

Things you wanted to know about the Latin hypercube design and were afraid to ask

Felipe A. C. Viana

Probabilistics Laboratory, GE Global Research, Niskayuna, NY, USA

1. Abstract

The growing power of computers enabled techniques coined for design and analysis of simulations to be applied to a large spectrum of problems and to reach high level of acceptance among practitioners. Generally, when simulations are time consuming, a surrogate model replaces the computer code in further studies. The very first step for successful surrogate modeling and statistical analysis is the planning of the input configuration that will be used to exercise the simulation code. Among strategies coined for computer experiments, Latin hypercube designs have become particularly popular. This paper provides a short overview of the research in Latin hypercube design of experiments, highlighting potential reasons of its widespread use. The discussion starts with the early developments in optimization of the point selection and goes all the way to the pitfalls of always using Latin hypercube designs for selecting experimental designs. Then, final thoughts are given on how the Latin hypercube design fits in the state of the art as well as opportunities for future research.

2. Keywords: Design and analysis of computer experiments, Latin hypercube sampling, space-filling designs, sequential sampling.

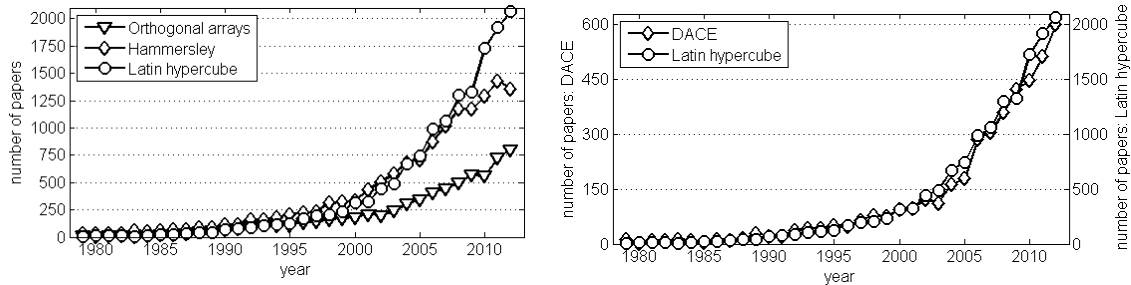
3. Introduction

Computer models are often used in sensitivity analysis, reliability assessment, design optimization and a number of other studies which tend to require many function evaluations. Very frequently, there is limited previous knowledge (particularly in situations like conceptual design) and engineers, designers, and analysts tend explore large design spaces. Years of research in mathematical formulation together with growing computer power enabled techniques coined for design and analysis of simulations [1]-[5] to be successfully applied to a variety of problems. Such techniques embrace the set of methodologies for generating a surrogate model (also known as metamodel or response surface approximation), which is used to replace the expensive simulation code. The goal is constructing an approximation of the response of interest based on a limited number of expensive simulations.

With that said, careful planning of the inputs for the computer codes is one of the most crucial steps for successful statistical modeling of the simulations. This is clearly elucidated in the vast literature about experimental designs for computer experiments [6]-[10]. Often, few statistical assumptions are made about the input/output relationship of computer models. That might be one of the reasons why the initial sample is planned to cover most of the considered domain (leading to space-filling experimental designs). Among strategies coined for computer experiments, Latin hypercube designs [11], [12] have become very popular. Other strategies include orthogonal arrays [13], and Hammersley designs [14], [15]. To illustrate their popularity, Figure 1 (a) shows an approximate number of publications that referred to at least one of these three techniques. The data was obtained using the Google Scholar (<http://scholar.google.com>) database. While the specific numbers may vary, as the Google Scholar database is updated, there is stronger growth in the number of papers using either Hammersley or Latin hypercube designs when compared to orthogonal arrays. Figure 1 (b) illustrates the close relationship between the growth in publications related to design of computer experiments and Latin hypercube design.

This paper aims at providing a short overview of the research in Latin hypercube design of experiments with few hypotheses to explain its extensive use. Given that Latin hypercube designs can create samples that poorly cover the input domain, optimization of the Latin hypercube is first discussed. New practitioners would find here some explanations for why peers recommend them to use Latin hypercube designs. Then, the pitfalls of using always Latin hypercube designs for selecting experimental designs are highlighted. Finally, the research in Latin hypercube designs is situated in the current state of the art and opportunities for future work are presented.

The remaining of the paper is organized as follows. Section 4 presents some general discussion about the Latin hypercube experimental design. Section 5 discusses five points of interest when researchers use such designs. Section 6 closes the paper recapitulating the important points and conclusions.

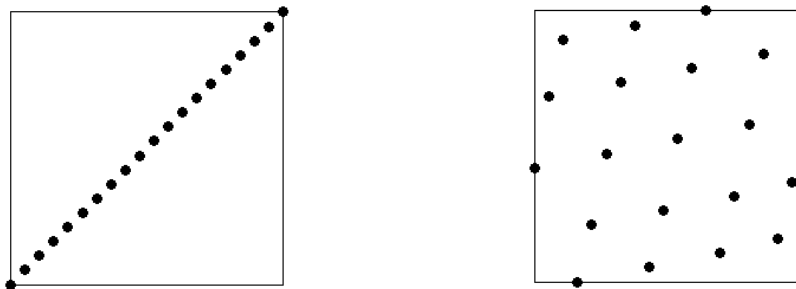


(a) Orthogonal arrays, Hammersley series, and Latin hypercube design. (b) DACE versus Latin hypercube related.

Figure 1: Number of papers published per year. Data obtained from the Google Scholar (<http://scholar.google.com>) database in the week of March 4, 2013. For the sampling schemes, the search was set with “any of these words”: “design of experiments” or “experimental design” or “sampling” plus “orthogonal arrays”, “Hammersley”, or “Latin hypercube”. For design and analysis of computer experiments (DACE), the search was set “any of these words”: “design of computer experiments” or “design of simulation experiments” or “design and analysis of computer experiments.”

4. Latin hypercube and other experimental designs

Here, an experimental design with p points in d dimensions is written as a $p \times d$ matrix $X = [x_1 \ x_2 \ \dots \ x_p]^T$, where each column represents a variable and each row $x_i = [x_i^{(1)} \ x_i^{(2)} \ \dots \ x_i^{(d)}]$ represents a sample. A Latin hypercube design is constructed in such a way that each of the d dimensions is divided into p equal levels (sometimes called bins) and that there is only one point (or sample) at each level. As originally proposed, a random procedure is used to determine the point locations. Figure 2 shows two examples of Latin hypercube designs with $d = 2$ and $p = 20$. Although unlikely, there is nothing preventing an experimental design to have poor space filling qualities, as the extreme case illustrated in Figure 2 (a). A better choice is shown in Figure 2 (b), where the points are more uniformly distributed over the domain.



(a) Design with poor space filling properties. (b) Design with good space filling properties.

Figure 2: Examples of Latin hypercube designs with $d = 2$ dimensions and $p = 20$ points.

There are other sampling techniques coined for with computer experiments¹. For example, orthogonal arrays organize the design matrix X in $p \times d$ matrix of integers $0 \leq x_{ij} \leq b - 1$. The array is said to have strength $t \leq d$ if in every p by t submatrix of X all of the b^t possible rows appear the same number of λ times ($p = \lambda b^t$). Latin hypercube sampling corresponds to strength $t = 1$, with $\lambda = 1$. Hammersley designs are based on Hammersley sequences. Much like Fibonacci series, the Hammersley sequences are built using operations on integer numbers. For further reading on these three sampling schemes, please refer to [14]-[16].

5. Five questions that make you think

This section presents an attempt to answer five intriguing questions about the Latin hypercube designs. By no means, the answers should be seen as definitive. Instead, they represent a partial (and why not say, personal) observation based on recent literature. The objective is trying to understand why Latin hypercube sampling is so popular, how much progress research has produced, what the limitations are, what the alternatives are, and what remains to be done.

¹ Although sub-efficient, there is nothing fundamentally wrong in using classical design of experiments (e.g, central composite designs, or D-optimal designs) for simulations. After all, in the early days of computer experiments, practitioners used to build polynomial response surfaces for simulations using such experimental designs.

5.1. Why do people like the Latin hypercube design so much?

There might be numerous reasons for the Latin hypercube popularity. One possible good reason is that it allows the creation of experimental designs with as many points as needed or desired. In the early days of design and analysis of computer experiments, the practitioners were using both sampling and modeling techniques developed for physical experiments (Appendix A briefly discusses physical and computer experiments.). The maturity of design and analysis of computer experiments as a discipline opened up the question: can sampling and modeling strategies be coined for computer experiments? Ideally, both the experimental design and the modeling strategies (not the topic here) would be developed simultaneously². Classical experimental designs meant to deal with the non-deterministic and relatively low dimensional nature of physical experiments³. For computer experiments, however, an attractive sampling technique would have to be flexible enough to provide data (a) for modeling techniques based on very different statistical assumptions and (b) capable of cover small and large design spaces (no constraints in terms of data density and location). The Latin hypercube design offers both.

Another good reason for the Latin hypercube popularity is flexibility. For example, if few dimensions have to be dropped out, the resulting design is still a Latin hypercube design (maybe sub-optimal, but a Latin hypercube nevertheless). That happens because, in Latin hypercube, samples are non-collapsing (orthogonality of the sampling points [22], [23]). Thus, if one cannot afford another set of data properly designed for the smaller domain, the existing data can be reused without reduction in number of sampled points. This is not the case when factorial or central composite designs are used. In such cases, once some of the dimensions are eliminated, points collapse one into another (reducing the sample size).

5.2. How far has optimization of the Latin hypercube design gone?

The need of flexible experimental design strategies took research in Latin hypercube design away from model-specific figures of merit (such as in D-optimal designs, which maximizes the determinant of the information matrix). For many of the applications, very little was assumed about the relationship between input and outputs (although covariance among inputs was often a topic such as when Latin hypercube sampling is used for multivariate integration). Instead, the figures of merit were expressed in terms of the input space (e.g., the minimum distance between points). The optimization of the Latin hypercube design has shown to be a challenging task because it is a combinatorial optimization problem with search space of the order of $(p!)^d$. For example, to optimize the location of 20 samples in two dimensions, the algorithm has to select the best design from more than 10^{36} possible designs. If the number of variables is increased to three, the number of possible designs is more than 10^{55} . Another challenge is the objective function computation that tends to be time consuming (and the number of operations grows very fast with number of points and variables).

The result is a very abundant literature on optimization of the Latin hypercube point location [24]-[35] (just to cite a few, but the list could go on and on). Table 1 summarizes some strategies found in the literature. Overall, two research focus are recurrent, namely, the optimization algorithm and the objective function. In few cases, researchers explored both simultaneously.

In terms of optimization algorithms, coordinate exchange, columnwise-pairwise, and enhanced stochastic evolutionary algorithms are naturally suitable for Latin hypercube optimization. That is because they can deal with combinatorial problems and they were designed to account for the non-collapsing structure of the Latin hypercube designs. However, literature has also shown the use of variations of discrete and continuous optimization methods such as simulated annealing and genetic algorithms [36].

As for the objective function, a lot of authors advocate in favor of criteria intended to space-filling designs (e.g., potential energy, entropy, and ϕ_p). Some authors argue that reduced correlation among inputs is also important (e.g., L2-discrepancy). Unfortunately, it is difficult to relate any of these criteria with the accuracy of the resulting metamodels. Probably because of that, the choice of objective function for Latin hypercube optimization is not unanimous. Nevertheless, some combinations of objective functions and algorithms might favor the overall performance of the optimization strategy. For example, Jin et al. [30] proposed strategies for efficient calculation

² After all, only so much could be achieved if the modeling approach remained the same (i.e., polynomial response surfaces). That is when techniques such as Gaussian process (and kriging), radial basis function, neural networks, support vector machines and others play an important role (see the abundant literature [17]-[21]).

³ The low dimensionality sometimes found in physical experiments is usually an imposition of the high cost associated with them (i.e., exploration of large design spaces is often prohibitively expensive).

of the ϕ_p , entropy and L2 discrepancy criteria that take advantage of the fact that only two elements in the design matrix are involved in each iteration of their algorithm (resulting in significant savings in computation time).

Table 1: Review of approaches for constructing the optimal Latin hypercube design (adapted from [34]).

Researchers	Year	Algorithm	Objective functions
Audze and Eglajs [24]	1977	Coordinates exchange	Potential energy
Park [25]	1994	2-stage: exchange- and Newton-type	Integrated mean-squared error and entropy criteria
Morris and Mitchell [26]	1995	Simulated annealing	ϕ_p criterion
Ye et al. [27]	2000	Columnwise-pairwise	ϕ_p and entropy criteria
Fang et al. [28]	2002	Threshold accepting	Centered L2-discrepancy
Bates et al. [29]	2004	Genetic algorithm	Potential energy
Jin et al. [30]	2005	Enhanced stochastic evolutionary algorithm	ϕ_p criterion, entropy and L2 discrepancy
Liefvendahl and Stocki [31]	2006	Columnwise-pairwise and genetic algorithms	Minimum distance and potential energy
van Dam et al. [32]	2007	Branch-and-bound	1-norm and infinite norm distances
Grosso et al. [33]	2008	Iterated local search and simulated annealing algorithms	ϕ_p criterion
Viana et al. [34]	2010	Translational propagation	ϕ_p criterion
Zhu et al. [35]	2012	Successive local enumeration	Potential energy

The growing power for computers changed the perception about the performance and limitation of the algorithms. For example, in 2000, Ye et al. [27] reported that generating an optimal Latin hypercube of 25 points and four variables would take several hours on a Sun SPARC 20 workstation. In 2005, Jin et al. [30] reported that it would take only 2.5 seconds to optimize the same size of design in a Pentium III 650 MHZ CPU. Today time-consuming designs might be on the order of hundreds of points and/or several tenths of variables.

Optimization and computer power are definitely strong drivers for the Latin hypercube popularity. The combination of these two elements has enabled experimental designs with very good space-filling properties at reasonable computational cost. Figure 3 shows the difference in number of papers acknowledging the use of Latin hypercube versus Hammersley designs over the years. The difference used to be small and favoring Hammersley designs. However, the trend started to change as the optimization strategies for Latin hypercube became mature and implementations became fast enough from the user's perspective. As a warning, this does not mean that Latin hypercube are better than Hammersley designs. It only gives a hint that affordably optimized Latin hypercube designs tend to be used more often.

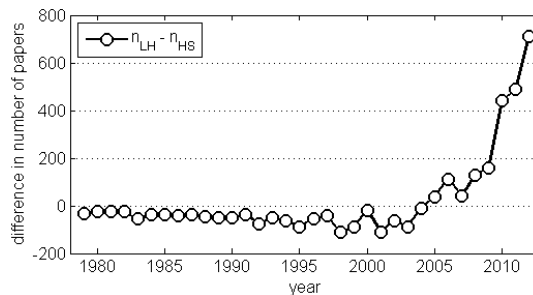


Figure 3: Difference between number of publications over the years. n_{LH} and n_{HS} are the number of papers using Latin hypercube and Hammersley designs according to Figure 1 (a), respectively.

5.3. Do Latin hypercube designs have any drawback?

Most certainly Latin hypercube designs have drawbacks. By virtue of their definition, Latin hypercube designs have good uniformity with respect to each dimension individually. On the other hand, desirable properties, such as space filling, or column-wise orthogonality come at the cost of very expensive optimization. As discussed previously, improved space-filling properties may be achieved by minimizing some form of distance measure, whereas orthogonality may be obtained by considering column-wise correlations. Unfortunately, literature [37]-[40] has shown that optimization with respect to either of these properties does not necessarily lead to the best experimental design with respect to the other property. Figure 4 shows the scatter plot of the column-wise correlation and ϕ_p criteria when 1000 two-dimensional Latin hypercube designs 20 points are created with the MATLAB function *lhsdesign* [41] (using default parameters). This figure illustrates how difficult it might be to find the experimental that minimizes both the column-wise correlation and ϕ_p criteria. In addition, it is clear that the best designs for the column-wise correlation can greatly differ in term of ϕ_p criterion, and vice versa. The interested reader can find at <http://harvest.nps.edu> a catalogue of ready-to-use, nearly orthogonal and good space-filling designs for up to 22 variables in as few as 129 points.

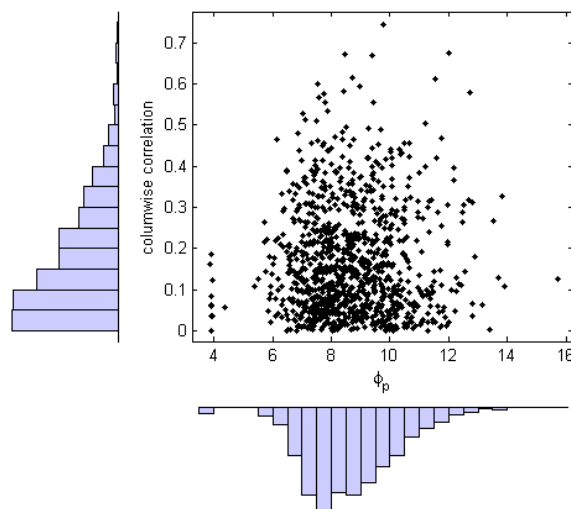


Figure 4: Scatterplot of column-wise correlation and ϕ_p criteria out of 1000 Latin hypercube designs with $d = 2$ dimensions and $p = 20$ points created with the MATLAB function *lhsdesign* [41].

As any other design of experiment, the Latin hypercube also suffers from the curse of dimensionality. While uniformity in each dimension is preserved, the space filling properties become questionable. As the number of variables increase, it becomes harder to fill the design space. When optimization pushes points further apart, the sample tends to create a vacuum in the center of the design space. Again, this is not only observed in the Latin hypercube designs and it might just be something that is typical in high dimensions.

5.4. When studying computer experiments, should Latin hypercube designs always be used?

In general, design for computer experiments should observe good coverage of the design space, many levels for each variable, and good projection properties. Sure, the Latin hypercube designs are conveniently coined for all that, but that does not guarantee they will always provide the best initial sample for a given problem. Figure 5 illustrates a Hammersley design in two-dimensional space. Using only visual inspection, this design could very well be rated as good the Latin hypercube of Figure 2-(b).

Because different sampling strategies (including their optimized variants) might lead to similar designs in term of space-filling properties, literature does not point to any particular approach that would work best all the time [42], [43]. In fact, Goel et al. [44] have empirically demonstrated losses associated with using a single experimental design strategy (they even suggested combing different design of experiments).

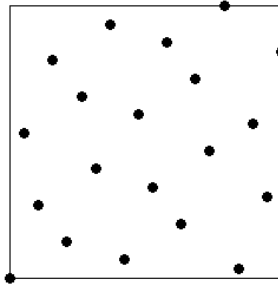


Figure 5: Example of Hammersley design with $d = 2$ dimensions and $p = 20$ points.

5.5. Where are the research opportunities in Latin hypercube design?

Although a lot was accomplished since the introductory papers of Mckay et al. [11] and Iman and Conover [12], there are still plenty of open issues. Table 2 discusses some research topics that are still open. Overall, one strong tendency is to expand the capabilities by relaxing some of the Latin hypercube properties.

Table 2: Comments on open research topics.

Research topic	Comments and example of work done
Optimization of Latin hypercube	<p>This has probably been one of the most visited research topics. The growing in computer power has made optimization feasible from small to moderate sampling sizes. However, creating experimental designs in large dimensional spaces (when the sample size is also naturally large) is still very challenging. The task is time consuming because computation of optimization criteria and number of iterations for convergence (and there is always the debate around the curse of dimensionality).</p> <p>Owen [45] introduced Latin supercube sampling (combination of Latin hypercube with Quasi Monte Carlo) for numerical integration of functions defined in very high dimensional spaces.</p>
Mixing discrete and continuous variables	<p>In Latin hypercube designs, each dimension is divided into equal number of levels (which is also equal to the number of points in the experimental design). That imposes a limitation when dealing with discrete variables. The number of levels in the experimental design might not coincide with the levels of the discrete variables. One can always map few Latin hypercube levels into a single level of the discrete variable. Nevertheless, that might end up being a sub-optimum strategy. There is definitely opportunity for research on efficient strategies for problems that combine discrete and continuous variables.</p> <p>Meckesheimer et al. [46] evaluated different sampling and surrogate techniques when applied to discrete/continuous problems.</p>
Incorporation of global sensitivity information	<p>In some applications, it is known beforehand that the response varies much slower in certain dimensions (or faster in others). In such cases, it would make sense to have few levels in the slower changing dimensions and more levels in the fast changing ones.</p> <p>Researchers like Tarantola et al. [47] have explored different strategies for assessing global sensitivities. Nevertheless, to the best of my knowledge, there is no sampling strategy that would use sensitivity information (i.e., without a model) to decide where to conduct simulations/experiments.</p>
Sequential sampling	<p>This is the case in which a first set of points is used to create a surrogate model, which turns out to perform poorly. Then, the experimental design is augmented so that with the new points the quality of the surrogate model would improve.</p> <p>Rennen et al. [48] proposed nested designs which the resulting new design is also a Latin hypercube (one advantage of such approach is the easy of computation and potentially large number of new points). On the other hand, several authors have proposed using surrogate models, such as the Gaussian process, to help identifying the next sampling points (e.g., by maximizing Gaussian process uncertainty). The interested reader is referred to [49], [50], and [10].</p>

6. Summary and Conclusions

This paper presented an overview of the research in Latin hypercube sampling. The objective was to present, especially to the newcomers, a critical view on this class of design of experiments. To do that, the discussion was divided into five topics:

- 1) Popularity: Latin hypercube are very well accepted, particularly in studying computer experiments, because of flexibility in terms of data density and location, and in addition, non-collapsing and space-filling properties.
- 2) Optimization: research (both in terms of algorithm and problem formulation) has enabled optimization of point location improving space filling and reducing correlation among points.
- 3) Drawbacks: research has not achieved definitive conclusion about an optimization scheme that generates space-filling and uncorrelated samples. In addition, as any other sampling scheme, the Latin hypercube design also suffers from the curse of dimensionality.
- 4) Alternatives: there is rich literature in experimental designs for computer experiments (e.g., orthogonal arrays and Hammersley designs). There is also evidence that combining different design strategies might be helpful when developing surrogate models.
- 5) Open issues: few suggestions were given in Table 2.

After going through these topics, the reader should have an appreciation for the work that was already done to improve the originally proposed Latin hypercube design (work that is the basis of its popularity). Hopefully, the discussions also raised awareness to the limitations and what remains to be done while providing a collection of relevant references for the interested reader.

Appendix

The author is very thankful to GE for supporting the publication of this paper. The views expressed here reflect the views of the author alone and do not necessarily reflect the views of GE.

Appendix

A. Design and analysis of physical versus computer experiments

Depending on the point of view, one might find more similarities than differences between physical and computer experiments. When it comes to the techniques for design and analysis of experiments, the differences might be more related to the history of developments than the problem itself. Certainly, design and analysis of physical experiments is a much older field (see the book by Montgomery [51] for a very good reference on the topic). Sampling and modeling techniques for physical experiments were first developed in a time where computers had very limited capabilities (that might be the reason for the extensive use of polynomial response surface). On the other hand, the design and analysis of computer experiments is relatively recent; and since the beginning, the field could take advantage of better computers.

In general, computer experiments are used to make analysis of physical systems easier (or sometimes possible). This notion implies that computer experiments tend to be relatively cheaper than their physical counterparts⁴. As a result, computer models are used in sensitivity analysis, reliability assessment, design optimization and a number of other studies that tend to require a large number of function evaluations. Very often, there is limited previous knowledge (particularly in situations like conceptual design) and engineers, designers, and analysts tend explore a large number of input variables (defined over relatively large domains). Physical experiments being expensive are used when very little is known about the actual system (e.g., early phases of new material development) and different versions of design validation.

⁴ Keep in mind that there might be cases in which that do not hold. For example, studying flapping wings in micro aerial vehicles [52] might sometimes be cheaper through prototyping rather than simulations. Once the test facility is available (which tends to be expensive part), building and testing different wing configurations becomes more affordable than running full-fledged unsteady computational fluid dynamics models.

Last but not least, there is a set of computer models that are deterministic⁵ (given a set of inputs, the outputs are always the same) and build to represent physical systems at given fidelity level (degree of physics the model can describe). On the other hand, physical experiments are stochastic because they are inevitably susceptible to measurement error to say the least. The notion of fidelity level may appear but physical experiments tend to be seen as observation of reality (again, probably corrupted by some measurement error). Obviously, there are cases (e.g., high-energy physics) where physical similitude is explored. In such cases, an easier to control experiment “approximates reality” (e.g., experiments conducted at lower temperatures with results appropriately scaled to estimate what would happen at high temperatures).

7. References

- [1] J Sacks, WJ Welch, TJ Mitchell, and HP Wynn, “Design and analysis of computer experiments,” *Statistical Science*, Vol. 4, pp. 409-435, 1989.
- [2] TJ Santner, BJ Williams, and WI Notz, *The design and analysis of computer experiments*, Springer, 2003.
- [3] TW Simpson, V Toropov, V Balabanov, and FAC Viana, “Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come – or not,” *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, Canada, Sep 10-12, 2008. AIAA 2008-5802.
- [4] AIJ Forrester, and AJ Keane, “Recent advances in surrogate-based optimization,” *Progress in Aerospace Sciences*, Vol. 45 (1-3), pp. 50-79, 2009.
- [5] JPC Kleijnen, *Design and Analysis of Simulation Experiments*, Springer, New York, 2009.
- [6] MD Morris and TJ Mitchel, “Exploratory designs for computer experiments,” *Journal of Statistical Planning and Inference*, Vol. 43, pp. 381-402, 1995.
- [7] TW Simpson, DKJ Lin, and W Chen, “Sampling strategies for computer experiments: design and analysis,” *International Journal of Reliability and Applications*, Vol. 2 (3), pp. 209-240, 2001.
- [8] KT Fang and R Li, “Uniform design for computer experiments and its optimal properties,” *International Journal of Materials and Product Technology*, Vol. 25 (1), pp. 198-210, 2006.
- [9] VCP Chen, KL Tsui, RR Barton, and M Meckesheimer, “A review on design, modeling and applications of computer experiments,” *AIEE Transactions*, Vol 38 (4), pp. 273-291, 2006.
- [10] L Pronzato and WG Müller, “Design of computer experiments: space filling and beyond,” *Statistics and Computing*, Vol. 22(3), pp. 681-701, 2012.
- [11] MD Mckay, RJ Beckman, and WJ Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, Vol. 21 (2), pp. 239-245, 1979.
- [12] RL Iman, WJ Conover, “Small sample sensitivity analysis techniques for computer models, with an application to risk assessment,” *Communications in Statistics*, Part A. Theory and Methods, Vol. 17, pp. 1749-1842, 1980.
- [13] AB Owen, “Orthogonal arrays for computer experiments, integration and visualization,” *Statistics Sinica*, Vol. 2, pp. 439-452, 1992.
- [14] UM Diwekar and JR kalagnanam, “Efficient sampling technique for optimization under uncertainty,” *AIChE Journal*, Vol 43 (2), pp 440-447, 1997.
- [15] JM Hammersley, “Monte Carlo methods for solving multivariate problems,” *Annals of the New York Academy of Sciences*, Vol. 86 (3), pp. 844-874, 1960.
- [16] AA Giunta, SF Wojtkiewicz Jr, and MS Eldred, “Overview of modern design of experiments methods for computational simulations,” *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, Jan 6-9, 2003 , AIAA 2003-0649.
- [17] CE Rasmussen and CKI Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.
- [18] ML Stein, *Interpolation of Spatial Data: some theory for kriging*, Springer Verlag, 1999.
- [19] J Park and IW Sandberg, “Universal approximation using radial-basis-function networks,” *Neural Computation*, Vol. 3 (2), pp. 246–257, 1991.
- [20] M Smith, *Neural Networks for Statistical Modeling*, Von Nostrand Reinhold, 1993.
- [21] B Schölkopf and AJ Smola, *Learning with Kernels*, The MIT Press, 2002.
- [22] TW Simpson, JD Peplinski, PN Koch, JK Allen, “Meta-models for computer based engineering design: survey and recommendations,” *Engineering with Computers*, Vol. 17(2), pp. 129-150, 2001.
- [23] JPC Kleijnen, SM Sanchez, TW Lucas, TM Cioppa, “A user’s guide to the brave new world of designing simulation experiments,” *INFORMS Journal on Computing*, Vol. 17(3), pp. 263-289, 2005.
- [24] P Audze, V Eglajs, “New approach for planning out of experiments,” *Problems of Dynamics and Strengths*, Vol. 35, pp. 104-107, 1977 (in Russian).

⁵ Obviously, there are stochastic computer codes (e.g., simulations based on Monte Carlo methods). The interested reader can find further reading in [53].

- [25] JS Park, "Optimal Latin-hypercube designs for computer experiments," *Journal of Statistical Planning and Inference*, Vol. 39, pp. 95-111, 1994.
- [26] MD Morris, TJ Mitchell, "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, Vol. 43, pp. 381-402, 1995.
- [27] KQ Ye, W Li, A Sudjianto, "Algorithmic construction of optimal symmetric Latin hypercube designs," *Journal of Statistical Planning and Inference*, Vol. 90, pp.145-159, 2000.
- [28] KT Fang, CX Ma, P Winker, "Centered L2-discrepancy of random sampling and Latin hypercube design and construction of uniform designs," *Mathematics of Computation*, Vol. 71, pp. 275-296, 2002.
- [29] SJ Bates, J Sienz, VV Toropov, "Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm" *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Palm Springs, CA, April 19-22, 2004.
- [30] R Jin, W Chen, and A Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments," *Journal of Statistical Planning and Inference*, Vol. 134, pp. 268-287, 2005.
- [31] M Liefvendahl and R Stocki "A study on algorithms for optimization of Latin hypercubes," *Journal of Statistical Planning and Inference*, Vol. 136 (9), pp. 3231-3247, 2006.
- [32] E van Dam, B Husslage, D den Hertog, and H Melissen, "Maximin Latin hypercube designs in two dimensions," *Operations Research*, Vol. 55 (1), pp. 158-169, 2007.
- [33] A Grosso, A Jamali, and M Locatelli, "Finding maximin Latin hypercube designs by iterated local search heuristics," *European Journal of Operational Research*, Vol. 197(2), pp. 541-547, 2009.
- [34] FAC Viana, G Venter, V Balabanov, "An algorithm for fast generation of optimal Latin hypercube designs," *International Journal for Numerical Methods in Engineering*, Vol. 82, pp. 135-156, 2010.
- [35] H Zhu, L Liu, T Long, and L Peng, "A novel algorithm of maximin Latin hypercube design using successive local enumeration," *Engineering Optimization*, Vol. 44 (5), pp. 551-564, 2012.
- [36] M Liefvendahl and R Stocki, "A study on algorithms for optimization of Latin hypercubes," *Journal of Statistical Planning and Inference*, Vol. 2006, pp. 3231-3247, 2006.
- [37] KQ Ye, "Orthogonal column Latin hypercubes and their application in computer experiments," *Journal of the American Statistical Association*, Vol. 93 (444), pp. 1430-1439, 1998.
- [38] TM Cioppa and TW Lucas, "Efficient nearly orthogonal and space-filling Latin hypercubes," *Technometrics*, Vol. 49 (1), pp. 45-55, 2007.
- [39] P Prescott, "Orthogonal-column Latin hypercube designs with small samples," *Computational Statistics and Data Analysis*, Vol. 53, pp. 1191-1200, 2009.
- [40] F Sun, MQ Liu, and DKJ Lin, "Construction of orthogonal Latin hypercube designs with flexible run sizes," *Journal of Statistical Planning and Inference*, Vol. 140, pp. 3236-3242, 2010.
- [41] Mathworks contributors, *MATLAB: The language of technical computing*, Version 7.12.0.635 (R2011a), The MathWorks Inc., 2011.
- [42] SB Crary, "Design of computer experiments for metamodel generation," *Analog Integrated Circuits and Signal Processing*, Vol 32 (1), pp. 7-16, 2002.
- [43] D Bursztyn and DM Steinberg, "Comparison of design for computer experiments," *Journal of Statistical Planning and Inference*, Vol. 136, pp. 1103-1119, 2006.
- [44] T Goel, RT Haftka, W Shyy, and LT Watson, "Pitfalls of using a single criterion for selecting experimental designs," *International Journal for Numerical Methods in Engineering*, Vol. 75, pp. 127-155, 2007.
- [45] AB Owen, "Latin supercube sampling for very high-dimensional simulations," *ACM Transactions on Modeling and Computer Simulation*, Vol. 8(1), pp. 71-102, 1998.
- [46] M Meckesheimer, RR Barton, TW Simpson, F Limayem, and B Yannou, "Metamodeling of combined discrete/continuous responses," *AIAA Journal*, Vol. 39 (10), pp. 1950-1959, 2001.
- [47] S Tarantola, W Becker, and D Zeitz, "A comparison of two sampling methods for global sensitivity analysis," *Computer Physics Communications*, Vol. 183 (5), pp. 1061-1072, 2012.
- [48] G Rennen, B Husslage, ER van Dam, and D den Hertog, "Nested maximin Latin hypercube designs," *Structural and Multidisciplinary Optimization*, Vol. 41 (3), pp. 371-395, 2010.
- [49] F Xiong, Y Xiong, W Chen, and S Yang, "Optimizing Latin hypercube design for sequential sampling of computer experiments," *Engineering Optimization*, Vol. 41, pp. 793-810, 2009.
- [50] 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*, Vol. 214, pp. 683-696, 2011.
- [51] DC Montgomery, *Design and Analysis of Experiments*, 2001.
- [52] P Ifju, D Jenkins, S Ettinger, Y Lian, W Shyy, and M Waszak, "Flexible-wing-based micro air vehicles," *40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, Jan 14-17, 2002, AIAA 2002-0705.
- [53] BD Ripley, *Stochastic Simulation*, Wiley, 2009.