

Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models



Razi Sheikholeslami ^{a,*}, Saman Razavi ^{a,b}

^a Global Institute for Water Security, School of Environment and Sustainability, University of Saskatchewan, Saskatoon, Canada

^b Department of Civil and Geological Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

ARTICLE INFO

Article history:

Received 18 August 2016

Received in revised form

6 March 2017

Accepted 11 March 2017

Available online 23 March 2017

Keywords:

Design of computer experiments

Sequential sampling

Optimal Latin hypercube sampling

Monte Carlo simulation

Uncertainty analysis

Sensitivity analysis

ABSTRACT

Efficient sampling strategies that scale with the size of the problem, computational budget, and users' needs are essential for various sampling-based analyses, such as sensitivity and uncertainty analysis. In this study, we propose a new strategy, called Progressive Latin Hypercube Sampling (PLHS), which sequentially generates sample points while progressively preserving the distributional properties of interest (Latin hypercube properties, space-filling, etc.), as the sample size grows. Unlike Latin hypercube sampling, PLHS generates a series of smaller sub-sets (slices) such that (1) the first slice is Latin hypercube, (2) the progressive union of slices remains Latin hypercube and achieves maximum stratification in any one-dimensional projection, and as such (3) the entire sample set is Latin hypercube. The performance of PLHS is compared with benchmark sampling strategies across multiple case studies for Monte Carlo simulation, sensitivity and uncertainty analysis. Our results indicate that PLHS leads to improved efficiency, convergence, and robustness of sampling-based analyses.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Simulation models have become an essential tool for environmental and water resources systems analysis and have been extensively employed to tackle various complex problems, including water supply systems design and operation, water quality and wastewater management, groundwater management, rainfall-runoff modelling, design optimization, risk assessment, and decision making. These models are typically characterized by: (1) highly non-linear/complex response surfaces, (2) large parameter/problem spaces (high-dimensional with large uncertainty at each dimension), and (3) high computational demand (long run times). These three characteristics challenge the use of sampling-based analyses such as uncertainty estimation (Liu and Gupta, 2007; Mugunthan and Shoemaker, 2006; Kuczera and Parent, 1998), sensitivity analysis (Sarrazin et al., 2016; Razavi and Gupta, 2015), surrogate modelling and optimization (Maier et al., 2014; Razavi et al., 2012a; Vrugt et al., 2006), and other Monte Carlo type simulations (e.g. Rezaie et al., 2007; Linkov and Ramadan, 2004). The first characteristic above necessitates the collection of a sufficiently

dense and properly distributed samples to adequately characterize the nonlinearity of the response surface, while the second requires a large sample size spreading across the entire high-dimensional space to ensure adequate exploration and coverage of the space. These, when coming with the third characteristic, impose significant computational burdens that may impede effective sampling-based analyses.

Sampling is a main building block of a range of algorithms designed for various types of environmental and water resources systems analysis. Depending on the type of analysis, sampling may be merely "input-oriented" (also called "model-free" here), where no adaptation made based on resulting model outputs, or it may also be "output-oriented", where the sampling procedure is guided/adapted based on feedback received from the model outputs during sampling. Examples of the former, which is our focus in this paper, include Monte Carlo simulation for uncertainty propagation, GLUE type methods for uncertainty analysis, sensitivity analysis, design of experiments, and some variations of surrogate modelling (e.g., response surface methodology and the other approaches categorized under "basic sequential framework" in Razavi et al. (2012b)). Examples of the latter include the surrogate modelling strategies where the response surface approximation evolves over time (i.e., the approaches categorized under "Adaptive-Recursive Framework" and "Metamodel-Embedded Evolution

* Corresponding author.

E-mail address: razi.sheikholeslami@usask.ca (R. Sheikholeslami).

Framework" in Razavi et al. (2012b)); generating perturbed hydrometeorological forcing data using the inverse approach (Guo et al., 2016); scenario generation and classification (Islam and Pruyt, 2016); and sensitivity analysis in multi-criteria decision analysis (Ganji et al., 2016).

The performance of a sampling strategy and the quality of its resulting samples directly controls the efficiency and robustness of any associated sampling-based analysis. A sample is deemed of quality if it possesses the intended distributional properties in parameter/problem space. The distributional properties of a sample are commonly assessed by one-dimensional projections for every dimension (i.e., marginal distributions), space-filling criteria, and correlation analysis (see Section 2). There is a wealth of literature over the past several decades on developing and improving various sampling strategies, including pseudo random sampling, stratified sampling, fractional and full factorial design (Box and Hunter, 1961), regular grid sampling, orthogonal design (Owen, 1992), Latin hypercube sampling (McKay et al., 1979), and Sobol' sequences (Sobol', 1967). Latin Hypercube Sampling (LHS), pioneered by McKay et al. (1979) and Iman and Conover (1980) and its variations such as orthogonal array-based LHS (Tang, 1993), orthogonal LHS (Ye, 1998), and symmetric LHS (Ye et al., 2000) are among the most commonly used sampling techniques for experiments with environmental and water resources systems models in a variety of application areas such as sensitivity and uncertainty analysis (e.g. Posselt et al., 2016; Gan et al., 2014; Zhan and Zhang, 2013), parameter calibration (e.g. Higdon et al., 2013), and surrogate modelling (e.g. Rajabi et al., 2015; Regis and Shoemaker, 2007). This may be mainly because of their (1) ease of use (comparable with random sampling), (2) insurance of one-dimensional projection properties ("Latin Hypercube" properties), and (3) ease in incorporating other criteria (e.g., orthogonality and symmetry) within sampling. Due to the second characteristic, LHS can be deemed a form of stratified sampling because it stratifies across the range of variables in accordance with the distributional properties of interest.

An effective sampling strategy needs to ensure the above properties, while being capable of scaling the distribution of sampled points with *any sample size*. The "proper sample size", however, for a given simulation model and sampling-based analysis is not typically known *a priori*. The proper sample size here refers to a sufficiently large number of sample points that will lead to convergence or robustness of the analysis results. On the other hand, the computational cost of a sampling-based analysis is linearly proportional to the sample size (i.e., number of model runs), assuming the computational demand of the sampling strategy is relatively negligible.

One major drawback of traditional LHS and many other sampling strategies is that they generate the entire sample points at once, a process that is referred to as "one-stage" or "one-shot" sampling hereafter. This requires users to specify the sample size prior to the associated sampling-based analysis. Users, therefore, tend to utilize larger sample sizes, which possibly impose unnecessarily larger computational demand, to avoid "under-sampling" of the parameter/problem space. Also, it is often the case that the user is not satisfied with the resulting sampling-based analysis (e.g., convergence criteria are not met), and needs to enlarge the sample size and resumes the sampling-based analysis with the updated/new sample. In this case, the user will have a dilemma: either to generate a new sample by LHS with the size of interest and add it to the previously generated sample with the tradeoff that the union of the two samples will *not* be Latin hypercube, or to discard the previous sample at a computational cost and generate a new, larger sample to preserve the distributional properties of interest. Such needs warrant the development and application of "multi-

stage" or "sequential" sampling, where sample size can grow progressively, while maintaining the desired distributional proprieties. This way, sequential sampling will allow the user to monitor the performance of the sampling-based analysis and assess the stopping criteria (e.g., convergence criteria) in an online manner.

In this paper, we introduce a new and efficient sequential version of LHS, called Progressive Latin Hypercube Sampling (PLHS). As opposed to the traditional LHS approach that generates the entire sample set in one stage, the proposed PLHS will generate a series of smaller sub-sets while: (i) the first sub-set is Latin hypercube; (ii) the progressive addition of sub-sets remains Latin hypercube and achieves maximum stratification in any one-dimensional projection; and thus (iii) the entire sample set is Latin hypercube. In other words, PLHS will preserve the desired distributional properties while the sample size grows during the analysis. With several sampling-based numerical experiments for sensitivity and uncertainty analysis, we show that the proposed PLHS has multiple advantages over the one-stage sampling strategies, including improved convergence of the associated analysis and the robustness of the results to sampling variability.

The paper is structured as follows. Section 2 briefly explains the properties of original LHS, discusses different strategies for optimal sampling, and reviews existing sequential sampling methods. Next, in Section 3, we propose the PLHS approach, followed by a description of a heuristic algorithm for efficiently generating the optimal PLHS. Section 4 describes case studies and experimental setup that are used to assess the PLHS. In Section 5, the experimental results and the analysis of the algorithm are provided and discussed. Finally, Section 6 concludes the paper.

2. Literature review

2.1. What is Latin hypercube sampling?

Latin hypercube sampling (LHS) was inspired by the concept of "Latin square" from combinatorial mathematics, where an n -by- n matrix is filled with n different objects (i.e., numbers, characters, symbols, etc.) such that each object occurs exactly once in each row and exactly once in each column— see Fig. 1a for an example with 4 objects. The term "Latin" in Latin squares was inspired by the work of the famous mathematician, Leonhard Euler, who used Latin characters as the objects (Wallis and George, 2011). Like Latin squares, the basic idea of LHS for a 2-dimesnional space and a sample size of n is partitioning each dimension into n disjoint intervals (levels) with equal marginal probability of $1/n$ and then randomly sampling once from each interval to ensure that there is

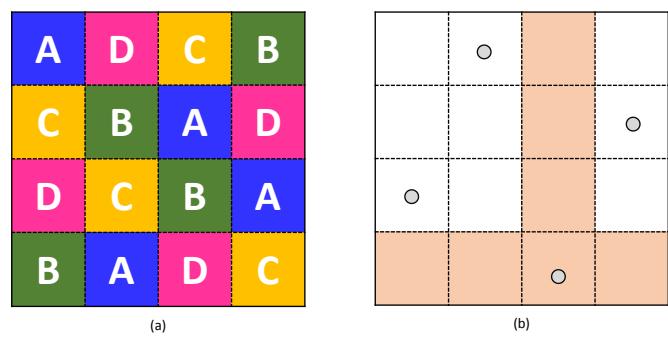


Fig. 1. An illustration of the basic idea of Latin hypercube sampling: (a) A 4 by 4 example of Latin square - 4 different Latin characters are arranged in a way that no letter appears more than once in each row or column. (b) A 2-dimesnional example of LHS with 4 sample points - There is only one point in each row and each column (the row and column taken by one of the sample points are darkened).

only one point at each level. Fig. 1b shows a 2-dimensional example with 4 sample points *uniformly distributed* across each dimension by LHS. With no loss of generality, and for the sake of simplicity, the definitions and examples presented in this paper are for the case of uniform distribution. For any other distribution (e.g., normal distribution), the uniformly distributed samples can be transformed by associated transformation functions.

In the general case, consider a unit hypercube in a p -dimensional space, $C^p = [0,1]^p$, divided into n intervals (n is the sample size) with an equal length of $1/n$ along each axis – this creates n equally probable intervals indexed by $q = 1, \dots, n$ corresponding to $[0, 1/n], [1/n, 2/n], \dots, [(n-1)/n, 1]$ for each dimension. LHS can be represented as n -by- p sample matrix $[x_{i,j}]$ ($i = 1, \dots, n$; $j = 1, \dots, p$), where $x_{i,j} \in [0, 1]$ such that $x_{i,j}$ in the j th column belongs to only one interval. In other words, q is a random permutation of $\{1, 2, \dots, n\}$ for each column, and each row of the matrix is a sample point. We denote this matrix by $\text{LHS}(n, p)$. Original LHS ensures that the resulting sample possesses one-dimensional projection properties, indicating the projection of sample points in the p -dimensional space onto any dimension will follow the uniform distribution (or any other distribution of interest). Therefore, a sample is said to be “Latin hypercube” if and only if it possesses the one-dimensional projection properties. Such a sample, however, is only guaranteed to maximize the stratification in marginal distributions, while the multi-variate distributional properties (e.g., space-filling properties) in the p -dimensional space are not necessarily accounted for. There have been research efforts across a variety of fields to improve the performance of the original LHS, and several strategies were built on the original LHS, including Orthogonal Array-based LHS (Tang, 1993), Orthogonal LHS (Ye, 1998), and Symmetric LHS (Ye et al., 2000). For more detail, interested readers are referred to the reviews of the state of the art by Viana (2013) and Helton and Davis (2003).

2.2. Sampling improvements based on the notion of optimization

In addition to the methodological improvements such as orthogonal LHS and symmetric LHS that attempt to systematically generate Latin hypercube samples of better quality, there have been a large number of studies that utilize the optimization theory to improve the performance of LHS (Pronzato and Müller, 2012; Xiong et al., 2009). Basically, the approach is to define secondary criteria (objective functions), in addition to being Latin hypercube, and formulate and solve an optimization problem to achieve (near)

optimal Latin hypercube samples. This can be very effective, as in any variation of LHS, there may exist a huge number of configurations (sample points arrangements) that satisfy the associated LHS criteria, but do poorly in terms of other criteria (e.g., space filling). The LHS algorithms typically *randomly* pick one of the many possible configurations, while optimization helps navigate through the myriad of choices and identify one that is (near) optimal in terms of secondary criteria.

The optimization problem for improving LHS belongs to the class of combinatorial optimization, with a total search space of $(n!)$ p configurations for an exhaustive search (Viana et al., 2010). The computational efficiency of an optimization-assisted LHS algorithm depends on the size of the search space and the efficiency of the optimization algorithm used. In practice, therefore, the sampling procedure can become computationally demanding for larger values of n and p . A variety of optimization algorithms have been used in the literature to solve such combinatorial problems, including simulated annealing, genetic algorithms, and the branch-and-bound-algorithm (see Table 1 for a list of studies). However, in performing sampling-based analysis using LHS, there are two notable remarks. First, once an optimal LHS is generated, it is independent of the considered application (model-free) and can be stored for future applications. Second, in most cases the computational cost of finding an optimal LHS is negligible in comparison with the time needed to run the computationally expensive computer simulations.

Determination of the appropriate objective function(s) to be imposed on Latin hypercube properties (i.e., one-dimensional projection properties) via optimization has been a major topic of research. Table 1 also reports different criteria that have been used in the literature as objective functions in different optimization-assisted LHS algorithms. These optimality criteria are mainly intended to improve space-filling properties to ensure that the sample points are uniformly scattered across the input space with minimal unsampled regions. The two most commonly used objective functions are (1) maximizing the minimum inter-point distance among all possible pairs of sample points, and (2) minimizing the correlations (absolute value) between all pairs of columns of the sample matrix. To explain these, Fig. 2 presents three example configurations when $n = 9$ and $p = 2$. The sample shown in Fig. 2a is optimal because it is Latin hypercube with strong space-filling properties, whereas the sample shown in Fig. 2b is not Latin hypercube, although it has strong space-filling properties. Fig. 2c shows a very poor Latin hypercube sample that, despite

Table 1
A review summary of studies for constructing optimal LHS.

Search algorithm	Criteria	Reference
Enhanced stochastic evolutionary algorithm	Max(dist) ^a	Husslage et al. (2011)
Translational propagation	Max(dist)	Viana et al. (2010)
Branch-and-bound	Max(dist)	van Dam et al. (2007)
Genetic algorithm	Max(dist)	Bates et al. (2004)
Simulated annealing	Max(dist)	Morris and Mitchell (1995)
Genetic algorithm	Min(L_2 -disc) ^b	Rainville et al. (2012)
Simulated annealing	Min(L_2 -disc), Max(dist)	Iooss et al. (2010)
Mixed integer linear programming	Min(corr) ^c	Hernandez (2008)
Florian's correlation reduction method	Min(corr)	Florian (1992)
Ranked Gram-Schmidt algorithm	Min(corr)	Owen (1994)
A heuristic algorithm	Min(corr), Min(L_2 -disc)	Cioppa and Lucas (2007)
Exchange algorithm	Max(ent) ^d	Jourdan and Franco (2010)
Columnwise-pairwise	Max(ent), Max(dist)	Ye et al. (2000)
Enhanced stochastic evolutionary algorithm	Max(ent), Min(L_2 -disc), Max(dist)	Jin et al. (2005)

^a Max(dist) = Maximize the inter-point distance.

^b Min(L_2 -disc) = Minimize L_2 -discrepancy.

^c Min(corr) = Minimize correlation.

^d Max(ent) = Maximize entropy.

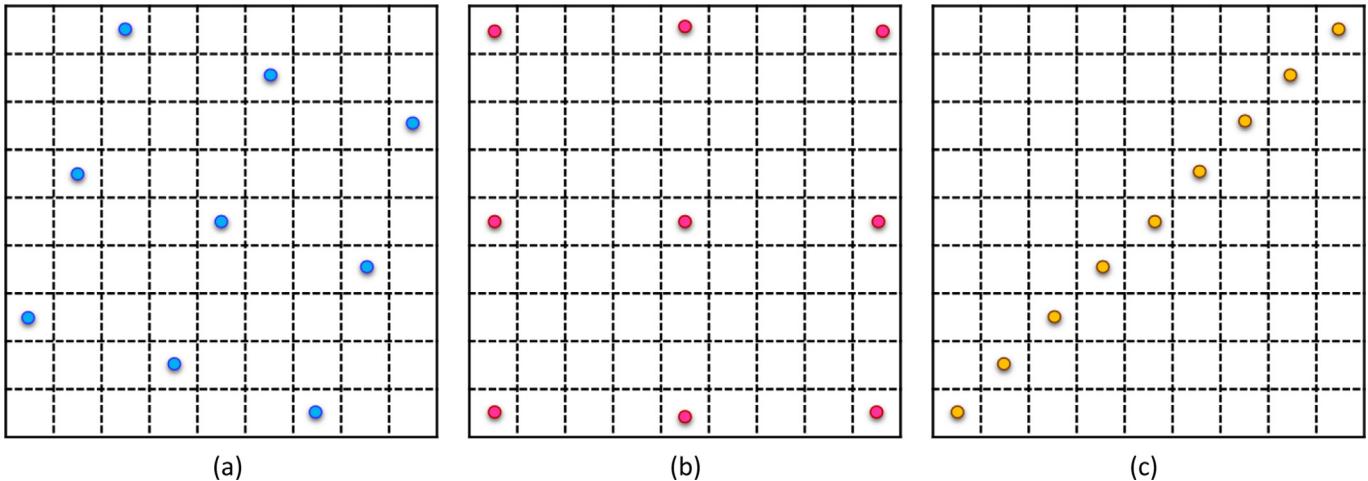


Fig. 2. Illustrative examples of different configurations of Latin hypercube and/or space-filling designs. (a) An optimal sample with respect to both Latin hypercube and space-filling properties, (b) An optimal sample with respect to only space-filling properties, and (c) An LHS with very poor space-filling properties and highly correlated factors.

possessing one-dimensional projection properties, its sample points are poorly scattered in the factor space.

2.3. Input-oriented sequential sampling

The input-oriented (also called model-free) sequential strategies sample the input space iteratively without any feedback from any resulting output space (model response). At each iteration, they aim to sample the input space as uniformly as possible based on some pre-specified criteria. Many strategies for input-oriented sequential sampling (hereafter called “sequential sampling” for simplicity) have been developed, while they can be classified under two general families of stochastic and deterministic strategies. Stochastic strategies extend the one-shot strategies by iteratively searching for new points that satisfy or optimize pre-specified criteria, typically by means of optimization and randomization (e.g. Vorechovský, 2009; Xiong et al., 2009). Deterministic strategies, however, utilize deterministic routines designed for space filling (e.g. Schretter et al., 2012). These strategies are very computationally efficient, while having certain drawbacks, as explained below.

The so-called low-discrepancy sequences (also known as quasi-random sequences) are among the most well-known deterministic sequential sampling strategies. These include the Halton (1960), Hammersley (1960), and Sobol' (Sobol', 1967) sequences, most of them utilize prime numbers as bases to generate sample points for each dimension. These sequences are only constrained by a low-discrepancy criterion to promote space-filling properties, with a caveat of possibly creating significant correlations between the factors, particularly in high-dimensional spaces (Loyola et al., 2016; Ong et al., 2012). The performance of low-discrepancy sequences has been extensively evaluated in the context of designs for computer experiments (e.g. Simpson et al., 2001; Kalagnanam and Diwekar, 1997).

The low-discrepancy sequences might have poor projection properties, in particular in high-dimensional spaces. Proper projection properties, also referred to as *non-collapsing* properties, are essential for effective sequential sampling (Pronzato and Müller, 2012; van Dam et al., 2007). A sample is said to possess projection properties when in any projection from its p -dimensional space to any lower dimensional sub-space, the sample points remain distinct from each other. This is vital in many sampling-based analysis. For illustration, Fig. 3a, b, and c depict a 2-

dimensional projection of the first 1,000 points of 100-dimensional, 10,000-point samples sequentially generated by Hammersley (HM), Halton (HS), and Sobol' sequences. Fig. 3f, g, and h show the same samples when transformed onto a standard normal distribution. As can be seen, there is a strong correlation between the sample points of the two factors (58th and 69th), leaving large regions un-sampled. It should be noted that the correlation problems on samples projected onto lower dimensional sub-spaces are not reflected in discrepancy measurements (Loyola et al., 2016). Typically, for high dimensions ($> \sim 20$), the sample size of low-discrepancy sequences should be quite large to ensure both space-filling and projection properties (Broad et al., 2015).

Various solutions have been proposed to resolve this issue, including using big prime numbers, leaping, and scrambling (see e.g. Kocis and Whiten, 1997). Fig. 3d and i show the improved performance of HS enabled with leaping and scrambling (HS-LS), where there is significant improvement in the dispersity of sample points but still there are some areas that remained un-sampled and some other areas with clusters of points. Fig. 3e and j show the superior performance of an enhanced Sobol' sequence by leaping and scrambling (Sobol'-LS), which has a high level of uniformity. Among these low-discrepancy samplings, we only used the Sobol'-LS as one of the benchmark strategies in this study.

Crombecq et al. (2011) in a comprehensive study on the state-of-the-art sequential sampling strategies showed that the strategies with both space-filling and projective properties outperform the ones that only have the space-filling properties. Furthermore, Gong et al. (2016) showed that among various sampling schemes, the Good Lattice Points (GLP) and Symmetric Latin hypercube (SLH) are the most efficient methods and have the highest uniformity scores compared with the Halton and Sobol' quasi-random sampling methods. Note that none of the existing sequential sampling strategies can explicitly consider and maintain projection (e.g., Latin hypercube) properties, and this was our motivation to develop Progressive Latin Hypercube Sampling (PLHS), as introduced in the next section.

3. Progressive Latin Hypercube Sampling

3.1. Definition

In a p -dimensional space, let $\mathbf{S}_k(n_k, p)$, where $k = 1, 2, \dots, T$, be a series of samples, each with a size of n_k , and let \mathbf{L}^t be a new sample

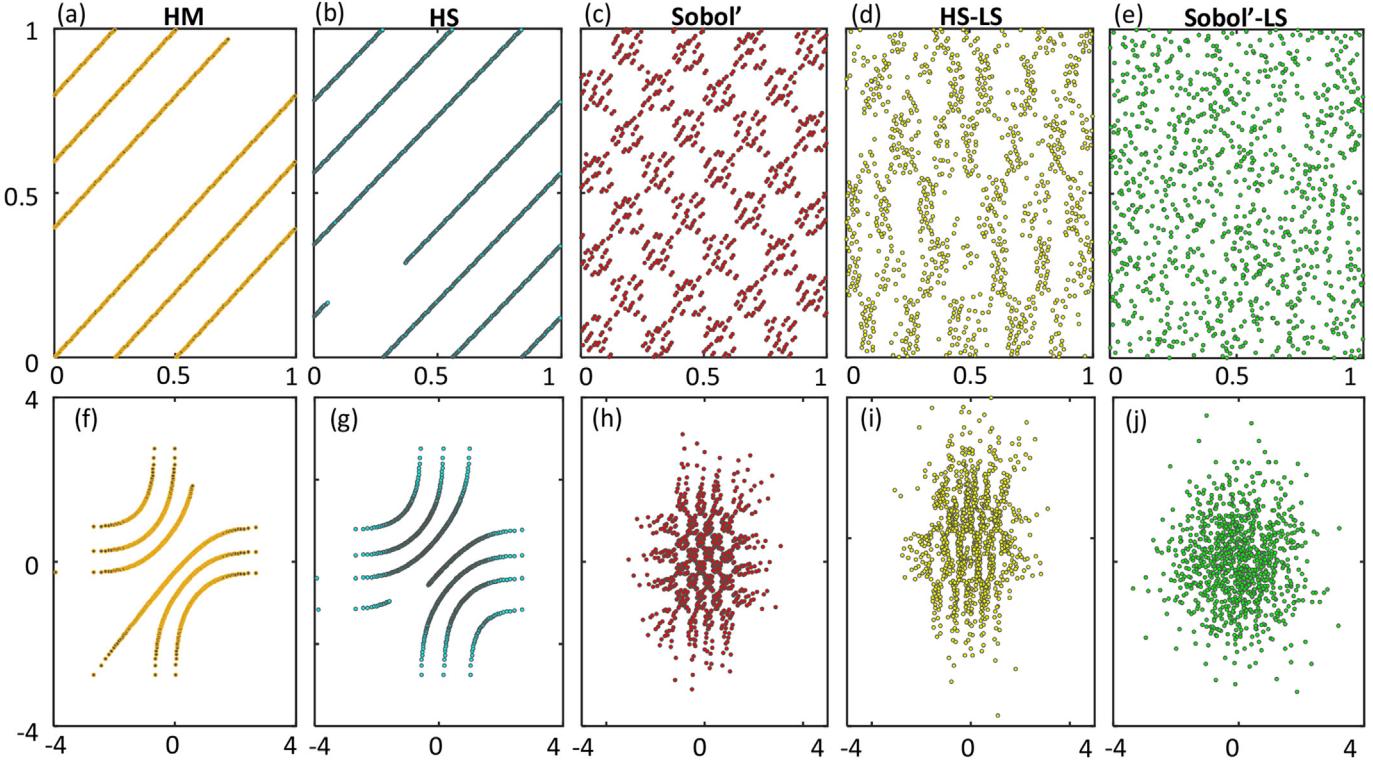


Fig. 3. Example performances of existing sequential sampling strategies in a 100-dimensional space when projected onto a 2-dimensional sub-space. Performance of (a) Hamersley, (b) Halton, (c) Sobol', and improved (d) Halton and (e) Sobol' sequences (by leaping and scrambling) in a uniform distribution case. The plots in the bottom panel shows the same samples of plots above when transformed into a standard normal distribution space. All samples are projected onto dimensions 58 (horizontal axis) and 69 (vertical axis).

set formed by the union of these (sub-) samples such that $\mathbf{L}^t = \cup_{k=1}^t \mathbf{S}_k(n_k, p)$, where $t = 1, 2, \dots, T$. We call sample \mathbf{L}^t to be *progressive Latin hypercube* if and only if for any t , \mathbf{L}^t is a Latin hypercube. In the case that $t = T$, we denote the new sample by $\text{PLHS}(n, p, T)$ where n is the summation of the slice sizes, $n = \sum_{k=1}^T n_k$. As such, $\text{PLHS}(n, p, T)$ is a Latin hypercube sample consisting of T sub-samples (also called *slices* hereafter) that their progressive union (\mathbf{L}^t from $k = 1$ to $k = T$) holds Latin hypercube properties.

3.2. Mathematical formulation

Given that PLHS is an extension of LHS, in the following, we first reformulate LHS. Unlike the classic approach that considers LHS as a combinatorial problem, we formulate a real-valued problem. Let $\mathbf{S}(n, p)$ be a sample matrix, which consists of $n \times p$ elements (variables) $x_{i,j} \in [0, 1]$ where $i = 1, \dots, n$ and $j = 1, \dots, p$. Also, consider the factor space $[0, 1]^p$ divided into n disjoint intervals (strata or bins) $[0, 1/n], [1/n, 2/n], \dots, [(n-1)/n, 1]$ indexed by q ($q = 1, \dots, n$) along each axis/dimension. We define a new set of auxiliary binary variables, $y_{q,j}$, such that

$$y_{q,j} = \begin{cases} 1 & \text{If there exist any } i \text{ for which} \\ & x_{i,j} \text{ lies in the interval } q \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Then $\mathbf{S}(n, p)$ is said to be Latin hypercube when the following condition is satisfied

$$\frac{\sum_{j=1}^p \sum_{q=1}^n y_{q,j}}{n.p} = 1 \quad (2)$$

The left hand side of Eq. (2) is essentially a function of the sample matrix, $F(\mathbf{S}(n, p))$, that varies between 1 (when the sample is Latin hypercube) and $1/n$ (when all sample points are located in a single interval at every dimension).

To extend the proposed formulation to PLHS, we define a set of auxiliary binary variables as $y_{q,j}^t$, corresponding to the sample matrix $\mathbf{L}^t = \cup_{k=1}^t \mathbf{S}_k(n_k, p)$, where $t = 1, 2, \dots, T$ represents the slice number. Note that the number of sample points in \mathbf{L}^t is $n_t = \sum_{k=1}^t n_k$, and the factor space $[0, 1]^p$ is divided into n_t disjoint intervals $[0, 1/n_t], [1/n_t, 2/n_t], \dots, [(n_t-1)/n_t, 1]$ indexed by q ($q = 1, \dots, n_t$) along each axis/dimension. Then the following equation is the necessary and sufficient condition for a sample to be said progressive Latin hypercube

$$\sum_{t=1}^T \frac{\sum_{j=1}^p \sum_{q=1}^{n_t} y_{q,j}^t}{n_t \cdot p} = T \quad (3)$$

The left hand side of Eq. (3) is the summation of $F(\mathbf{L}^t(n_t, p))$ from Eq. (2) for $t = 1, \dots, T$, $\sum_{t=1}^T F(\mathbf{L}^t(n_t, p))$. Now, generation of PLHS via the above mathematical formulation can be seen as an optimization problem, as follows

$$\text{Maximize} \left(\sum_{t=1}^T \frac{\sum_{j=1}^p \sum_{q=1}^{n_t} y_{q,j}^t}{n_t \cdot p} \right) \quad (4)$$

with $x_{i,j}$ as decision variables and $y_{q,j}^t$ as auxiliary variables.

3.3. Practical implementation

The real-valued optimization problem of Eq. (4) can be solved via various optimization solvers to generate a progressive Latin hypercube sample. As an alternative to directly solving this optimization problem, we also introduce two heuristic algorithms that efficiently construct PLHS for practical applications. The first algorithm, referred to as “doubling procedure” in this study, is an iterative approach which constructs a PLHS such that, at each step, the size of the sample is doubled (Section 3.3.1). Hence, the doubling procedure has limited flexibility in sample size. To gain flexibility in sample size, we develop the second algorithm for generating the PLHS, using an optimization approach (Section 3.3.2). Any of these algorithms can be used in conjunction with other desired criteria such as maximum space filling and/or minimum correlation in addition to PLHS properties to improve the overall quality of sampling. Section 3.3.3 explains a range of criteria that can be used during sampling or to test the quality of an already generated sample set.

3.3.1. An iterative approach: doubling procedure (perfect-PLHS)

Perhaps, the only possible way to iteratively generate PLHS – adding new slices to an already generated LHS – is through the following algorithm. In this algorithm, the user chooses the size of the first slice S_1 (i.e., n_1), and then the size of the second slice S_2 will be $n_2 = n_1$, resulting in a sample L_2 of size $2n_1$. Subsequently, the size of the third slice S_3 will be $n_3 = 2n_1$, resulting in a sample L_3 of size $4n_1$ and so on. This means the size of sample (L_j) in this algorithm grows geometrically, as $n_1 \times 2^{(j-1)}$. Here we use an example shown in Fig. 4 to clarify this and explain how the algorithm works. With no loss of generality and for the sake of simplicity, this example is designed in a 2-dimensional space. Fig. 4a shows the first slice with $n_1 = 3$, which is Latin hypercube. In Fig. 4b, each of the three intervals for each dimension is divided into two equal intervals, and in Fig. 4c, three new sample points are added to create a Latin hypercube sample of size 6. Again, in Fig. 4d, each of the six intervals at each dimension is divided into two equal

intervals resulting in 12 disjoint intervals for each dimension, and in Figs. 4e, 6 new sample points are added such that the resulting 12-point sample is Latin hypercube. This doubling procedure of sample size at each slice can be continued until obtaining the required sample size (3, 6, 12, 24, and so on). A similar approach to the above algorithm was proposed by Sallaberry et al. (2008), called the “two-multiple” algorithm, and further developed by Williamson (2015) for uncertainty quantification.

3.3.2. An optimization approach (quasi-PLHS)

This approach utilizes an existing sampling strategy as a building block, called Sliced Latin hypercube Sampling (SLHS), originally developed by Qian (2012) and further enhanced by Ba et al. (2015) and Chen et al. (2016). The SLHS is a special type of LHS that generates a Latin hypercube sample (say with size n) formed by a collection of smaller, equally sized Latin hypercube samples (say with size $m = n/T$ where T is the number of sub-samples). Mathematically stated, let $LHS_k(m, p)$ for $k = 1, 2, \dots, T$ be a set of sample matrices and S^t be a new sample set formed by the aggregation of these sub-samples such that $S^t = \cup_{k=1}^t LHS_k(m, p)$, where $1 < t \leq T$. A sample matrix is $SLHS(n, p, T)$ if and only if S^T , the union of all sub-samples, is Latin hypercube (i.e., $LHS(n, p)$ where $n = m \times T$). As such, although the entire sample is Latin hypercube, the progressive addition of the sub-samples may not be Latin hypercube. This is the fundamental difference between SLHS and our proposed PLHS. The construction procedure of the SLHS is presented in Appendix A.

Utilizing SLHS, we developed an algorithm to approximately generate (quasi-) PLHS efficiently. The algorithm has two main steps. In the first step, SLHS is used to generate a set of T slices. In the second step, the order (arrangement) of these slices is permuted to maximize one-dimensional projection properties when the slices are progressively combined. In other words, in this step, we search for an optimal permutation (ordering) of the slices to maximize objective function defined by Eq. (4). The two steps can be repeated (with different initial random seed for SLHS) until a desirable sample is found.

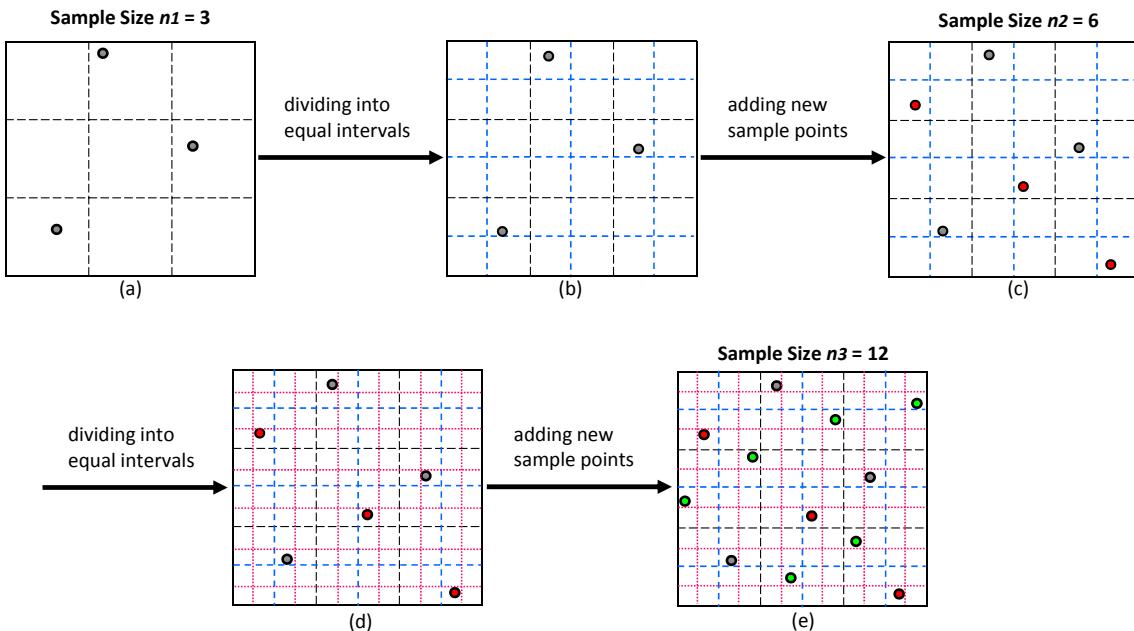


Fig. 4. A doubling procedure of sample size for generating perfect-PLHS with $n = 12$ and $p = 2$: (a) An initial LHS with 3 sample points, (b) Dividing an initial sample domain into 6 intervals with equal marginal probability (c) The second slice with 6 sample points, (d) Dividing the second slice into 12 intervals with equal marginal probability, and (e) The third slice with 12 sample points.

Mathematically, we define a permutation π as a bijective function from $\{1, 2, \dots, T\}$ to $\{1, 2, \dots, T\}$. It is convenient to think of permutation π as a sequence $\{\pi(1), \pi(2), \dots, \pi(T)\}$. For example, if $\pi = \{2, 1, 3\}$ then $\pi(1) = 2$, $\pi(2) = 1$, and $\pi(3) = 3$; hence, $\pi(k)$ is the k th element of this sequence. Let Π be a set of random permutations of the integers $\{1, 2, \dots, T\}$. Then, the problem is finding an optimal permutation of slices, $\pi^* \in \Pi$, to maximize $F(\bigcup_{k=1}^T \mathbf{LHS}_{\pi^*(k)}(m, p))$, for $t = 1, \dots, T$ (Eq. (4)).

This optimization formulation is an ordering problem of Latin hypercube slices to maximize PLHS properties. Here, we employ a greedy search algorithm explained in the following to find a near optimal solution to this optimization problem. Note that in this algorithm, the initial slice $\pi^*(1)$ is chosen randomly.

Algorithm 1

- Select sample size n , number of dimensions/factors p , and total number of slices T .
 - Generate a set of T slices using SLHS algorithm, $\mathbf{SLHS}(n, p, T) = \bigcup_{k=1}^T \mathbf{LHS}_k(m, p)$
 - Randomly arrange slices in a sequence, $\{\pi(1), \pi(2), \dots, \pi(T)\}$.
 - Select an integer, r , randomly, between 1 and T , and set $k = 1$, $\pi^*(1) = r$, and $\mathbf{S} = \mathbf{LHS}_{\pi^*(1)}(m, p)$.
 - Mark the r th slice as visited and the other $T-1$ slices as unvisited.
- while** k is smaller than T
- Set $F_{\text{best}} = 0$ and $k = k + 1$.
 - For** $j =$ all unvisited slices
 - Construct a new set as $\mathbf{S}^{\pi(j)} = \mathbf{S} \cup \mathbf{LHS}_{\pi(j)}(m, p)$.
 - Evaluate the objective function value using Eq. (4), $F(\mathbf{S}^{\pi(j)})$.
 - If** $F(\mathbf{S}^{\pi(j)}) > F_{\text{best}}$
 - $F_{\text{best}} = F(\mathbf{S}^{\pi(j)})$
 - $\pi^*(k) = \pi(j)$ - end if** - end for**
 - Mark the $\pi^*(k)$ -th slice as visited and the other $(T-k)$ slices as unvisited.
 - $\mathbf{S} = \mathbf{S} \cup \mathbf{LHS}_{\pi^*(k)}(m, p)$.
- end while**
- Set $\mathbf{PLHS}(n, p, T) = \bigcup_{k=1}^T \mathbf{LHS}_{\pi^*(k)}(m, p)$

Algorithm 1. The pseudo-code of the greedy search for generating quasi-PLHS.

Algorithm 1 is a straightforward heuristic algorithm which is based on the so-called nearest neighbor greedy heuristic. Using **Algorithm 1**, we construct a permutation of slices $\pi^* = \{\pi^*(1), \pi^*(2), \dots, \pi^*(T)\}$ with the initial slice $\pi^*(1)$ chosen randomly. In general, the next $\pi^*(k)$ is selected such that it maximizes F at each stage. The proposed procedure can be further improved if we repeat the algorithm by running it for different initial SLHS trials and then choosing the best PLHS among them. For each experiment in this study, we ran the algorithm 100 times with different random seeds and reported the best sample found based on the objective function. Note that in general, the choice of m and T is rather arbitrary. However, to generate a sample with size $T \times m$, the algorithm is more computationally efficient for smaller numbers of slices, T . For example producing $T = 10$ slices with $m = 1,000$ samples in each slice is much more efficient than producing $T = 1,000$ slices with $m = 10$.

3.3.3. Other criteria

Like many other sampling strategies, a range of secondary criteria is available to be used in conjunction with the primary criteria such as the PLHS requirements (i.e. progressively attaining the projection

properties). Any of these criteria can be optimized during sampling or used to evaluate an already generated sample. Below, we outline three possible criteria that work for maximum space-filling, lower discrepancy, and minimum pairwise correlation.

The maximin distance criterion is a measure of space-filling, which aims to maximize the minimum distance among every pair of points in a sample, i.e.

$$\text{Maximize} \left(\min_{i,j=1,2,\dots,n \atop i \neq j} \{d(\mathbf{X}^i, \mathbf{X}^j)\} \right) \quad (5)$$

where $d(\dots)$ is a distance measure (here the Euclidean measure) and

n is the sample size.

The maximin criterion is mainly intended to improve space-filling properties to ensure that generated sample points are evenly spread across the entire factor space – i.e., sample points are located almost equally far apart. This optimality criterion has reportedly worked well to ensure adequate space-filling (Santner et al., 2003).

The so-called L2-star discrepancy is a metric to evaluate the uniformity and discrepancy of sample points in an input space. Given a set of n points $\mathbf{S} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}^T$ with $\mathbf{X}_i \in \mathbb{R}^p$ in the p -dimensional unit cube $C^p = [0, 1]^p$, this metric can be approximately calculated by (Warnock, 1972)

$$D^{L2}(\mathbf{S}) = \left(\frac{1}{3} \right)^p + \frac{1}{n^2} \sum_{k=1}^n \sum_{i=1}^n \prod_{j=1}^p \left[1 - \max(x_{kj}, x_{ij}) \right] - \frac{2^{1-p}}{n} \sum_{i=1}^n \prod_{j=1}^p (1 - x_{ij}^2) \quad (6)$$

where x_{ij} is the j th coordinate of the i th point (\mathbf{X}_i) in the sample set

S. This metric varies between 0 and 1, and lower discrepancy values indicate better spread of points in the input space. In addition to the L_2 -star discrepancy, other types of discrepancy metrics are available in the literature such as centered discrepancy, symmetric discrepancy, and wrap-around discrepancy (Hickernell, 1998; Gong et al., 2016). Here, we used the L_2 -star discrepancy (Eq. (6)), which is a popular metric representing the overall uniformity in the high-dimensional space (Niederreiter, 1992).

Finally, we use the maximum pairwise correlation as a standard measure of linear dependence between two variables. This measure is based on the Pearson product-moment correlation coefficient. For any two columns ($i \neq j$) of a sample matrix, it can be calculated by

$$\rho_{ij} = \frac{\sum_{k=1}^n (x_{k,i} - \bar{x}_{:,i})(x_{k,j} - \bar{x}_{:,j})}{\sqrt{\sum_{k=1}^n (x_{k,i} - \bar{x}_{:,i})^2 \sum_{k=1}^n (x_{k,j} - \bar{x}_{:,j})^2}} \quad (7)$$

where $\bar{x}_{:,i} = \sum_{k=1}^n \frac{x_{k,i}}{n}$, and $\bar{x}_{:,j} = \sum_{k=1}^n \frac{x_{k,j}}{n}$.

The maximum absolute value of ρ_{ij} for all pair-wise combinations of factors is commonly used as a measure of sample quality and denoted by ρ_{\max} (Cioppa and Lucas, 2007). Therefore, a sample with smaller ρ_{\max} is deemed to be of higher quality.

In our experiments, we utilized the maximin distance criterion directly during sampling. We also used the L_2 -star discrepancy and maximum pairwise correlation criteria to independently evaluate the generated samples via different strategies. In PLHS, the maximin distance criterion for every slice as well as for the entire sample were calculated and aggregated via a weighting approach.

4. Computational experiments

We used two test problems and designed four experiments to evaluate the performance of PLHS against other sampling methods in the input space in terms of space-filling, correlation, and projective properties (first experiment), and in the output space in terms of the statistical measures and sensitivity metrics (last three experiments). The two test problems and the experimental setup are described as follows.

4.1. Test problems

4.1.1. A 2-D analytical function

We compared the performance of the different sampling algorithms on a two-dimensional toy function, Y , defined by

$$Y(x_1, x_2) = 2(x_1)^2 + 3(x_2)^2 + x_1 \cdot x_2 \quad (8)$$

where x_1 and x_2 are random variables uniformly distributed in interval $[-1, 1]$. This function, adapted from the analysis of Razavi and Gupta (2015), is a quadratic function with an interaction term (the third term).

4.1.2. A real world problem

We employed the HYMOD conceptual hydrologic model (Fig. 5) to assess PLHS in real-world problems. HYMOD has five parameters that need to be specified/calibrated by the user (Table 2). Details of HYMOD can be found in Boyle (2001) and Wagener et al. (2001). The HYMOD model used in this study was adopted from Vrugt et al. (2003), developed for the Leaf River watershed located north of Collins, Mississippi, USA. We used the Nash-Sutcliffe metric on streamflows (NS) as well as on the logarithm of streamflows (NS-log) as the model outputs.

We ran the model with 500,000 randomly generated parameter sets (uniformly distributed in the parameter ranges of Table 2) by original LHS to generate the “true” cumulative distribution functions (CDFs) of the model responses. Fig. 6 shows the CDFs for NS and NS-log metrics.

4.2. Design of experiments

4.2.1. Experiment I

We designed the first experiment to evaluate PLHS in terms of achieving the maximum stratification when projected onto univariate margins using the proposed objective function (Eq. (4)) in 2-, 5-, and 100-dimensional input spaces. Moreover, in this experiment the quality of different sampling strategies were compared using the low-discrepancy (Eq. (6)) and maximum pairwise correlation (Eq. (7)) as the performance measures.

4.2.2. Experiment II

One of the frequent uses of sampling is in Monte Carlo simulation. This experiment investigated the effectiveness of PLHS in Monte Carlo-based estimation of the first- and second-order moments (mean and variance) of variable $Z = Y(x_1, x_2)$ as a function of two uniformly distributed variables, defined by Eq. (8). Here, the performance of sampling methods were evaluated based on the deviation (error) of the estimates of the mean $E(Z)$, and variance, $Var(Z)$, from their true values, which can be analytically calculated as $E(Z) = 1.6667$ and $Var(Z) = 1.2667$.

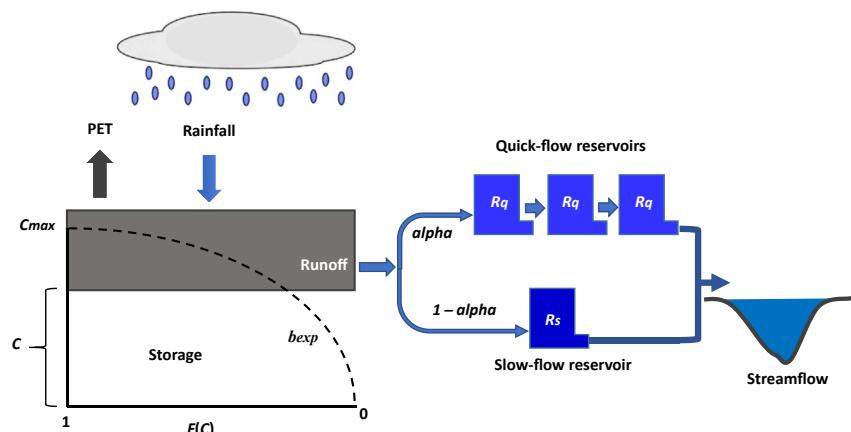


Fig. 5. The structure of the HYMOD rainfall-runoff model that consists of a soil moisture module (parameters: b_{exp} and C_{\max}) and a routing module (parameters: α , R_s , and R_q).

Table 2

Description of the HYMOD parameters.

Parameter	Range	Unit	Description
C_{\max}	1.00–500.00	[mm]	Maximum storage capacity
b_{\exp}	0.10–2.00	[–]	Degree of spatial variability of the soil moisture capacity
α	0.10–0.99	[–]	Factor distributing the flow between two series of reservoirs
R_q	0.10–0.99	[day]	Residence time of the quick release reservoirs
R_s	0.00–0.10	[day]	Residence time of the slow release reservoir

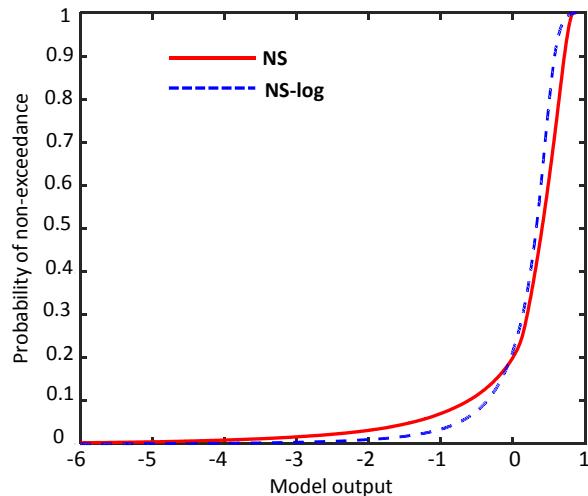


Fig. 6. The true cumulative distribution functions (CDFs) of the HYMOD model outputs: Nash-Sutcliffe metrics on streamflows (NS) and on the logarithm of streamflows (NS-log).

4.2.3. Experiment III

In this experiment we assessed how PLHS works to improve the convergence and robustness of global sensitivity analysis (GSA). We used a recently developed GSA framework, known as Variogram Analysis of Response Surfaces (VARS), proposed by Razavi and Gupta (2016a). VARS is a general framework that utilizes directional variogram and covariogram functions to characterize the full spectrum of sensitivity-related information, thereby providing a comprehensive set of global sensitivity metrics with minimal computational cost. VARS generates a new set of sensitivity metrics called IVARS (Integrated Variogram Across a Range of Scales) that summarize the variance of change (rate of variability) in model response at a range of perturbation scales in the parameter space. VARS also generates the Sobol' (variance-based) total-order effect and the Morris (derivative-based) elementary effects. Here, we utilized the STAR-VARS implementation of VARS developed in Razavi and Gupta (2016b). It has been shown that STAR-VARS is highly efficient and statistically robust, providing stable results within 1–2 orders of magnitude smaller numbers of sampled points (model runs), compared with the original Sobol' and Morris approaches (Razavi and Gupta (2016 a,b)).

STAR-VARS utilizes a star-based sampling, which consists of two elements: (1) Latin hypercube sampling to identify star centers, and (2) a structured-sampling approach to identify star points. In this experiment, we replaced the first element by PLHS (and the benchmark sampling algorithms) to generate star centers. The sample size reported in this experiment refers to the number of star centers taken by the first element above. The total number of function evaluations (the total sample including sample points from the second element) were 2,300 and 950 for the HYMOD and 2-dimensional case studies, respectively. We set Δh (VARS

resolution parameter) to 0.1 (as recommended in Razavi and Gupta (2016b)). We compared the performance of VARS enabled with different sampling algorithms in generating IVARS₁₀ (integrated variogram in range 0–10% of the parameter range), IVARS₅₀ (integrated variogram in range 0–50% of the parameter range), and VARS-T0 (VARS-derived Total-Order effect). These global sensitivity metrics are calculated simultaneously in a VARS run. The global sensitivity of NS to the five HYMOD model parameters and two-dimensional test function were assessed. The “true” values of these global sensitivity metrics for this case study were adopted from Razavi and Gupta (2015; 2016b).

4.2.4. Experiment IV

This experiment was designed to evaluate PLHS in an uncertainty analysis context, where of interest is to understand how uncertainty in model inputs propagates to the model output via Monte Carlo simulations. With no lack of generality, we tested how PLHS works to approximate the distribution of the output of the HYMOD model. This experiment has fundamental elements in common with GLUE-type analyses, and was also intended to reflect how PLHS could improve the efficiency and robustness of GLUE. The goal here was to assess how *accurate* and *robust* PLHS could approximate the true distribution of the model output as the sample size grows. The accuracy was assessed by comparing the approximate CDF with the true CDF using two similarity metrics that measure the difference between distributions of the random variables.

The first metric is the Kolmogorov-Smirnov (K-S) distance measure (Kolmogorov, 1933). As a measure of similarity, the K-S calculates the maximum distance between the two CDFs as follows

$$K-S = \max_y |CDF_{true}(y) - CDF_{appx}(y)| \quad (9)$$

where y is the model output (NS or NS-log), and CDF_{true} and CDF_{appx} are the true and approximated distributions of the model output.

The second metric is called energy distance which characterizes the equality of two distributions (Székely and Rizzo, 2005). Suppose that X and Y are independent sets of real-valued random variables with cumulative distribution functions of CDF_{true} and CDF_{appx} , respectively. The energy distance is the squared root of

$$D^2(CDF_{true}, CDF_{appx}) = 2E \|X - Y\| - E \|X - X'\| - E \|Y - Y'\| \quad (10)$$

where $\|\cdot\|$ is the Euclidean norm; X' denotes an independent and identically distributed (iid) copy of X ; and similarly, Y and Y' are iid. Mathematically, it has been proven that the Energy distance is twice the Cramér's distance (Székely and Rizzo, 2013), i.e.

$$D^2(CDF_{true}, CDF_{appx}) = 2 \int_{-\infty}^{\infty} (CDF_{true}(y) - CDF_{appx}(y))^2 dx \quad (11)$$

Table 3

Setup summary of computational experiments.

Case study	Sample size (n)	Number of slices (T)	Slice size (m)	Dimension (p)
Experiment I	1,000	100	10	2-D
	1,000	10	100	5-D
	1,000	10	100	100-D
Experiment II	1,000	100	10	2-D
	50	10	5	2-D
Experiment III	50	10	5	5-D
	500	10	50	5-D
Experiment IV				

In this experiment, we used normalized version of the energy distance, D^2_N , which can be estimated by dividing D^2 by an estimate off $\|X - Y\|$. For more details on energy distance, see Székely and Rizzo (2013, 2005) and Aslan and Zech (2005).

The K-S and D^2_N values vary between zero and one, with zero indicating the two distributions are identical. In this experiment, we compared different sampling strategies based on how rapidly their values of K-S and D^2_N converge to zero by growing the sample size.

4.3. Setup for sampling strategies

The two developed PLHS algorithms, through doubling procedure (called perfect-PLHS) and optimization approach (called quasi-PLHS), were tested in four different experiments explained above. For comparison purposes, the same experiments were also carried out with random sampling (RAND), original LHS, Sobol' sequence, and SLHS. Table 3 summarizes the experimental setup for each case study. For RAND and LHS techniques, in order to construct samples sequentially, we simply generated sample points at each new slice and added them to previous ones, regardless of the properties of previously generated slices. It should be noted that the samples generated for Experiment I were also used for numerical simulations in Experiment II. For Sobol' sequence (enabled with leaping and scrambling), RAND, and original LHS techniques, we used the built-in functions of MATLAB to generate the samples and for the SLHS technique, we employed the method introduced by Ba et al. (2015) (see Appendix A). Moreover, for each replicate of LHS, SLHS, and PLHS algorithms, we ran 100 trials with different

random seeds and reported the best sample according to the maximin criterion. Also, to ensure a fair comparison, we accounted for sampling variability due to randomness in the comparisons by carrying out 100 replicates of each experiment. This allowed us to see a range of possible performances for each algorithm and to assess their robustness against their random components.

5. Results and discussion

5.1. Performance evaluation in the input space

Fig. 7 compares the average objective functions (Eq. (4)), F , as the sample size grows for different sampling methods obtained from the 100 replicates (to normalize this value, it is divided by the total number of slices T). The results are based on 2, 5, and 100-dimensional input spaces. The value of F associated with perfect-PLHS remains at one for all the slice numbers (not shown), and that of quasi-PLHS (using Algorithm 1) is higher than those produced by RAND, LHS, Sobol' and SLHS. This indicates that PLHS better preserves the projective properties and achieves the best stratification. Importantly, the F values of RAND and LHS tend to degrade by increasing the sample size, suggesting they might not be appropriate for sequential sampling. As shown in Fig. 7, in lower dimension the Sobol' sequence is better than RAND and LHS; however, by increasing the dimensionality of the space to 100, it performs almost like the RAND technique in terms projection properties.

Fig. 8 compares the discrepancy metric D^{L2} of different sampling strategies defined by Eq. (6) for the 100-dimensional input space.

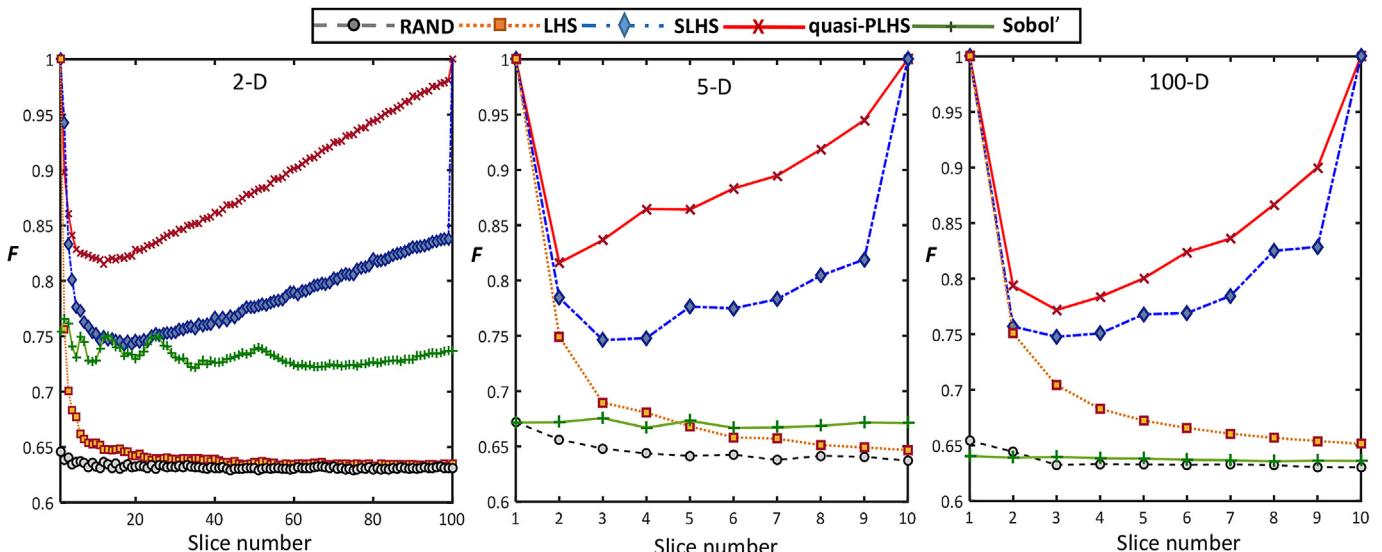


Fig. 7. Comparison of different sampling algorithms in preserving one-dimensional projection properties as the sample size grows (the average of objective function defined by Eq. (4) over 100 replicates) – the objective function value of one indicates perfect performance (perfect-PLHS not shown as it remains on one for all slice numbers).

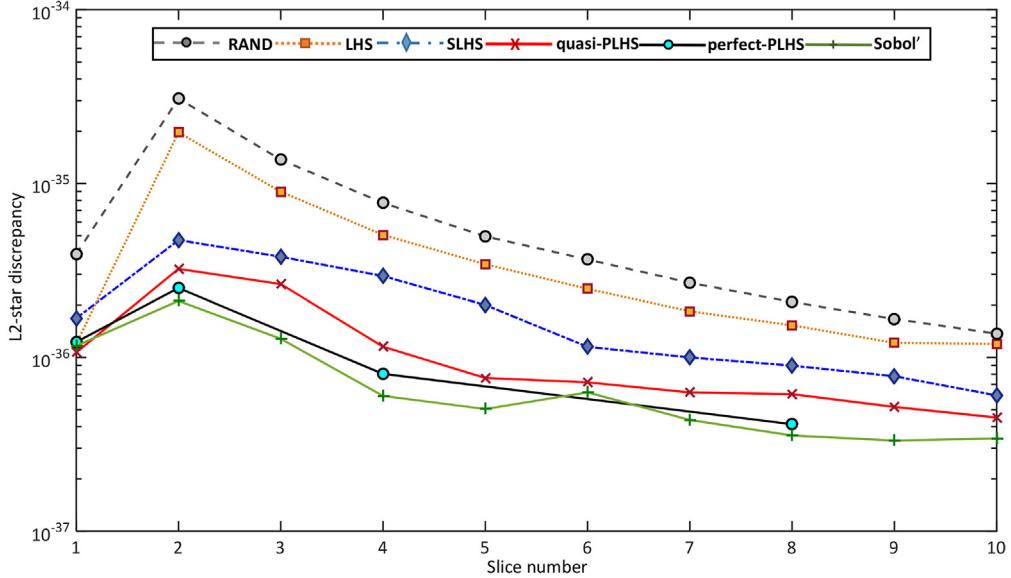


Fig. 8. Comparison of different sampling strategies in uniformly spreading sample points based on the discrepancy metric (Eq. (6)) – The results are average of 100 replicates for the 100-dimenasional case – a lower discrepancy metric indicates a better dispersion of sample points.

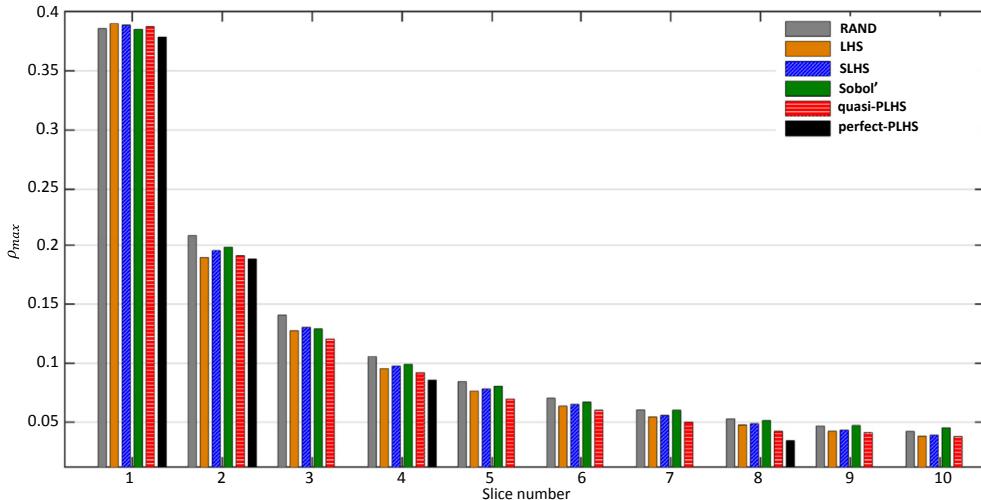


Fig. 9. Comparison of different sampling strategies in terms of maximum pairwise correlation between the factors (Eq. (7)) – The results are average of 100 replicates for the 100-dimenasional input space.

Here, the perfect-PLHS was constructed by doubling the initial sample size of 100, resulting in 100, 200, 400, and 800 sample points, equivalent to slice numbers 1, 2, 4, and 8 of the other sampling strategies. As can be seen, the Sobol' sequence has the lowest D^{L2} discrepancy values, which is comparable with the perfect-PLHS. The best performance of Sobol' is due to the fact that this method is inherently constrained by a low-discrepancy criterion. Then, the quasi-PLHS is the superior algorithm, as it consistently has the lower discrepancy values for any sample size. SLHS comes fourth, followed by LHS and RAND.

The maximum pairwise correlations (ρ_{max}) for different sampling strategies in the 100-dimenasional input space are illustrated in Fig. 9. Similar to Fig. 8, the perfect-PLHS was constructed with total sample size of 800 (equivalent to slice numbers 1, 2, 4, and 8). Overall, at each slice number, the performance of all the algorithms are pretty close, with the perfect- and quasi-PLHS strategies having the minimum (best) ρ_{max} of all. The results of RAND, LHS, and SLHS

are comparable in terms of ρ_{max} in most cases, with the exception of the last slice, where SLHS demonstrates considerable improvement. Furthermore, the Sobol' method has the higher ρ_{max} values compared to the SLHS and LHS.

5.2. Estimating the mean and variance of the 2-D problem

Fig. 10 shows the performance of the different sampling algorithms, as their sample size grows, in estimating the mean and variance of the 2-dimensional test problem described by Eq. (8). The average error of these estimates were computed as the proportion of the “true” mean (1.6667) and variance (1.2667). According to Fig. 10a and b, the Sobol', perfect- and quasi-PLHS methods outperformed the other algorithms, as they resulted in the lowest average error (over 100 replicates) in estimating both of the mean and variance. To further scrutinize this observation, Fig. 10c and d show the standard deviation of the estimates of the

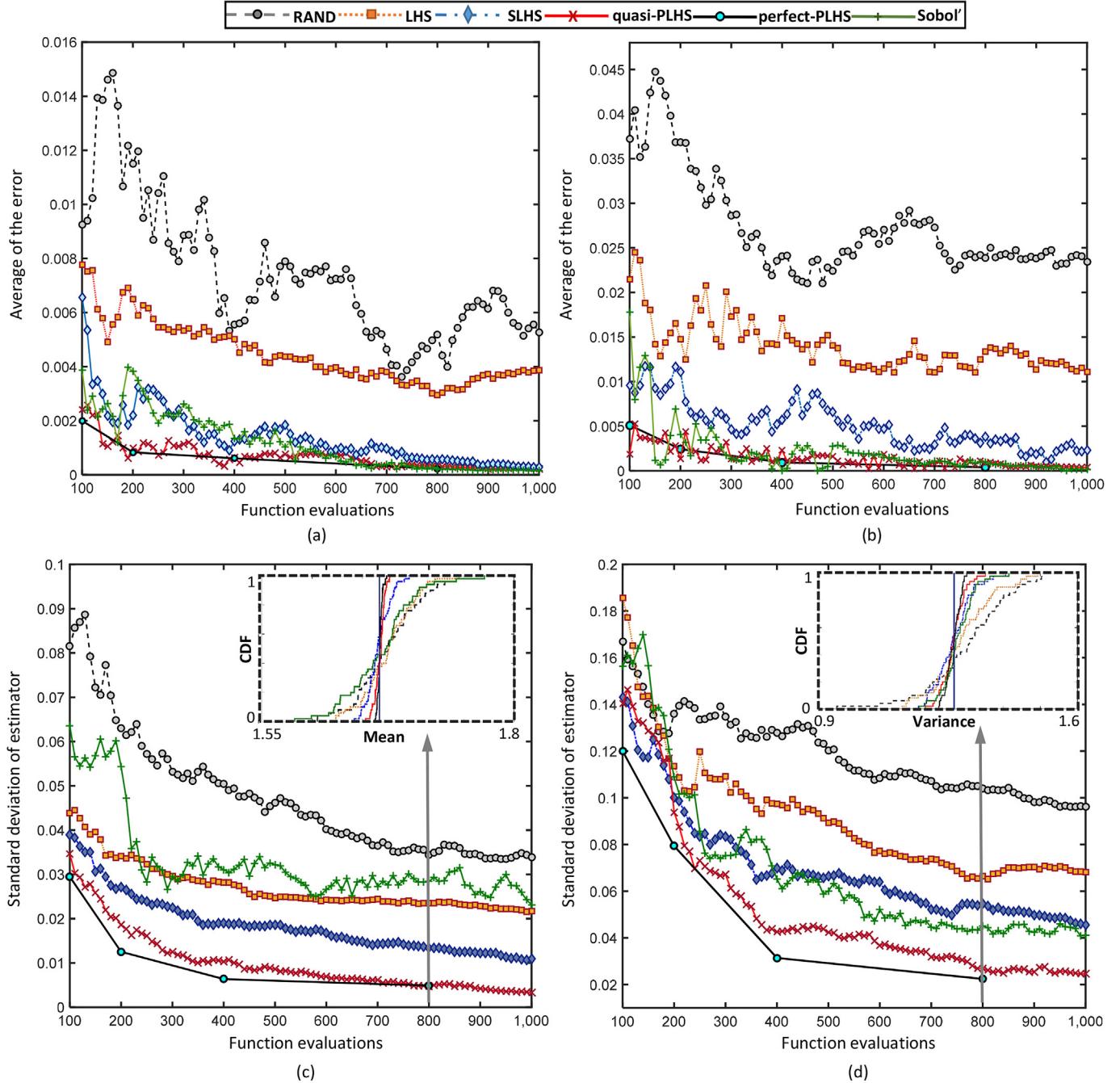


Fig. 10. Case study 2: Comparison of different sampling strategies in estimating the mean (a & c) and variance (b & d) of the 2-dimensional problem. In (a) and (b), the deviations (errors) from the true mean and true variance were averaged over 100 replicates. In (c) and (d), the standard deviation of the estimated mean and variance (regardless of the true values) over the 100 replicates were calculated.

mean and variance over the 100 replicates. Both perfect- and quasi-PLHS resulted in the least standard deviation among all the algorithms for any given number of function evaluations; however, the Sobol' method has the higher values of standard deviations. SLHS performs better than LHS and RAND in these analyses. Moreover, the CDFs of the estimates are compared in Fig. 10c and d for the 80th slice, i.e. after 800 function evaluations. These results indicate the superiority of the proposed variants of PLHS compared with the alternatives (1) in approximating statistics of a model response surface with fewer function evaluations, and (2) in terms of robustness against sampling variability and randomness, as with

perfect- and quasi-PLHS, the estimates of the mean and variance across the 100 replicates with different random seeds were significantly closer to each other.

5.3. Comparison in a sensitivity analysis context

Fig. 11 and Fig. 12 demonstrate the performance of different sampling algorithms in global sensitivity analysis of the test problems. The 5th and 95th percentiles (90% interval) of the 100 replicates of this experiment with different random seeds were investigated to gain a view of the range of possible performances of

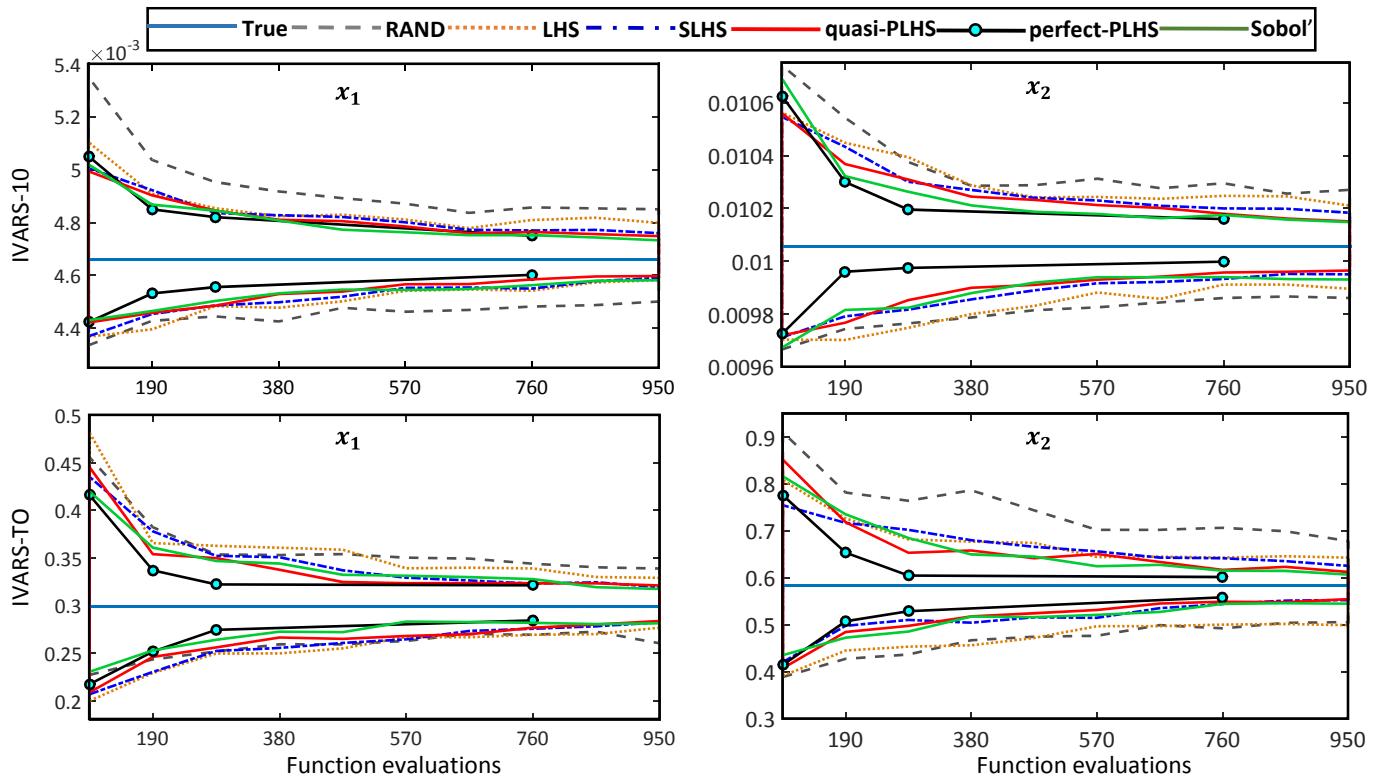


Fig. 11. Comparison of different sampling strategies in global sensitivity analysis of 2-D problem using the VARS method. The 5th and 95th percentiles of the 100 replicates are shown along with true values. Here, the IVARS₁₀ and VARS-T0 (Total-Order effect) metrics were illustrated.

different algorithms. This also helped to assess the robustness of the sensitivity analysis to sampling variability. The first test problem has only two input variables (x_1 and x_2) where parameter x_2 is more sensitive than x_1 according to all used sensitivity metrics. The second test problem has 5 parameters, and their ranks according to IVARS₅₀ and VARS-T0 are as follows R_q , C_{\max} , α , b_{\exp} , and R_s . Results of IVARS₁₀ and VARS-T0 for test problem 1 and IVARS₅₀ and VARS-T0 for test problem 2 were (arbitrarily) chosen to be shown.

As can be seen, for the first test problem perfect-PLHS, quasi-PLHS, and Sobol' techniques outperformed the other sampling algorithms, as their 90% intervals are consistently narrower for any sample size for the 2-D case study. However, the performance of Sobol' method decreased for the sensitivity analysis of the 5-D HYMOD model, particularly in lower number of function evaluations. As expected, perfect-PLHS worked slightly better than quasi-PLHS, at the trade-off of not providing flexibility in sample size. The performance of SLHS was significantly better than LHS and RAND in most cases, and RAND comes the last one. These results indicate that enabling VARS with PLHS improves convergence and robustness of the sensitivity analysis, specifically in more complex problems.

5.4. Comparison in an uncertainty analysis context

Fig. 13 shows the performance of the different sampling algorithms in approximating the CDFs of the model outputs in a Monte Carlo simulation setting. Also, to assess the efficiency of different sampling algorithms in GLUE-type analyses (Beven and Binley, 1992), we set a behavioral/non-behavioral threshold of NS = 0.50 (Fig. 13 c). As can be seen, both variants of PLHS resulted in minimum average K-S and D^2_N similarity measures almost everywhere, followed by Sobol', SLHS, LHS, and RAND. This superiority demonstrates that PLHS can adequately explore the model response

surface for any sample size, and can characterize the CDF of the model response more efficiently, within less numbers of model runs.

This feature of PLHS is helpful when performing density-based (moment-free) uncertainty and sensitivity analysis, such as in regional sensitivity analysis approach of Hornberger and Spear (1981). The density-based techniques aims to characterize uncertainty and sensitivity in terms of the entire distribution of the model output (density functions). Typically, these methods measure the difference between conditional and unconditional density functions using a distance-based metrics such as Minkowski class of distance (Chun et al., 2000) or δ -density (Borgonovo, 2007) or entropy-based metrics such as Shannon entropy (Krykacz-Hausmann, 2001) or Kullback-Leibler entropy (Liu et al., 2006). Regardless of the chosen metric, quantifying density-based statistics with desirable accuracy and robustness requires a large number of function evaluations, which may come with a high computational burden (Castaings et al., 2012). The proposed PLHS can be an alternative sampling approach in performing density-based techniques by reducing the computational cost and improving robustness and accuracy of the estimations.

6. Conclusions

Modern environmental models are typically characterized by complex response surfaces, large parameter/problem spaces, and high computational demands. These attributes impede effective implementation of various sampling-based analyses, such as sensitivity and uncertainty analysis, which require running such computationally intensive models many times to adequately explore and characterize the model response surface across the parameter/problem space. In this context, the proper choice of sample size to maximize the amount of information extracted from

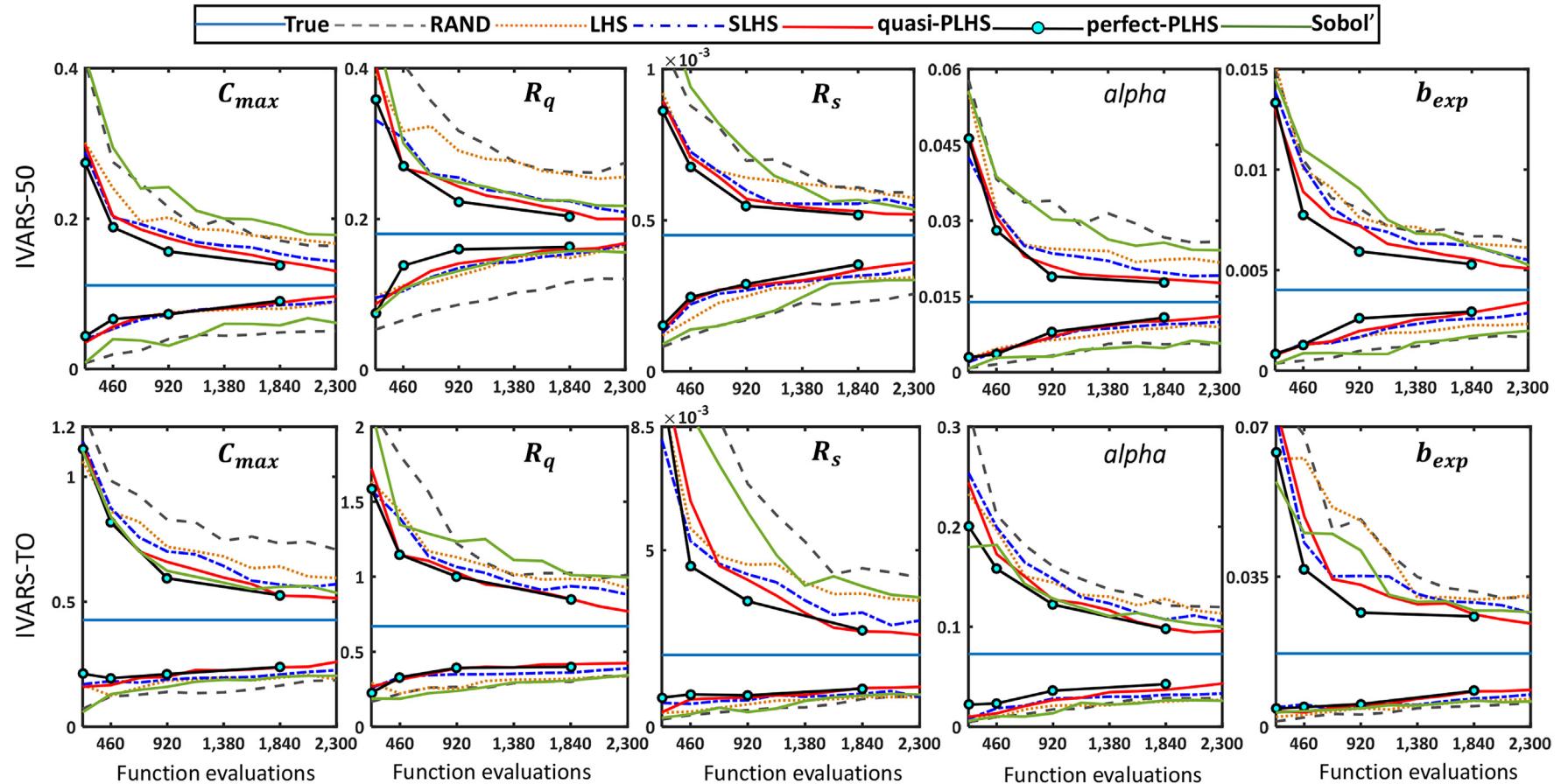


Fig. 12. Comparison of different sampling strategies in global sensitivity analysis of NS criterion to HYMOD model parameters. The top panel is for the IVARS₅₀ metric and the bottom panel is for VARS-TO (Total-Order effect) metric. The 5th and 95th percentiles of the 100 replicates are shown along with true values as the sample size grows.

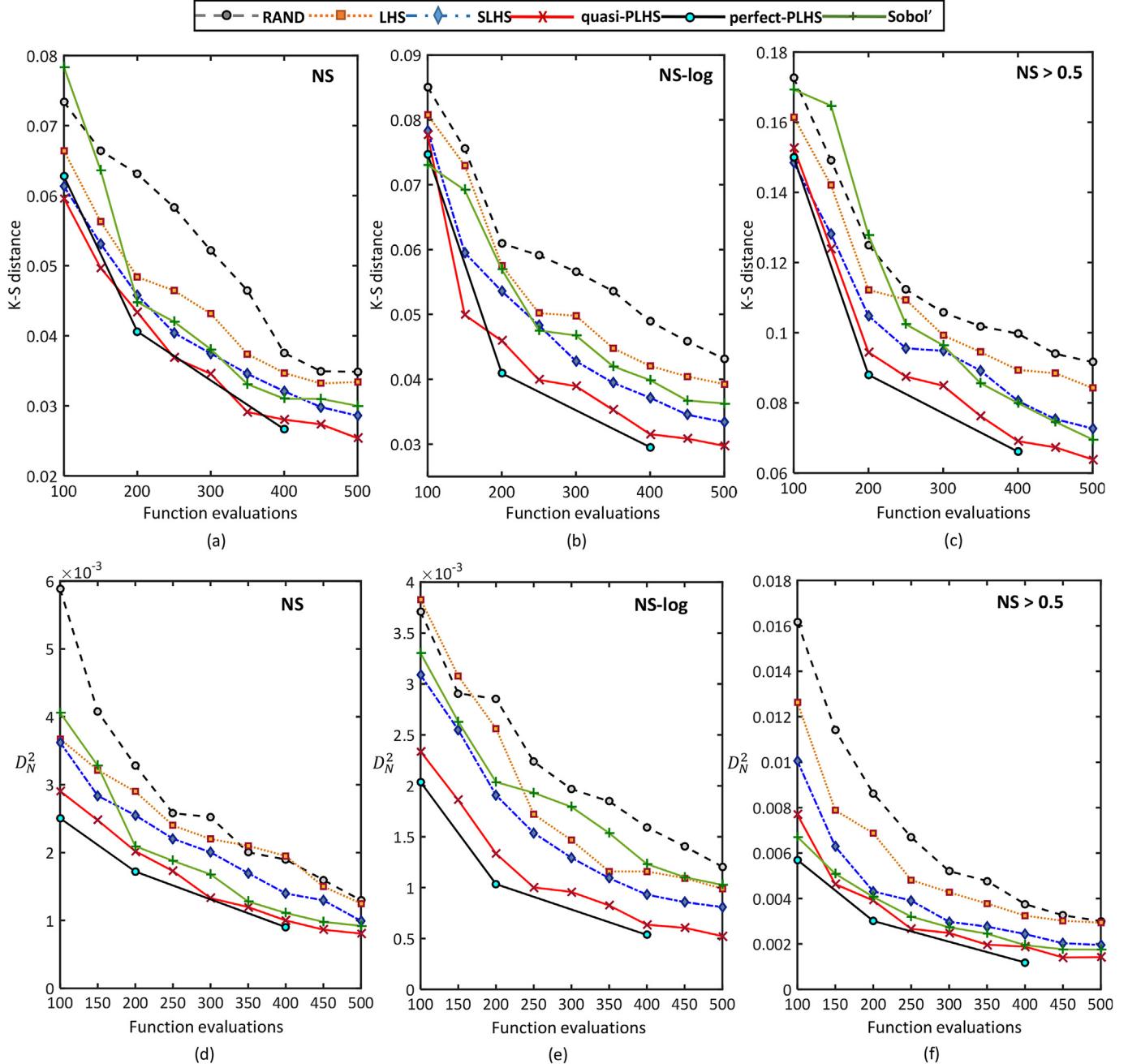


Fig. 13. Comparison of different sampling strategies in approximating the CDFs of the HYMOD model using K-S distance (top panel) and energy distance (bottom panel) metrics. (a) and (b) Show the results for NS and NS-log, respectively. (c) Shows the results for NS when of interest is to approximate the CDF of the model outputs with $NS > 0.5$ (only model outputs for behavioral parameter sets) to assess the sampling performance in a GLUE-type analysis – in each plot, the values were averaged over 100 replicates.

the model and the proper distribution of the sample points in the input space are very important. The minimum sample size is crucial in reducing the computational cost, while the proper location of sample points in the input space ensures sufficient coverage of the output space needed to characterize the complexity/nonlinearity of the response surface.

To address these, we introduced a novel strategy, called PLHS (Progressive Latin Hypercube Sampling), for sequentially sampling the input space while progressively maintaining the Latin hypercube properties. The proposed PLHS is composed of a series of smaller slices generated in a way that the union of these slices from the beginning to the current stage optimally preserves the desired

distributional properties and at the same time achieves maximum space-filling. Motivations behind developing PLHS include:

- PLHS rectifies a disadvantage of the original LHS that new sample points cannot be added sequentially to the sample set. The lack of this capability limited the utility of LHS in any sampling-based analyses where termination criteria are often checked incrementally.
- PLHS is superior to traditional sequential sampling strategies such as Halton, Hammersley, and Sobol' sequences in most cases, as unlike those, it preserves projection properties along

with other desired sample properties, in particular in high-dimensional problems.

We tested the performance of PLHS against benchmark sampling strategies by numerical experiments across four case studies using two test problems. These experiments were designed to evaluate PLHS in the input space in terms of space-filling, correlation, and projection properties and in the output space in terms of statistical measures and sensitivity metrics. The numerical experiments indicated that:

- PLHS enables improved characterization of the model input spaces and response surfaces with less computational budget (smaller sample size), compared with other sampling strategies.
- PLHS provides improved convergence rate and increased robustness to sampling variability and randomness when used in sampling-based analyses, compared with other sampling strategies.
- PLHS helps avoid over- or under-sampling by enabling users to monitor the performance of the associated sampling-based analysis as the sample size grows.

The above benefits of using PLHS will prove most useful when the simulation models are computationally intensive, which is the case in many modern Earth and environmental systems models. PLHS can minimize the computational burden of sampling-based analyses of these models by conducting only model runs that are necessary to achieve the results of desired quality.

Software availability

The PLHS software is embedded in the VARS-Tool software package, which is a MATLAB Toolbox for next generation sensitivity and uncertainty analysis. The software is freely available for non-commercial use upon request from the authors.

Acknowledgments

The first author was partially supported by a scholarship provided by the School of Environment and Sustainability at the University of Saskatchewan. The authors would like to thank the reviewers, Holger Maier, Wei Gong, and one anonymous reviewer, for their constructive comments that helped us improve this paper.

Appendix A. Construction procedure for the SLHS

Here, we describe an efficient algorithm for constructing $\text{SLHS}(n, p, T)$ with n samples, T slices/sub-samples, where each slice contains $m = n/t$ points in a p -dimensional input space (p, n, t , and m are positive integers). This construction method is based on the procedure proposed by Ba et al. (2015). The algorithm can be carried out in four steps as follows:

Step 1. Generate T small LHSs (slices) independently denoted by $\text{LHS}_k(m, p)$ for $k = 1, 2, \dots, T$, where each LHS contains m points in p factors.

Step 2. For each slice, divide the factor space $[0,1]^p$ into m disjoint intervals and index the i th sample point by $q_{ij} \in \{1, 2, \dots, m\}$ along the j th dimension regarding its interval. For all $k = 1, 2, \dots, T$ create an index matrix $\mathbf{A}_k = [q_{ij}]$ where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, p$. Let \mathbf{M} be an $n \times p$ matrix formed by the union of these index matrices such that $\mathbf{M} = \cup_{k=1}^T \mathbf{A}_k$.

Step 3. In each column of matrix \mathbf{M} , find all the elements that have the same index values ($q_{ij} = q$). Replace these elements

with random permutations of integers generated from $\{(q-1) \times T + 1, \dots, q \times T\}$.

Step 4. Randomly choose samples points from n equally probable intervals $[0, 1/n], [1/n, 2/n], \dots, [(n-1)/n, 1]$ corresponding to the elements of index matrix \mathbf{M} . Denote the resulting sample set by $\text{SLHS}(n, p, T)$.

The resulting SLHS can be further improved by going back to **Step 1** and optimizing the sample points in each slice in terms space-filling or correlation criteria.

References

- Aslan, B., Zech, G., 2005. Statistical energy as a tool for binning-free, multivariate goodness-of-fit tests, two-sample comparison and unfolding. Nucl. Instrum. Meth. A 537, 626–636. <http://dx.doi.org/10.1016/j.nima.2004.08.071>.
- Ba, S., Myers, W.R., Brennenman, W.A., 2015. Optimal sliced Latin hypercube designs. Technometrics 57 (4), 479–487. <http://dx.doi.org/10.1080/00401706.2014.957867>.
- Bates, S.J., Sienz, J., Toropov, V.V., 2004. Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm. In: Proceedings of the 5th ASMO-UK/ISSMO Conference on Engineering Design Optimization, pp. 19–22. California, US.
- Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty predication. Hydrol. Process 6 (3), 279–298. <http://dx.doi.org/10.1002/hyp.3360060305>.
- Borgonovo, E., 2007. A new uncertainty importance measure. Reliab. Eng. Syst. Saf. 92 (6), 771–784. <http://dx.doi.org/10.1016/j.ress.2006.04.015>.
- Box, G.E.P., Hunter, J.S., 1961. The 2k-p fractional factorial designs: part I. Technometrics 3 (3), 311–351. <http://dx.doi.org/10.2307/1266725>.
- Boyle, D., 2001. Multicriteria Calibration of Hydrological Models (Ph.D. thesis). Dep. of Hydrol. and Water Resour., Univ. of Ariz., Tucson.
- Broad, D.R., Dandy, G.C., Maier, H.R., 2015. A systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design. Environ. Model. Softw. 69, 382–395. <http://dx.doi.org/10.1016/j.envsoft.2014.11.015>.
- Castaings, W., Borgonovo, E., Morris, M., Tarantola, S., 2012. Sampling strategies in density-based sensitivity analysis. Environ. Model. Softw. 38, 13–26. <http://dx.doi.org/10.1016/j.envsoft.2012.04.017>.
- Chen, H., Huang, H., Lin, D.K.J., Liu, M.-Q., 2016. Uniform sliced Latin hypercube designs. Appl. Stoch. Models Bus. Ind. 32, 574–584. <http://dx.doi.org/10.1002/asmb.2192>.
- Chun, M.H., Han, S.J., Tak, N.I.L., 2000. An uncertainty importance measure using a distance metric for the change in a cumulative distribution function. Reliab. Eng. Syst. Saf. 70 (3), 313–321. [http://dx.doi.org/10.1016/S0951-8320\(00\)00068-5](http://dx.doi.org/10.1016/S0951-8320(00)00068-5).
- Cioppa, T.M., Lucas, T.W., 2007. Efficient nearly orthogonal and space-filling Latin hypercubes. Technometrics 49 (1), 45–55. <http://dx.doi.org/10.1198/004017006000000453>.
- Crombecq, K., Laermans, E., Dhaene, T., 2011. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. Eur. J. Oper. Res. 214, 683–696. <http://dx.doi.org/10.1016/j.ejor.2011.05.032>.
- Florian, A., 1992. An efficient sampling scheme: updated Latin hypercube sampling. Probabilist. Eng. Mech. 7 (2), 123–130. [http://dx.doi.org/10.1016/0266-8920\(92\)90015-A](http://dx.doi.org/10.1016/0266-8920(92)90015-A).
- Gan, Y., Duan, Q., Gong, W., Tong, C., Sun, Y., Chu, W., Ye, A., Miao, C., Di, Z., 2014. A comprehensive evaluation of various sensitivity analysis methods: a case study with a hydrological model. Environ. Model. Softw. 51, 269–285. <http://dx.doi.org/10.1016/j.envsoft.2013.09.031>.
- Ganji, A., Maier, H.R., Dandy, G.C., 2016. A modified Sobol' sensitivity analysis method for decision-making in environmental problems. Environ. Model. Softw. 75, 15–27. <http://dx.doi.org/10.1016/j.envsoft.2015.10.001>.
- Gong, W., Duan, Q.A., Li, J.D., Wang, C., Di, Z.H., Ye, A.Z., Miao, C.Y., Dai, Y.J., 2016. An intercomparison of sampling methods for uncertainty quantification of environmental dynamic models. J. Environ. Inf. 28 (1), 11–24. <http://dx.doi.org/10.3808/jei.201500310>.
- Guo, D., Westra, S., Maier, H.R., 2016. An inverse approach to perturb historical rainfall data for scenario-neutral climate impact studies. J. Hydrol. <http://dx.doi.org/10.1016/j.jhydrol.2016.03.025>.
- Halton, J.H., 1960. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. 2, 84–90.
- Hammersley, J.M., 1960. Monte Carlo methods for solving multivariable problems. Ann. N. Y. Acad. Sci. 86 (3), 844–874. <http://dx.doi.org/10.1111/j.1749-6632.1960.tb42846.x>.
- Helton, J.C., Davis, F.J., 2003. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab. Eng. Syst. Saf. 81 (1), 23–69. [http://dx.doi.org/10.1016/S0951-8320\(03\)00058-9](http://dx.doi.org/10.1016/S0951-8320(03)00058-9).
- Hernandez, A.S., 2008. Breaking Barriers to Design Dimensions in Nearly Orthogonal Latin Hypercubes (Ph.D. thesis). Naval Postgraduate School, Monterey, California.
- Hickernell, F.J., 1998. A generalized discrepancy and quadrature error bound. Math.

- Comput.* 7 (221), 299–322, 6.
- Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., Heitmann, K., Price, S., 2013. Computer model calibration using the ensemble Kalman filter. *Technometrics* 55 (4), 488–500. <http://dx.doi.org/10.1080/00401706.2013.842936>.
- Hornberger, G., Spear, R., 1981. An approach to the preliminary analysis of environmental systems. *J. Environ. Manag.* 12, 7–18.
- Husslage, B.G., Rennen, G., van Dam, E.R., den Hertog, D., 2011. Space-filling Latin hypercube designs for computer experiments. *Optim. Eng.* 12 (4), 611–630. <http://dx.doi.org/10.1007/s11081-010-9129-8>.
- Iman, R.L., Conover, W.J., 1980. Small sample sensitivity analysis techniques for computer models, with an application to risk assessment. *Commun. Stat. A-Theor.* 9 (17), 1749–1874. <http://dx.doi.org/10.1080/03610928008827996>.
- Iooss, B., Boussouf, L., Feuillard, V., Marrel, A., 2010. Numerical studies of the metamodel fitting and validation processes. *Int. J. Adv. Syst. Meas.* 3, 11–21.
- Islam, T., Pruyt, E., 2016. Scenario generation using adaptive sampling- the case of resource scarcity. *Environ. Model. Softw.* 79, 285–299. <http://dx.doi.org/10.1016/j.envsoft.2015.09.014>.
- Jin, R., Chen, W., Sudjianto, A., 2005. An efficient algorithm for constructing optimal design of computer experiments. *J. Stat. Plan. Infer.* 134 (1), 268–287. <http://dx.doi.org/10.1016/j.jspi.2004.02.014>.
- Jourdan, A., Franco, J., 2010. Optimal Latin hypercube designs for the Kullback-Leibler criterion. *AStA. Adv. Stat. Anal.* 94 (4), 341–351. <http://dx.doi.org/10.1007/s10182-010-0145-y>.
- Kalagnanam, J.R., Diwekar, U.M., 1997. An efficient sampling technique for off-line quality control. *Technometrics* 39 (3), 308–319.
- Kocis, L., Whiten, W.J., 1997. Computational investigations of low-discrepancy sequences. *ACM Trans. Math. Softw.* 23 (2), 266–294. <http://dx.doi.org/10.1145/264029.264064>.
- Kolmogorov, A., 1933. Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari* 4, 83–91.
- Krykacz-Hausmann, B., 2001. Epistemic sensitivity analysis based on the concept of entropy. In: Proceedings of SAMO2001. CIEMAT, Madrid, pp. 31–35.
- Kuczera, G., Parent, E., 1998. Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm. *J. Hydrol.* 211 (1), 69–85. [http://dx.doi.org/10.1016/S0022-1694\(98\)00198-X](http://dx.doi.org/10.1016/S0022-1694(98)00198-X).
- Linkov, I., Ramadan, A.B., 2004. Comparative Risk Assessment and Environmental Decision Making. Springer, Netherlands. <http://dx.doi.org/10.1007/1-4020-2243-3>.
- Liu, H., Chen, W., Sudjianto, A., 2006. Relative entropy based method for probabilistic sensitivity analysis in engineering design. *J. Mech. Des.* 128 (2), 326–336. <http://dx.doi.org/10.1115/1.2159025>.
- Liu, Y., Gupta, H.V., 2007. Uncertainty in hydrologic modeling: toward an integrated data assimilation framework. *Water Resour. Res.* 43 (7), W07401. <http://dx.doi.org/10.1029/2006WR005756>.
- Loiola, D., Pedergnana, M., García, S.G., 2016. Smart sampling and incremental function learning for very large high dimensional data. *Neural Netw.* 78, 75–87. <http://dx.doi.org/10.1016/j.neunet.2015.09.001>.
- Maier, H.R., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L.S., Cunha, M.C., Dandy, G.C., Gibbs, M.S., Keedwell, E., Marchi, A., Ostfeld, A., 2014. Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions. *Environ. Model. Softw.* 62, 271–299. <http://dx.doi.org/10.1016/j.envsoft.2014.09.013>.
- McKay, M.D., Conover, W.J., Beckman, R.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2), 239–245. <http://dx.doi.org/10.2307/1268522>.
- Morris, M.D., Mitchell, T.J., 1995. Exploratory designs for computational experiments. *J. Stat. Plan. Infer.* 43 (3), 381–402. [http://dx.doi.org/10.1016/0378-3758\(94\)00035-T](http://dx.doi.org/10.1016/0378-3758(94)00035-T).
- Mugunthan, P., Shoemaker, C.A., 2006. Assessing the impacts of parameter uncertainty for computationally expensive groundwater models. *Water Resour. Res.* 42, 1–15. <http://dx.doi.org/10.1029/2005WR004640>.
- Niederreiter, H., 1992. Random number generation and quasi-Monte Carlo methods. In: Proceedings of CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia.
- Ong, M.S., Kuang, Y.C., Ooi, M.P.-L., 2012. Statistical measures of two dimensional point set uniformity. *Comput. Stat. Data Anal.* 56 (6), 2159–2181. <http://dx.doi.org/10.1016/j.csda.2011.12.005>.
- Owen, A.B., 1994. Controlling correlations in Latin hypercube samples. *J. Am. Stat. Assoc.* 89 (428), 1517–1522. <http://dx.doi.org/10.2307/2291014>.
- Owen, A.B., 1992. Orthogonal arrays for computer experiments, integration and visualization. *Stat. Sin.* 2 (2), 439–452.
- Posselt, D.J., Fryxell, B., Molod, A., Williams, B., 2016. Quantitative sensitivity analysis of physical parameterizations for cases of deep convection in the NASA GEOS-5. *J. Clim.* 29, 455–479. <http://dx.doi.org/10.1175/JCLI-D-15-0250.1>.
- Pronzato, L., Müller, W.G., 2012. Design of computer experiments : space filling and beyond. *Stat. Comput.* 22 (3), 681–701. <http://dx.doi.org/10.1007/s11222-011-9242-3>.
- Qian, P.Z.G., 2012. Sliced Latin hypercube designs. *J. Amer. Stat. Assoc.* 107 (497), 393–399. <http://dx.doi.org/10.1080/01621459.2011.644132>.
- Rainville, F.-M.D., Gagné, C., Teytaud, O., Laurendeau, D., 2012. Evolutionary optimization of low-discrepancy sequences. *ACM Trans. Model. Comput. Simul.* 22, 1–25. <http://dx.doi.org/10.1145/2133390.2133393>.
- Rajabi, M.M., Ataie-Ashtiani, B., Janssen, H., 2015. Efficiency enhancement of optimized Latin hypercube sampling strategies: application to Monte Carlo uncertainty analysis and meta-modeling. *Adv. Water Resour.* 76, 127–139. <http://dx.doi.org/10.1016/j.advwatres.2014.12.008>.
- Razavi, S., Gupta, H.V., 2015. What do we mean by sensitivity analysis? The need for comprehensive characterization of “global” sensitivity in earth and environmental systems models. *Water Resour. Res.* 51 (5), 3070–3092. <http://dx.doi.org/10.1002/2014WR016527>.
- Razavi, S., Gupta, H.V., 2016a. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 1. Theory. *Water Resour. Res.* 52, 423–439. <http://dx.doi.org/10.1002/2015WR017558>.
- Razavi, S., Gupta, H.V., 2016b. A new framework for comprehensive, robust, and efficient global sensitivity analysis: 2. Application. *Water Resour. Res.* 52, 440–455. <http://dx.doi.org/10.1002/2015WR017559>.
- Razavi, S., Tolson, B.A., Burn, D.H., 2012a. Numerical assessment of metamodeling strategies in computationally intensive optimization. *Environ. Model. Softw.* 34, 67–86. <http://dx.doi.org/10.1016/j.envsoft.2011.09.010>.
- Razavi, S., Tolson, B.A., Burn, D.H., 2012b. Review of surrogate modeling in water resources. *Water Resour. Res.* 48 (7), W07401. <http://dx.doi.org/10.1029/2011WR011527>.
- Regis, R.G., Shoemaker, C.A., 2007. Parallel radial basis function methods for the global optimization of expensive functions. *Eur. J. Oper. Res.* 182, 514–535. <http://dx.doi.org/10.1016/j.ejor.2006.08.040>.
- Rezaie, K., Amalnik, M.S., Gereie, A., Ostadi, B., Shakhseniaee, M., 2007. Using extended Monte Carlo simulation method for the improvement of risk management: consideration of relationships between uncertainties. *Appl. Math. Comput.* 190 (2), 1492–1501. <http://dx.doi.org/10.1016/j.amc.2007.02.038>.
- Sallaberry, C.J., Helton, J.C., Hora, S.C., 2008. Extension of Latin hypercube samples with correlated variables. *Reliab. Eng. Syst. Saf.* 93, 1047–1059. <http://dx.doi.org/10.1016/j.ress.2007.04.005>.
- Santner, T.J., Williams, B.J., Notz, W.I., 2003. The design and analysis of computer experiments. Springer Series in Statistics. Springer, New York.
- Sarrasin, F., Pianosi, F., Wagener, T., 2016. Global sensitivity analysis of environmental models: convergence and validation. *Environ. Model. Softw.* 79, 135–152. <http://dx.doi.org/10.1016/j.envsoft.2016.02.005>.
- Schretter, C., Kobbel, L., Dehaye, P.O., 2012. Golden ratio sequences for low-discrepancy sampling. *J. Graph. Tools* 16 (2), 95–104.
- Simpson, T.W., Lin, D.K.J., Chen, W., 2001. Sampling strategies for computer experiments design and analysis. *Int. J. Reliab. Appl.* 2 (3), 209–240.
- Sobol', I., 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* 7 (4), 86–112. [http://dx.doi.org/10.1016/0041-5553\(67\)90144-9](http://dx.doi.org/10.1016/0041-5553(67)90144-9).
- Székely, G.J., Rizzo, M.L., 2005. A new test for multivariate normality. *J. Multivar. Anal.* 93 (1), 58–80. <http://dx.doi.org/10.1016/j.jmva.2003.12.002>.
- Székely, G.J., Rizzo, M.L., 2013. Energy statistics: a class of statistics based on distances. *J. Stat. Plann. Inference* 143 (8), 1249–1272. <http://dx.doi.org/10.1016/j.jspi.2013.03.018>.
- Tang, B., 1993. Orthogonal array-based Latin hypercubes. *J. Am. Stat. Assoc.* 88 (424), 1392–1397. <http://dx.doi.org/10.2307/2291282>.
- van Dam, E.R., Husslage, B., den Hertog, D., Melissen, H., 2007. Maximin Latin hypercube designs in two dimensions. *Oper. Res.* 55 (1), 158–169. <http://dx.doi.org/10.1287/opre.1060.0317>.
- Viana, F.A., 2013. Things you wanted to know about the Latin hypercube design and were afraid to ask. In: Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization. Florida, Orlando.
- Viana, F.A.C., Venter, G., Balabanov, V., 2010. An algorithm for fast generation of optimal Latin hypercube designs. *Int. J. Numer. Meth. Eng.* 82 (2), 135–156. <http://dx.doi.org/10.1002/nme.2750>.
- Vorechovský, M., 2009. Hierarchical subset Latin hypercube sampling for correlated random vectors (Madeira, Portugal). In: Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering.
- Vrugt, J.A., Gupta, H.V., Bouten, E., Sorooshian, S., 2003. A shuffled complex evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.* 39 (8) <http://dx.doi.org/10.1029/2002WR001642>.
- Vrugt, J.A., Nuallán, B.O., Robinson, B.A., Bouten, W., Dekker, S.C., Sloot, P.M., 2006. Application of parallel computing to stochastic parameter estimation in environmental models. *Comput. Geosci.* 32 (8), 1139–1155. <http://dx.doi.org/10.1016/j.cageo.2005.10.015>.
- Wagener, T., Boyle, D.P., Lees, M.J., Wheater, H.S., Gupta, H.V., Sorooshian, S., 2001. A framework for development and application of hydrological models. *Hydro. Earth Syst. Sci.* 5, 13–26. <http://dx.doi.org/10.5194/hess-5-13-2001>.
- Wallis, W.D., George, J., 2011. Introduction to Combinatorics. CRC Press, London.
- Warnock, T.T., 1972. Computational investigations of low-discrepancy pointsets. In: Zaremba, S.K. (Ed.), Applications of Number Theory to Numerical Analysis. Academic Press, New York, pp. 319–343.
- Williamson, D., 2015. Exploratory ensemble designs for environmental models using k-extended Latin Hypercubes. *Environmetrics* 26, 268–283. <http://dx.doi.org/10.1002/env.2335>.
- Xiong, F., Xiong, Y., Chen, W., Yang, S., 2009. Optimizing Latin hypercube design for sequential sampling of computer experiments. *Eng. Optim.* 41 (8), 793–810. <http://dx.doi.org/10.1080/03052150902852999>.

- Ye, K.Q., 1998. Orthogonal column Latin hypercubes and their application in computer experiments. *J. Am. Stat. Assoc.* 93 (444), 1430–1439. <http://dx.doi.org/10.2307/2670057>.
- Ye, K.Q., Li, W., Sudjianto, A., 2000. Algorithmic construction of optimal symmetric Latin hypercube designs. *J. Stat. Plan. Infer.* 90 (1), 145–159. [http://dx.doi.org/10.1016/S0378-3758\(00\)00105-1](http://dx.doi.org/10.1016/S0378-3758(00)00105-1).
- Zhan, Y., Zhang, M., 2013. Application of a combined sensitivity analysis approach on a pesticide environmental risk indicator. *Environ. Model. Softw.* 49, 129–140. <http://dx.doi.org/10.1016/j.envsoft.2013.08.005>.