Correcting error of three algorithms under different dimensions.



effect is calculated by:

$$e_p = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(p_{ij} - \hat{p}_{ij})^2}{n^2}} \tag{20}$$

where P is the original non-positive correlation matrix and $\hat{P}$ is the modified matrix. p_{ij}, $\hat{p}_{ij}$ are the elements of P and $\hat{P}$.

The correctness of Hypersphere decomposition or Hd is largely influenced by the optimization algorithms. In this test case as seen in fig (11), Hd takes the longest time. For example, the simulation time of Hd is 0.4023s, the time of Sd is 0.0001s, and time for Mapm is 0.0002s under the dimension of 8. In this test instance, Sd performs well. It is a good idea to use it as a starting point for another algorithm or just as a quick approximation method to speed up the calculation. Overall, Mapm is considered to have the best effect of correction. The time is significantly less than Hd and the error is minimal. Mapm is considered to possess accuracy, speediness, and controllability at the same time, essentially addressing all the issue with the first two algorithms.

6. Conclusion

In this report, we discuss about the proposed two improvements to Latin Hypercube sampling. It has been demonstrated that section-mean sampling provides significantly greater accuracy in the simulated mean and variance for each variable. In sample ordering, we showed the flaws in the updated Latin Hypercube sampling approach in simulating target correlation between the variables. The proposed single-switch-optimized approach demonstrated accurate matching of target correlations between the variables despite taking a lot more computational time. These proposed improvements help us identify the strengths and limitations of Latin Hypercube sampling. By introducing three new algorithms to updated Latin Hypercube sampling for non-positive correlation matrix, we found better results in simulating correlation between the variables. With the proposed improvements, Latin Hypercube works well for simulating both correlated and uncorrelated variables. However, Latin Hypercube is not used for simulating independent variable as the output statistics are not accurate. In future if the computational power increases, single-switch optimizer may able to incorporate higher order moments which help us in simulating pseudo-independent variables. At present, Latin hypercube sampling still remains a powerful tool for Monte Carlo simulations.

References

[1] Florian, A. (1992). An efficient sampling scheme: Update latin hypercube sampling. probabilistic engineering mechanics. (7), 123–130.

[2] Huntington, D., & Lyrintzis, C. (1998). *Improvements to and limitations of latin hypercube sampling*. Elsevier.

[3] Iman, R. L., & Conover, W. J. (1980). Small sample sensitivity analysis techniques for computer models, with an application to risk assessment. (A9), 1749–1842.

[4] Iman, R., & Conover, W. (1982). Distribution free approach to inducing rank correlation among input variables. (B11), 311–334.

[5] Shinozuka, M., & Jan, C. M. (1972). Digital simulation of random processes and its applications. *Journal of sound and vibration*, (25), 111–128.

[6] Yang, Y. (2017). *An improved latin hypercube sampling method to enhance numerical stability considering the correlation of input variables.* IEEEAccess.