ORIGINAL PAPER

Latin hypercube sampling with inequality constraints

**Matthieu Petelet · Bertrand Iooss ·
Olivier Asserin · Alexandre Loredo**

Abstract In some studies requiring predictive and CPU-time consuming numerical models, the sampling design of the model input variables has to be chosen with caution. For this purpose, Latin hypercube sampling has a long history and has shown its robustness capabilities. In this paper we propose and discuss a new algorithm to build a Latin hypercube sample (LHS) taking into account inequality constraints between the sampled variables. This technique, called constrained Latin hypercube sampling (cLHS), consists in doing permutations on an initial LHS to honor the desired monotonic constraints. The relevance of this approach is shown on a real example concerning the numerical welding simulation, where the inequality constraints are caused by the physical decreasing of some material properties in function of the temperature.

1 Introduction

With the advent of computing technology and numerical methods, investigation of computer code experiments remains an important challenge. Complex computer

M. Petelet · O. Asserin
CEA, DEN, DM2S, SEMT, LTA, 91191, Gif sur Yvette, France

B. Iooss (✉)
EDF, R&D, 6 Quai Watier, 78401 Chatou, France
e-mail: bertrand.iooss@edf.fr

A. Loredo
LRMA, EA 1859, Université de Bourgogne, Nevers, France

models calculate several output values which can depend on a large number of input parameters and physical variables. These computer models are used to make simulations as well as predictions, uncertainty and sensitivity analyses or to solve optimization problems (Fang et al. 2006; Kleijnen 2008; De Rocquigny et al. 2008; Levy and Steinberg 2010).

However, complex computer codes are often too time expensive to be directly used to perform such studies. For uncertainty propagation and sensitivity analyses, it has been shown that the sampling design is one of the key issues (Saltelli et al. 2000; Fang et al. 2006). Moreover, to avoid the problem of huge calculation time, it is often useful to replace the computer code by a mathematical approximation, called a surrogate model or a metamodel (Simpson et al. 2001b; Fang et al. 2006; Volkova et al. 2008). The optimal exploration of the variation domain of the input variables is therefore especially important in order to avoid non-informative simulation points (Sobol 1976; Simpson et al. 2001a; Bursztyn and Steinberg 2006; Iooss et al. 2010; Levy and Steinberg 2010).

Thirty years ago, McKay et al. (1979) have introduced the concept of Latin hypercube sampling (LHS) for numerical experiments. Compared to simple random sampling (SRS), which ensured independence between samples, LHS ensures the full coverage of the range of the input variables. More precisely, LHS allows to accurately reproduce the one-dimensional projections of the input sampling design. In terms of uncertainty and sensitivity analyses, it has been theoretically and experimentally proved that LHS is more precise and robust than simple random sample (Stein 1987; Owen 1992; Saltelli et al. 2000; Helton and Davis 2003). Moreover, in the last 20 years, several improvements have been proposed in order to optimize the space filling properties of LHS designs (Park 1994; Fang et al. 2006; Pistone and Vicario 2010; Jourdan and Franco 2010).

Our starting point is that the initial LHS algorithm supposes independence between input variables, while in some situations this assumption is irrelevant. First, let us recall the two forms of dependencies between variables: the statistical and the physical ones.

– The correlation coefficient is the simplest measure of the statistical dependence between two variables. For the SRS, the so-called joint normal transform method consists in inducing a correlation structure on the transformed marginals (Kurowicka and Cooke 2006). The rank correlation coefficient, based on the rank transformation (which turns each variable value to its rank in the sample), is known as a more robust measure. Iman and Conover (1982) have introduced an algorithm to consider rank correlations between variables in LHS. Some limitations of these two dependence measures have led to the introduction of other statistical dependence modelling (copulae, vines, etc., see Kurowicka and Cooke 2006).
– Physical dependencies between variables can arise when a variable has a formal relation in function of other variables. Such input constraints have been studied by Borgonovo (2008) who has proposed a novel way to solve the sensitivity analysis problem in presence of equality constraints. Another currently encountered physical dependence, which is the subject of this paper, concerns the existence of inequality relations between the variables. It is the case when one variable is physically constrained to be larger (respectively, smaller) than another. For example, a

geometric parameter (radius, height, etc.) of two physical objects can be subject to a rigorous increasing order if one object is included inside the other.

When building the sampling design, the inequality constraints have to be honored in order to avoid some physical incoherence in the input sets that will be run with the computer model. A first solution could be to sequentially simulate the input variables, allowing to bound one variable by another in order to enforce an inequality constraint. However, as we will see, this procedure affects the one-dimensional projections of the sampling design. We need a procedure which separates the effect of dependence (the inequality constraints) from the effects of marginal distributions (i.e. the probability laws defined for each variable). To attain this objective, we propose an algorithm which builds a LHS satisfying the inequality constraints.

This paper is devoted to the detailed presentation of this algorithm, called the constrained LHS (cLHS). In the next section, we introduce this algorithm by giving some examples. We compare it with a SRS-based algorithm and illustrate the algorithmic performances. In the third section, we explain in detail the cLHS algorithm. As the inequality constraints can be too stringent to find a cLHS, we derive a necessary and sufficient condition proving its existence from an initial LHS. Then, our methodology is applied on a real problem involving welding simulation models. A conclusion gives finally some prospects to improve the cLHS algorithm.

2 The sampling techniques

The goal of the sampling step is to generate a matrix $\mathbf{X}^n = (x_j^{(i)})_{i=1..n, j=1..p}$, where n is the number of experiments and p is the number of variables. The most common sampling method is indisputably the pure Monte Carlo (i.e. SRS), mainly because of its simplicity (Gentle 2003). It consists of randomly sampling n independent input variables. However, it is known to have poor space filling properties: SRS leaves large unsampled regions and can propose too close points. An example of a SRS is presented on Fig. 1(a).

2.1 Latin hypercube sampling

McKay et al. (1979) suggested an alternative method of generating $\mathbf{X}^n$ that they called Latin hypercube sampling (LHS), which is an extension of stratified sampling. LHS ensures that each of the input variables has all of its range represented. Let the range of each variable X_j, $j = 1, \ldots, p$, be simultaneously partitioned into n equally probable intervals. We note X_j^n the n-sample of the variable X_j. A LHS of size n is obtained from a random selection of n values—one per stratum—for each X_j. Thus we obtain p n-tuples that form the p columns of the $n \times p$ matrix of experiments $\mathbf{X}^n$ generated by LHS: the ith line of this matrix contains the p input variables and will correspond to the ith code execution. Once a point is selected in an interval, no other point could be selected in this interval (see Fig. 1 (b)). Let us remark that the partition into equally probable intervals allows one to take into account non-uniform densities of probability like a normal distribution for example. Figure 1 shows 10 samples of two random variables obtained with SRS and LHS schemes. We can see

(a) Simple Random Sampling

(b) Latin Hypercube Sampling
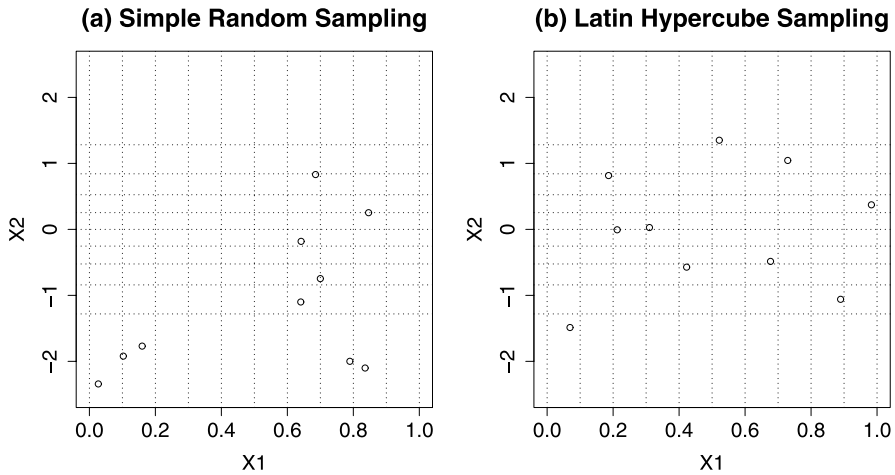
Fig. 1 Examples of two ways to generate a sample of size $n = 10$ from two variables $\mathbf{X} = [X_1, X_2]$ where X_1 has a uniform distribution $\mathcal{U}[0, 1]$ and X_2 has a normal distribution $\mathcal{N}(0, 1)$

that the result of LHS is more spread out and does not display the clustering effects found in SRS.

Mathematically, if $X_1, \ldots, X_p$ are mutually independent random variables with invertible continuous distribution functions F_j, $j = 1, \ldots, p$, respectively, then the LHS ith sample for the jth variable can be created as

$$x_j^{(i)} = F_j^{-1} \left(\frac{\pi_j^{(i)} - \xi_j^{(i)}}{n} \right), \tag{1}$$

where the π_j are independent uniform random permutations of the integers $\{1, 2, \ldots, n\}$, and the $\xi_j^{(i)}$ are independent $\mathcal{U}[0, 1]$ random numbers independent of the π_j.
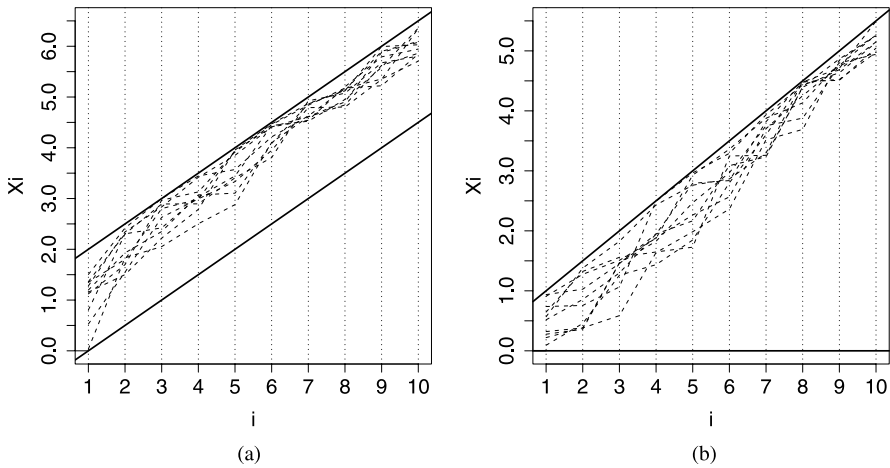
2.2 Constrained simple random sampling

To take into account inequality constraints between variables, the simplest approach is based on SRS and consists on bounding one variable by another in order to enforce the inequality constraints. This approach is called the constrained Simple Random Sampling (cSRS).

In Fig. 2, we see the effect of an inequality constraint between two variables in terms of bivariate plots. Of course, the introduction of the truncation during the simulation creates statistical dependences between X_1^n and X_2^n. This correlation depends on the distribution functions of X_1 (called F_1) and X_2 (called F_2). In our example, the correlation coefficient $\rho(X_1, X_2)$ is worth 31%. In Fig. 2, the one-dimensional marginal projections of the samples are also shown. For the cSRS, the one-dimensional marginal of X_2^n does not correspond to F_2 anymore but to a transformed distribution F_2' (which depends on F_1).
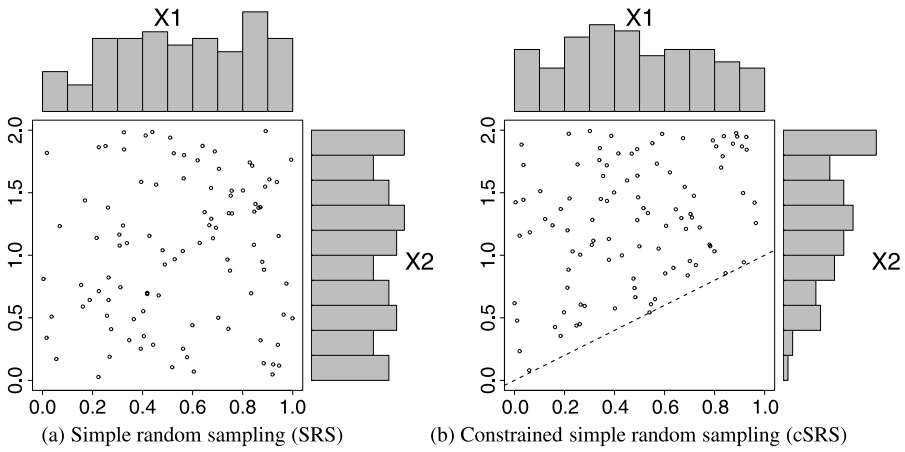
Fig. 2 Comparison between Monte Carlo samples of size $n = 100$ from two variables $\mathbf{X} = [X_1, X_2]$ where $X_1 \sim \mathcal{U}[0, 1]$ and $X_2 \sim \mathcal{U}[0, 2]$. The inequality constraint for cSRS is $X_1 < X_2$

Fig. 3 Constrained simple random samples of size $n = 10$ from $p = 10$ variables X_i, $i = 1, \ldots, 10$, with $X_i < X_{i+1}$ for $i = 1, \ldots, 9$. The upper and the lower curves represent the bounds of the variation ranges for these 10 variables. (**a**) $X_i \sim \mathcal{U}[0 + \frac{i-1}{2}, 2 + \frac{i-1}{2}]$; (**b**) $X_i \sim \mathcal{U}[0, 1 + \frac{i-1}{2}]$

This problem becomes more dramatic when the input dimension increases and when several sequential inequality constraints have to be satisfied, as for example if $X_i < X_{i+1}$ for $i = 1, \ldots, p - 1$ and p is large. Figure 3(a) shows an example of cSRS of $p = 10$ variables with such increasing constraints. In this graph, each observation is represented as a line. Because of the sequential algorithm starting at X_1, all the curves are concentrated near the upper bound curve of the variables. Figure 3(b) clearly reveals that the sampling of the first variable X_1 is adequate with its uniform distribution, and that the samples of the following variables (X_2 to X_{10}) progressively take place in the upper region of their variation range.

In summary, in some practical situations, users would like to simulate samples which follow all one-dimensional marginals and which take into account some inequality constraints. The following section proposes such an algorithm.

2.3 Constrained Latin hypercube sampling

In order to follow all one-dimensional marginals, our sampling procedure uses LHS. Our method, first proposed in Petelet (2007), consists in doing permutations on an initial LHS to enforce the desired monotonic constraint. It is based on the fact that permuting two values of a variable in a LHS does not break the LHS structure of the sample (Iman and Conover 1982). An appropriate algorithm scans the starting LHS to find the couples of values that violate the monotonic constraint. Then the algorithm finds and executes the combinations of permutations which have to be done to satisfy the inequality constraint between X_i and X_{i+1} for the n experiments. Details of the algorithm are given in Sect. 3.

Figure 4 shows the work done on a couple of parameters on which an increasing constraint is enforced. The distribution of X_1 and X_2 are kept uniform (the slight variation of the height of one class in the histogram of X2 is not significant). In the bivariate plots, we see that the cLHS constraints (the increasing inequality and the honoring of all one-dimensional marginals) tend to gather the sample points along the inequality frontier line $X_1 = X_2$. It appears that the severity of the inequality constraint effects strongly depend on the one-dimensional marginal distributions F_1 and F_2 of the variables. The limit case is illustrated on Fig. 4(b) where X_1 and X_2 have the same one-dimensional marginal distributions F_1 and F_2 (then the same upper and lower bounds). In such a case where half the area of the bivariate plot is forbidden all the points are located on this frontier line. This effect results of a too severe constraint and reveals the need of a constraint intensity measurement. Petelet (2007) has defined this constraint intensity measurement as the ratio between the triangular forbidden area (S_T) in the bivariate plot and the rectangular area (S_R) of the domain defined by the upper and lower bounds of the variables:

$$\gamma = \frac{S_T}{S_R}. \tag{2}$$

The γ measurement can be used if each of the input variable has some upper and lower bounds, i.e. if the support of their distribution function is defined on a bounded domain. In the general case of an inequality constraint between X_i and X_j, respectively, defined on $[b_i, h_i]$ and $[b_j, h_j]$, we obtain

$$\gamma(X_i, X_j) = \frac{S_T(X_i, X_j)}{S_R(X_i, X_j)} = \begin{cases} \frac{(h_i - b_j)^2}{2(h_i - b_i)(h_j - b_j)} & \text{for the constraint } X_i < X_j, \\ \frac{(h_j - b_i)^2}{2(h_i - b_i)(h_j - b_j)} & \text{for the constraint } X_i > X_j. \end{cases} \tag{3}$$

The intensity constraint measurements for the Fig. 4 cases are worth $\gamma(X_1, X_2) = 25\%$ for (a) and $\gamma(X_1, X_2) = 45.5\%$ for (b). With some heuristic arguments, Petelet
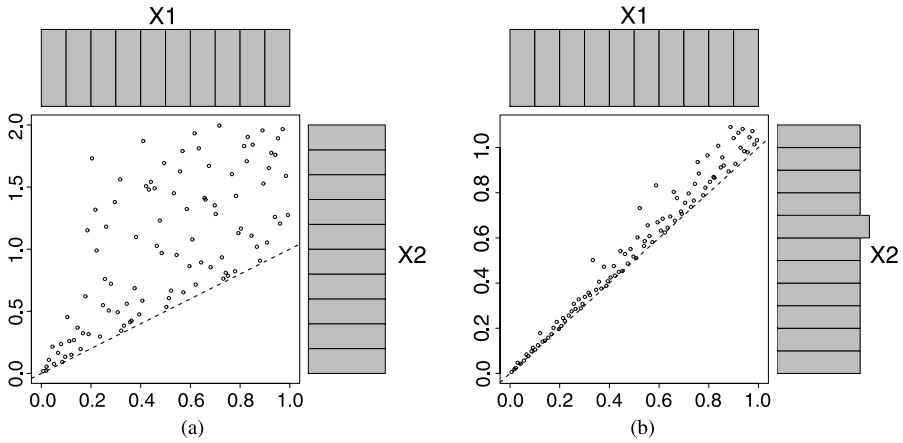
Fig. 4 Comparisons between constrained Latin hypercube samples of size $n = 100$ from two variables $\mathbf{X} = [X_1, X_2]$ with $X_1 \sim \mathcal{U}[0, 1]$ and the inequality constraint $X_1 < X_2$: (**a**) $X_2 \sim \mathcal{U}[0, 2]$; (**b**) $X_2 \sim \mathcal{U}[0, 1.1]$

(2007) has found a nearly linear link between $\gamma(X_i, X_j)$ and the correlation coefficient $\rho(X_i, X_j)$ for $\gamma(X_i, X_j) \in [0, 0.3]$:

$$\rho(X_i, X_j) \simeq 2.778\gamma(X_i, X_j). \tag{4}$$

As γ is positive, the correlation coefficient will be always positive. For example, this relation shows that if the inequality constraint is kept smaller than 15%, the correlation between the variables will be smaller than 40%.

For the same cases than in Figs. 3, Fig. 5 shows the cLHS of $p = 10$ variables with sequential increasing constraints. As before, in this graph, each observation is represented as a line. At present, the curves correctly fill the variation ranges of all the variables. The constraint intensity measurements for Fig. 5(a) are worth $\gamma(X_i, X_{i+1}) = 18.75\%$ for $i = 1, \ldots, 9$. This value is rather suitable: correlations between variables are smaller than 52% (value obtained thanks to (4)). For Fig. 5(a), the constraint intensity measurements increase from $\gamma(X_1, X_2) = 33.33\%$ to $\gamma(X_9, X_{10}) = 45.45\%$.

When the upper and lower bounds of the variables are similar, the cLHS tends to give homothetic translated trajectories, as shown by Fig. 5(b). In the limit case, if all the X_i's have the same one-dimensional marginal distributions (then the same upper and lower bounds), the obtained curves are parallel and regularly spaced between the lower bound curve and the upper bound curve. This is one of the drawback of our algorithm, caused by the imposed LHS structure. Moreover, the feasibility of the cLHS depends on the bound values of the constrained variables. For example, for the constraint $X_i < X_j$, the algorithm does not work if $h_i > h_j$ or if $b_i > b_j$. From (3), this implies that the constraint intensity measurement $\gamma(X_i, X_j)$ is upper bounded by 0.5.
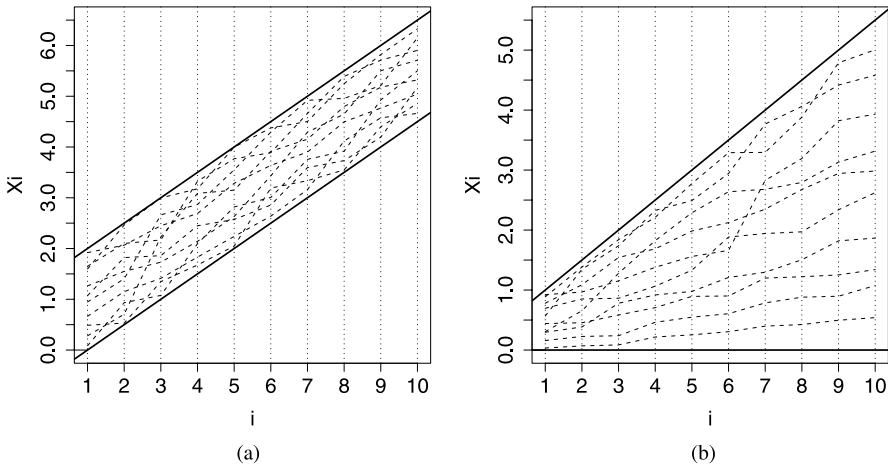
Fig. 5 Constrained Latin hypercube samples of size $n = 10$ from $p = 10$ variables X_i, $i = 1, \ldots, 10$, with $X_i < X_{i+1}$ for $i = 1, \ldots, 9$. The upper and the lower curves represent the bounds of the variation range for these 10 variables. (**a**) $X_i \sim \mathcal{U}[0 + \frac{i-1}{2}, 2 + \frac{i-1}{2}]$; (**b**) $X_i \sim \mathcal{U}[0, 1 + \frac{i-1}{2}]$

3 The constrained Latin hypercube sampling algorithm

In the following, we explain the cLHS algorithm in the case of a strict increasing constraint between two variables X_1 and X_2 (with distribution functions F_1 and F_2, respectively). The developments for the strict decreasing constraint case are exactly the same, by inverting the inequalities sense. Moreover, the extension of our algorithm to non-strict inequality constraints is straightforward.

For the increasing constraint case, we assume the following hypotheses.

– $\mathbf{X} = (X_1, X_2)$ is defined on a bounded domain $\mathcal{X} \in \mathbb{R}^2$. The support of F_j for $j = \{1, 2\}$ is $[b_j, h_j]$.
– The bounds are subject to the following inequalities:

$$b_1 \leq b_2 \quad \text{and} \quad h_1 \leq h_2. \tag{5}$$

These inequalities seem natural: if we impose some increasing constraint between X_1 and X_2, we hope that the same increasing constraints exist for their minimal and maximal bounds.

Let us define the matrix $\mathbf{C}^n = \mathbf{C}(X_1^n, X_2^n)$ of size $n \times n$:

$$\mathbf{C}^n = (c_{ij})_{i=1..n, j=1..n} = \begin{pmatrix} \mathbb{1}_{x_2^{(1)} > x_1^{(1)}} & \cdots & \mathbb{1}_{x_2^{(1)} > x_1^{(n)}} \\ \vdots & \ddots & \vdots \\ \mathbb{1}_{x_2^{(n)} > x_1^{(1)}} & \cdots & \mathbb{1}_{x_2^{(n)} > x_1^{(n)}} \end{pmatrix}, \tag{6}$$

where $\mathbb{1}_{x>y} = 1$ if $x > y$ and $\mathbb{1}_{x>y} = 0$ otherwise. $\mathbf{C}^n$ is called the compatibility matrix between X_1^n and X_2^n. This matrix allows us to identify which combinations of

elements of X_1^n and X_2^n are incompatible, i.e. those with a decreasing relation. Therefore, the inequality constraint between the two samples is honored if the diagonal of $\mathbf{C}^n$ contains only 1's.

For our cLHS algorithm, if we choose to leave X_1^n unchanged and give the possibility to permute some elements of X_2^n, we define our final objective as getting a sample $X_2'^n$ such that

$$\sum_{i=1}^{n} c'_{ii} = n, \tag{7}$$

with $\mathbf{C}' = (c'_{ij})_{i=1..n, j=1..n}$ the compatibility matrix between X_1^n and $X_2'^n$.

At present, it would be convenient to know if this objective can be achieved for a specific sample $\mathbf{X}^n$. Let us define the sample vector

$$\mathbf{S}^n = (S_i)_{i=1..n} = \left(\sum_{j=1}^{n} c_{ij} \right)_{i=1..n}. \tag{8}$$

S_i gives the number of elements of X_1^n which satisfy the constraint with $x_2^{(i)}$. We also define $(\tilde{S}_i)_{i=1..n}$ the ordered sample of $(S_i)_{i=1..n}$. To insure that we can obtain a sample $X_2'^n$ (by permutations of the X_2^n elements) satisfying the increasing constraint, the following assertions have to be true.

– The smallest element of X_2^n has one or more than one smaller elements in X_1^n, which is equivalent to saying that $\min(X_2^n) \geq \min(X_1^n)$, then again, to saying that $\tilde{S}_1 \geq 1$.
– ...
– The ith-smallest element of X_2^n has i or more than i smaller elements in X_1^n, which is equivalent to saying that $\tilde{S}_i \geq i$.
– ...
– The nth-smallest elements of X_2^n have n or more than n smaller elements in X_1^n, which is equivalent to saying that $\tilde{S}_n \geq n$.

From these assertions, we obtain the following result:

Proposition *If* (X_1^n, X_2^n) *is a LHS, the inequality*

$$\min\left[(\tilde{S}_1 \cdots \tilde{S}_n) - (1 \cdots n) \right] \geq 0 \tag{9}$$

is a necessary and sufficient condition to guarantee the existence of a sample $X_2'^n$ such that $(X_1^n, X_2'^n)$ satisfies the increasing constraint $X_1 < X_2$, where $X_2'^n$ is obtained from permutations of the elements of X_2^n.

Therefore, the first step of our methodology will be to test this criterion. If (9) is not verified for a chosen $\mathbf{X}^n = (X_1^n, X_2^n)$, a new sample for X_2^n (keeping the LHS property for $\mathbf{X}^n$) is created and the scan starts again. Our hypotheses on the bounds of X_1 and X_2 (see (5)) guarantee that a LHS satisfying the increasing constraint exists.

We start from an initial LHS $\mathbf{X}^n = (X_1^n, X_2^n)$, with a compatibility matrix $\mathbf{C}$, satisfying the existence criterion (9). We want to obtain the LHS $\mathbf{X}'^n = (X_1^n, X_2'^n)$ (with a compatibility matrix $\mathbf{C}'$) satisfying the increasing constraint between X_1 and X_2'. Our objective is therefore to obtain the result of (7).

Let us note $\tilde{X}_1^n$ the reverse ordered sample of X_1^n. This vector $\tilde{X}_1^n$ contains the elements $\tilde{x}_1^{(1)} \geq \tilde{x}_1^{(2)} \geq \cdots \geq \tilde{x}_1^{(n)}$. We put in the sample vector A^n a sequence of indices: the indices in X_1^n of the $\tilde{X}_1^n$ elements. Mathematically, it follows that

$$x_1^{(A_i)} = \tilde{x}_1^{(i)}. \tag{10}$$

Our permutation algorithm is based on the treatment of the X_1^n elements in a sequential manner (because the constraint is more difficult to be satisfied by the first values of X_1^n). We describe the algorithm with the following four steps.

Algorithm cLHS

1. Initialisation: $\mathbf{C}'^n = C^n$ and $\mathbf{B}' = (1, \ldots, n)$.

2. For $i = 1, \ldots, n$:

   – we put in the vector $\mathbf{B}$ the indices in X_2^n of the elements compatible with $x_1^{(A_i)}$:

   $$k = 1$$
   $$\text{For } j = 1, \ldots, n:$$
   $$\text{if } c'_{jA_i} = 1 \text{ then } B_k = k \text{ and } k = k + 1$$

   – we randomly choose an element in $\mathbf{B}$ and put it in B_i'.
   – the index B_i' corresponds to the one that will be permuted in X_2^n. We turn to zero the line B_i' in the compatibility matrix $\mathbf{C}'^n$ (in order to block up the index B_i'):

   $$\text{For } j = 1 \ldots n: c'_{B_i'j} = 0$$

3. The vector $\mathbf{B}'$ contains the indices that will be used to make the permutations in X_2^n. We obtain the new sample of the variable X_2:

$$(X_2')_{i=1..n} = (X_2)_{i=B_1'..B_n'}. \tag{11}$$

4. Finally, the permutation matrix $\mathbf{C}'^n$ is calculated with X_1^n and $X_2'^n$ by (6) in order to test the equality of (7).

End of algorithm

The extension of the CLHS algorithm to the multivariate case $\mathbf{X} = (X_1, \ldots, X_p)$ with $X_j < X_{j+1}$ for $j = 1, \ldots, p - 1$ is straightforward and is done in a sequential manner. We first simulate a LHS $\mathbf{X}^n = (X_1^n, \ldots, X_p^n)$. Leaving X_1^n unchanged, we sequentially build with the cLHS algorithm $X_j'^n$ from $X_{j-1}'^n$ and X_j^n for $j = 2, \ldots, p$. At each step j, before applying the algorithm, the criterion (9) is tested. If this criterion is not verified, a new LHS for X_j^n is created, and so on until the criterion is verified.

4 The cLHS algorithm in an example

We propose a simple example with a sample of size $n = 6$ and two variables X_1 and X_2 subject to a decreasing constraint $X_1 > X_2$. Points are uniformly sampled on the domain $\mathcal{X} = [20, 30] \times [16, 26]$ of (X_1, X_2). We simulate an initial LHS (Fig. 6(a)) and obtain the following design matrices:

$$
X_1^n = \begin{pmatrix} 23.98 \\ 26.91 \\ 26.52 \\ 21.99 \\ 29.23 \\ 21.10 \end{pmatrix}, \qquad
X_2^n = \begin{pmatrix} 22.18 \\ 20.45 \\ 23.77 \\ 18.31 \\ 16.45 \\ 25.49 \end{pmatrix}, \qquad \text{then} \quad
C^n = \begin{pmatrix} \mathbf{1} & 1 & 1 & 0 & 1 & 0 \\ 1 & \mathbf{1} & 1 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{1} & 0 & 1 & 0 \\ 1 & 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & \mathbf{1} & 1 \\ 0 & 1 & 1 & 0 & 1 & \mathbf{0} \end{pmatrix}.
$$

As the diagonal of this matrix has a null term, (7) is not fulfilled. Then, the cLHS algorithm has to be applied in order to obtain a LHS satisfying the constraint.

First, we test the existence criterion. We obtain $\mathbf{S}^n = (4\ 6\ 4\ 6\ 6\ 3)$ and $\tilde{\mathbf{S}}^n = (3\ 4\ 4\ 6\ 6\ 6)$. The existence criterion (see (9)) is then fulfilled.

Second, we apply the algorithm and obtain the following result:

$$
X_2'^n = \begin{pmatrix} 20.45 \\ 25.49 \\ 22.18 \\ 18.31 \\ 23.77 \\ 16.45 \end{pmatrix}. \tag{12}
$$

X_1^n has not been modified while elements of X_2^n have been permuted to obtain $X_2'^n$, which is a sample satisfying the decreasing constraint. Other $X_2'^n$ samples could be found, the choice made during the cLHS algorithm being random. Figure 6(b) shows our final sampling result.

5 An application case: welding thermomechanical models

The robust increase in computer power has tremendously contributed to a growing fad for welding simulation. The industrial requirements are more and more numerous: supports to develop new processes, control of mechanical welding effects (in particular, residual stresses and distortions), argument in a nuclear safety analysis reports, etc. Thus, through the use of high-performance computers and advanced models, numerical simulation is expected to become an important tool for innovation in welding engineering.

However, running a welding simulation model requires a large number of inputs—about 500—including for example meshing inputs, boundary and initial conditions as well as material properties and process parameters, and generates several outputs, including spatial distributions of displacements and residual stresses in the weldment. In particular, among inputs, the determination of material properties is

(a) Initial LHS   (b) Final LHS
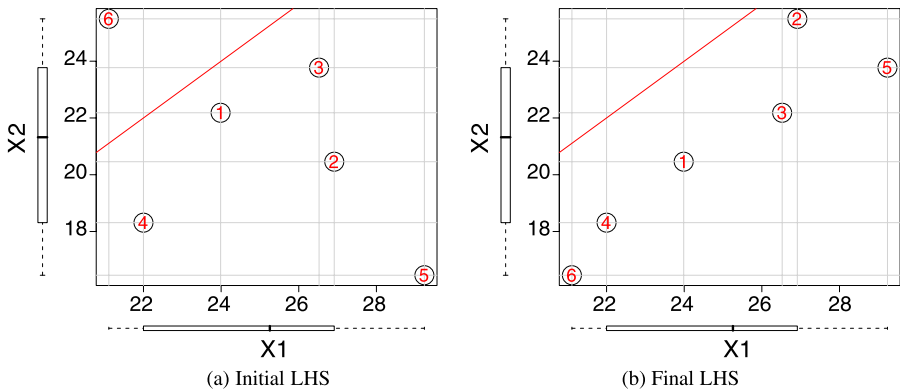
Fig. 6 Illustration of the cLHS algorithm ($n = 6$) with a decreasing constraint between two uniformly-distributed random variables X_1 and X_2 ($\gamma(X_1, X_2) = 18\%$). The *diagonal line* corresponds to the frontier line $X_1 = X_2$

one of the key problems of welding simulation. The features of material properties are that they are dependent on temperature and that their full characterization is very expensive, often difficult or even sometimes impossible. In this context, the global sensitivity analyses of the numerical welding simulation model allows to determine which material properties are the most sensitive in a numerical welding simulation and in which range of temperature (Petelet 2007; Petelet et al. 2006; Asserin et al. 2009).

Let us show the application of our methodology on the range of steel material. Five input variables are the mechanical properties used by the model: Young's modulus, thermal computation coefficient, Poisson's ratio, yield strength and hardening modulus. Because of their dependence on temperature, it has been required to sample each material property at a discrete set of temperatures: seven levels are chosen from 20°C to 1100°C. We obtain $35 = 5 \times 7$ input variables, each following an uniform distribution defined by its minimal and maximal bounds (taken from the literature). Moreover, some material properties used in this model are monotonically decreasing as function of temperature. Therefore, the constrained Latin hypercube sampling strategy, described in this paper, can be used to generate the input design. This strategy allows our sampled variables to honor their uniform repartition defined by their minimal and maximal bounds, that is to say, sampling in the physical bounds.

To illustrate this application, we present the sampling of the Young's modulus. Figure 7 shows the result on this 7-dimensional variable, which follows a decreasing constraint. These curves present three randomly selected materials among the 800 created and the bounds of the domain. For the sensitivity analysis process, the Young's modulus is represented by only seven parameters (Asserin et al. 2009). However, one should keep in mind that for the mechanical computation, the curve represents truly the considered dependence of this modulus because the algorithm uses intermediate values according to a piecewise linear interpolation.

It would be interesting to present the sensitivity analysis results obtained by employing the classical LHS (unconstrained) and by employing the constrained LHS.
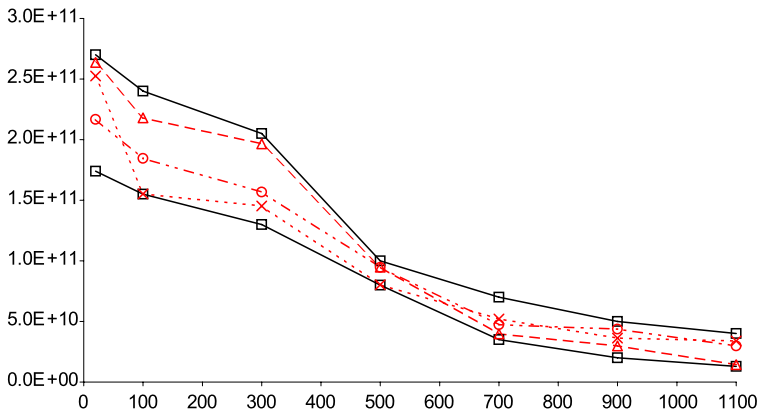
Fig. 7 Example of $n = 3$ samples of the Young's modulus obtained by cLHS (*dashed lines*). The upper and the lower curves (*solid lines*) represent the bounds of the domain for these $p = 7$ variables

However, some model calculations do not converge when using non-physical evolution of the material properties (i.e. unconstrained LHS). Moreover, performing model calculations with non-monotonic evolutions would be an aberration from a physical point of view.

6 Conclusion

In this paper, we have proposed a new algorithm (called cLHS) allowing to obtain samples of several variables constrained by some inequality relations. This situation can frequently arise in application cases, while very few works have been devoted to this issue. The cLHS algorithm allows to satisfy the inequality constraint while leaving unchanged the one-dimensional marginals that we have defined for each variable. In order to honor these one-dimensional marginals, the LHS-based technique has been preferred. To our knowledge, this inequality constraint problem has not been studied for the LHS building issue.

We have shown the interest of this algorithm in an application case involving welding simulation model. The cLHS algorithm has proven its efficiency to sample a multi-dimensional variable taking under consideration its physical nature.

The current cLHS algorithm has one main drawback. When the minimal (respectively, maximal) bounds of the two constrained variables move closer, the space filling properties of the sample points deteriorate: the sample points are gathered along the inequality frontier line. The cLHS is therefore efficient if the bounds between the variables are sufficiently distant. A constraint intensity measurement, noticed γ, has been defined in order to quantify this effect. Moreover, a linear relation has been proposed between γ and the correlation coefficient of the constrained variables. This allows to a priori know (from the variable bound values), the effects of the inequality constraint in terms of correlation of the simulated sample.

In a future work, it will be interesting to quantify this phenomenon by linking a space filling measure (as the discrepancy) with the bound values. More generally,

if we want to suppress this undesired effect, the LHS framework has to be left out. A first idea would be to work with entropy-based designs by optimizing the entropy on one-dimensional marginals. Developing design optimization algorithms under inequality constraints would be an interesting research way.

If more than two variables are under study, the cLHS algorithm is limited to inequality constraints defined in a sequential order. A complex inequality constraint could involve more than two variables (e.g. $X_1 < X_2 + X_3$) or could be nonsequential (e.g. $X_1 < X_2$ and $X_1 < X_3$). If such situations are identified in specific application cases, there is no doubt that some extensions of the cLHS algorithm are possible.

References

Asserin, O., Loredo, A., Petelet, M., Iooss, B.: Global sensitivity analysis in welding simulations—What are the material data you really need? Finite Elem. Anal. Des. (2009, submitted). Available at URL: http://hal.archives-ouvertes.fr/hal-00419162/fr/

Borgonovo, E.: Sensitivity analysis of model output with input constraints: A generalized rationale for local methods. Risk Anal. **28**, 667–680 (2008)

Bursztyn, D., Steinberg, D.: Comparison of designs for computer experiments. J. Stat. Plan. Inference **136**, 1103–1119 (2006)

De Rocquigny, E., Devictor, N., Tarantola, S. (eds.): Uncertainty in Industrial Practice. Wiley, New York (2008)

Fang, K.-T., Li, R., Sudjianto, A.: Design and Modeling for Computer Experiments. Chapman & Hall/CRC, London (2006)

Gentle, J.: Random Number Generation and Monte Carlo Methods. Springer, Berlin (2003)

Helton, J., Davis, F.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab. Eng. Syst. Saf. **81**, 23–69 (2003)

Iman, R., Conover, W.: A distribution-free approach to inducing rank correlation among input variables. Commun. Stat. **11**(3), 311–334 (1982)

Iooss, B., Boussouf, L., Feuillard, V., Marrel, A.: Numerical studies of the metamodel fitting and validation processes. Int. J. Adv. Syst. Meas. **3**, 11–21 (2010)

Jourdan, A., Franco, J.: Optimal Latin hypercube designs for the Kullback-Leibler criterion. Adv. Stat. Anal. **94**(4), 341–351 (2010)

Kleijnen, J.: Design and Analysis of Simulation Experiments. Springer, Berlin (2008)

Kurowicka, D., Cooke, R.: Uncertainty Analysis with High Dimensional Dependence Modelling. Wiley, New York (2006)

Levy, S., Steinberg, D.: Computer experiments: A review. Adv. Stat. Anal. **94**(4), 311–324 (2010)

McKay, M., Beckman, R., Conover, W.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**, 239–245 (1979)

Owen, A.: A central limit theorem for Latin hypercube sampling. J. R. Stat. Soc. B **16**, 140–149 (1992)

Park, J.-S.: Optimal Latin-hypercube designs for computer experiments. J. Stat. Plan. Inference **39**, 95–111 (1994)

Petelet, M.: Analyse de sensibilité globale de modèles thermomécaniques de simulation numérique du soudage. Thèse de l'Université de Bourgogne (2007)

Petelet, M., Asserin, O., Iooss, B., Loredo, A.: Echantillonnage LHS des propriétés matériau des aciers pour l'analyse de sensibilité globale en simulation numérique du soudage. In: Actes de MATERI-AUX, Dijon, France, 2006

Pistone, G., Vicario, G.: Comparing and generating Latin Hypercube designs in Kriging models. Adv. Stat. Anal. **94**(4), 353–366 (2010)

Saltelli, A., Chan, K., Scott, E. (eds.): Sensitivity Analysis. Wiley Series in Probability and Statistics. Wiley, New York (2000)

Simpson, T., Lin, D., Chen, W.: Sampling strategies for computer experiments: Design and analysis. Int. J. Reliab. Appl. **2**, 209–240 (2001a)

Simpson, T., Peplinski, J., Kock, P., Allen, J.: Metamodel for computer-based engineering designs: Survey and recommendations. Eng. Comput. **17**, 129–150 (2001b)

Sobol, I.: Uniformly distributed sequences with additional uniformity property. USSR Comput. Math. Math. Phys. **16**, 236–242 (1976)

Stein, M.: Large sample properties of simulations using Latin hypercube sampling. Technometrics **29**, 143–151 (1987)

Volkova, E., Iooss, B., Van Dorpe, F.: Global sensitivity analysis for a numerical model of radionuclide migration from the RRC "Kurchatov Institute" radwaste disposal site. Stoch. Environ. Res. Risk Assess. **22**, 17–31 (2008)