# Recursive estimation procedure of Sobol' indices based on replicated designs

Laurent Gilquin, Élise Arnaud, Clémentine Prieur, Hervé Monod

# Recursive estimation procedure of Sobol' indices based on replicated designs

Laurent Gilquin[a,b,*], Elise Arnaud[b], Clémentine Prieur[b], Hervé Monod[c]

[a] *Inria Grenoble - Rhône-Alpes, Inovallée, 655 avenue de l'Europe, 38330 Montbonnot*
[b] *Univ. Grenoble Alpes, Jean Kunzmann Laboratory, F-38000 Grenoble, France*
*CNRS, LJK, F-38000 Grenoble, France, Inria*
[c] *MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-En-Josas, France*

**Abstract**

In the context of global sensitivity analysis, the replication procedure allows to estimate Sobol' indices at an efficient cost. However this method still requires a large number of model evaluations. In this paper, we consider the ability of increasing the number of evaluation points, thus the accuracy of estimates, by rendering the replication procedure recursive. The key feature of this approach is the construction of structured space-filling designs. For the estimation of first-order indices, we exploit a nested Latin Hypercube already introduced in the literature. For the estimation of closed second-order indices, two methods are proposed to construct iteratively an orthogonal array. One of the two leads to a partition of the coordinate space over a Galois field. Various space-filling criteria are used to evaluate our designs.

*Keywords:* sensitivity analysis, Sobol' indices, space-filling designs, Orthogonal Array, recursive estimator

## 1. Introduction

Mathematical models used in various fields are often quite complex. The behavior of some of these models may only be explored through the study of uncertainties propagated from their inputs. Sensitivity analysis studies how the uncertainty on an output of a mathematical model can be attributed to sources of uncertainty among the inputs. Global sensitivity analysis is a

*Corresponding author
*Email address:* laurent.gilquin@inria.fr (Laurent Gilquin)

common practice to identify influent inputs of a complex model and detect the potential interactions between them. Among the large number of available approaches, the variance-based method introduced by Sobol' [1] allows to calculate sensitivity indices called Sobol' indices. The method is based on a variance decomposition of the model's output into fractions which can be attributed to sets of inputs, assuming that the uncertainty on the sets of inputs is modeled by independent probability distributions. The influences of each set are summarized by the Sobol' indices which are scalars between 0 and 1. An index close to 1 means that the set of inputs is influent. At the opposite, an index equal to 0 means that the set of inputs is not correlated to the output. One can distinguish first-order indices that estimate the main effect of each set of inputs from higher-order indices that estimate the corresponding order of interactions between sets of inputs. Various procedures have been proposed in the literature (see Saltelli [2] for a survey) to estimate Sobol' indices. Unfortunately, these procedures require a significant number of model runs. This cost can be prohibitive for expansive models. A solution to reduce this cost lies in the use of replicated designs.

The notion of replicated designs to estimate first-order Sobol' indices probably goes back to McKay [3] and appears later in Mara *et al.* [4]. These latter authors combine replicated designs with "pick-freeze" estimators [1] to estimate first-order Sobol' indices. This procedure has been further deeply studied and generalized in Tissot and Prieur [5] to the estimation of closed second-order indices. The generalization to closed second-order Sobol' indices relies on the replication of Orthogonal Arrays (see Lemieux [6] or Owen [7]). The procedure in Mara *et al.* [4] has the major advantage of reducing drastically the estimation cost as the number of runs (one design of size $n$ and a replication of this design) becomes independent of the input space dimension. However, if the input space is not properly explored (if $n$ is too small), the Sobol' indices estimates may not be accurate enough.

To address this issue, we need a procedure allowing to add new sample points to the initial design. We also need a recursive formulation for our Sobol' index estimator. Adding new sample points is an easy task in case the initial design is composed with independent and identically distributed points in $\mathbb{R}^d$. However, in the replication procedure, as it has been introduced in [5], the initial design is a Latin Hypercube Sampling (resp. an Orthogonal Array (OA)) for the estimation of first- (resp. closed second-) order indices. An algorithm for the construction of nested Latin Hypercubes has been proposed by Qian [8]. It allows to double the size of the sample at each update. Our

approach to render the replication procedure recursive for the estimation of first-oder Sobol' indices is based on this construction.

In this paper, we propose two new algorithms to construct nested Orthogonal Arrays. These two algorithms start from an initial OA of size $n$ and update it sequentially by adding $n$ new points at each step. At step $k$, $k \geq 1$, the updated design is of size $k \times n$ and possesses an OA structure. An intuitive construction consists in duplicating the initial design at each step. However, this approach is not satisfying as there is no benefit in exploring redundant input space locations. In ([9], [10] and [11]) families of "nested Orthogonal Arrays" are constructed. The constructions proposed in these papers suffer from at least one of the following drawbacks:

- The size $n$ of the initial design is rather large, hence at each step a large number $n$ of new points is added.
- The constructions deal only with specific values of the input space dimension. There exists indeed an upper bound on the number of columns of the designs constructed.
- The discretization is not the same in each dimension, more precisely only one dimension is finely discretized.

The two algorithms proposed in this paper do not suffer from these drawbacks.

The paper is organized as follows. In Section 2, backgrounds are given on Sobol' indices and on the replication estimation procedure. Then, the process rendering the replication procedure recursive is detailed. Section 3 deals with the construction of the nested space-filling structures: nested Latin Hypercube and nested Orthogonal Arrays of strength two. In Section 4, regularity and uniformity properties of these two designs are studied. The end of this section is devoted to the application of the recursive replication procedure. A classical case study is addressed to demonstrate the interest in using the recursive approach.

## 2. Replicated designs and recursive estimation of sensitivity indices

### *2.1. Definition of Sobol' indices*

Consider the following model defined from a black box perspective:

$$f : \left\{ \begin{array}{ccc} \mathbb{R}^d & \rightarrow & \mathbb{R} \\ x = (x_1, \ldots, x_d) & \mapsto & y = f(x) \end{array} \right. \tag{1}$$

where $y$ is the output of the model $f$, $x$ the input vector and $d$ the dimension of the input space. Denote by $\subsetneq$ the proper (strict) inclusion symbol and by $\subseteq$ the inclusion symbol.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We model the uncertainty on the inputs by a random vector $X = (X_1, \ldots, X_d)$ whose components are independent. Let $P_X = P_{X_1} \otimes \ldots \otimes P_{X_d}$ denote the distribution of $X$. We assume that $f \in \mathbb{L}^2(P_X)$. The model $f$ can then be uniquely decomposed into summands of increasing dimensions (functional ANOVA decomposition [1, 12]):

$$f(X) = f_0 + \sum_i f_i(X_i) + \sum_{i<j} f_{ij}(X_i, X_j) + \cdots + f_{1\ldots d}(X_1, \ldots, X_d) \ , \quad (2)$$

where $\mathrm{E}[f_I(X_I) f_K(X_K)] = 0, \ \forall \ (I, K) \subseteq \{1, \ldots, d\}^2, \ I \neq K$. This implies that $f_0 = \mathrm{E}[Y]$ and that the components are mutually orthogonal with respect to $P_X$. Let $I \subseteq \{1, \ldots, d\}$, each component is defined by:

$$f_I(X_I) = \mathrm{E}[Y|X_I] - \sum_{J \subsetneq I} f_J(X_J).$$

The functional decomposition can be used to measure the global sensitivity of the output $Y$ to the input $X_i$. By squaring and integrating (2), due to orthogonality we get:

$$V = \sum_i V_i + \sum_{i<j} V_{ij} + \cdots + V_{1,\ldots,d} \quad (3)$$

where $$V_I = \mathrm{Var}[f_I(X_I)] = \mathrm{Var}[\mathrm{E}[Y|X_I]] - \sum_{J \subsetneq I} V_J$$

and $$V = \mathrm{Var}[Y] \, .$$

Resulting from this decomposition, the Sobol' indices are defined by:

$$S_I = \frac{V_I}{V} \, .$$

Let $|I|$ denote the cardinal of $I$. The Sobol' index $S_I$ measures the contribution to $V$ of the $|I|^{\text{th}}$-order interaction between the $\{X_i\}$, $i \in I$. Closed Sobol' indices are defined by:

$$\underline{S_I} = \frac{\mathrm{Var}[\mathrm{E}[Y|X_I]]}{V} \ .$$

The closed Sobol' index $\underline{S_I}$ measures the contribution of the $X_i$, $i \in I$ by themselves or in interaction with each other. As an example, if there exist $i \neq j$, $(i,j) \in \{1,\dots,d\}^2$ such that $I = \{i,j\}$, then $\underline{S}_{ij} = S_{ij} + S_i + S_j$. At last, note that:

$$\sum_{I \subseteq \{1,\dots,d\}, I \neq \varnothing} S_I = 1,$$

allowing a direct interpretation of the value of each index.

Most of the time, there does not exist any explicit formulation of Sobol' indices which thus need to be estimated.

*2.2. Estimation of Sobol' indices*

Let $X$ and $X'$ be two independent vectors distributed as the inputs vector. We define:

$$X = (X_1, \dots, X_d),\ X_I = (X_1^*, \dots, X_d^*),$$

where $X_i^* = X_i$ if $i \in I$ and $X_i^* = X'_i$ otherwise. We also define $Y$ and $Y_I$ the corresponding model outputs: $Y = f(X),\ Y_I = f(X_I)$. Then, the following expression (see [13, Lemma 1.2]) for $\underline{S_I}$ is obtained:

$$\underline{S_I} = \frac{\mathrm{Cov}(Y, Y_I)}{\mathrm{Var}[Y]}.$$

To estimate $\underline{S_I}$, we proceed as in [1] and introduce two designs, each of size $n$:

$$\mathbf{X} = \begin{pmatrix} X_1^1 \dots X_i^1 \dots X_d^1 \\ \vdots \quad \vdots \quad \vdots \\ X_1^j \dots X_i^j \dots X_d^j \\ \vdots \quad \vdots \quad \vdots \\ X_1^n \dots X_i^n \dots X_d^n \end{pmatrix} = \begin{pmatrix} X^1 \\ \vdots \\ X^j \\ \vdots \\ X^n \end{pmatrix}, \mathbf{X'} = \begin{pmatrix} X'^1_1 \dots X'^1_i \dots X'^1_d \\ \vdots \quad \vdots \quad \vdots \\ X'^j_1 \dots X'^j_i \dots X'^j_d \\ \vdots \quad \vdots \quad \vdots \\ X'^n_1 \dots X'^n_i \dots X'^n_d \end{pmatrix} = \begin{pmatrix} X'^1 \\ \vdots \\ X'^j \\ \vdots \\ X'^n \end{pmatrix}$$

From these two designs we construct the $n \times d$ matrix $\mathbf{X}_I = \begin{pmatrix} {X_I}^1 \\ \vdots \\ {X_I}^j \\ \vdots \\ {X_I}^n \end{pmatrix}$ with $X_I^j = (X_1^{*,j}, \dots, X_d^{*,j})$, $j = 1, \dots, n$ where $X_j^{*,j} = X_i^j$ if $i \in I$ and $X_j^{*,j} = X'^j_i$

otherwise. We then define $\mathbf{Y} = \begin{pmatrix} Y^1 \\ \vdots \\ Y^j \\ \vdots \\ Y^n \end{pmatrix} = f(\mathbf{X})$ and $\mathbf{Y}_I = \begin{pmatrix} {Y_I}^1 \\ \vdots \\ {Y_I}^j \\ \vdots \\ {Y_I}^n \end{pmatrix} = f(\mathbf{X}_I)$. Following [14], we propose the following estimator for $\underline{S_I}$:

$$\widehat{\underline{S_I}} = \frac{\frac{1}{n}\sum_{j=1}^{n} Y^j Y_I^j - \left(\frac{1}{n}\sum_{j=1}^{n} Y^j\right)\left(\frac{1}{n}\sum_{j=1}^{n} Y_I^j\right)}{\frac{1}{n}\sum_{j=1}^{n} (Y^j)^2 - \left(\frac{1}{n}\sum_{j=1}^{n} Y^j\right)^2}\ , \tag{4}$$

Other choices are possible for the estimator. We focus on (4) whose asymptotic properties have been studied in [13].

The main drawback of the aforementioned approach is the high number of model evaluations needed. Estimating all first- (resp. all closed second-) order Sobol' indices requires $n(d+1)$ (resp. $n(\binom{d}{2}+1)$) model evaluations. The larger $n$, the more accurate the estimation of Sobol' indices. For models with substantial computational times, this solution becomes rapidly intractable in case of large input space dimension. Some improvements have been introduced by Saltelli [2] to reduce the number of evaluations but with a cost still depending on the dimension $d$ of the input space.

A solution to reduce drastically this costs lies in the use of replicated designs. In the following, we review the procedure, denoted by replication procedure, based on replicated designs to estimate first- or closed second-order Sobol' indices. For the sake of clarity of the paper, we assume that the variables $X_1, \ldots, X_d$ are independent random variables uniformly distributed on $[0, 1]$. The generalization to other product distributions is provided in Remark 1.

*2.2.1. Replication procedure for first-order indices*

Mara and Rakoto Joseph [4] proposed a strategy based on replicated designs to estimate all first-order Sobol' indices with only two designs each of size $n$. More precisely, denote by $\mathbf{X} = (X_i^j)_{1\leq j\leq n,\ 1\leq i\leq d}$ the first design. The second design **X'** is said to be replicated from **X** if it is obtained by permuting the entries of each column: $\mathbf{X'} = (X_i^{\pi_i(j)})_{1\leq j\leq n,\ 1\leq i\leq d}$, where $\pi_1, \ldots, \pi_d$ are $d$ permutations on $\{1, \ldots, n\}$ selected randomly and independently without

replacement. Let $\mathbf{Y}$ and $\mathbf{Y'}$ be the vectors of evaluations associated to those two designs:

$$\mathbf{Y} = (Y^1, \ldots, Y^n)^T, \ \mathbf{Y'} = (Y'^1, \ldots, Y'^n)^T,$$

where $T$ denotes the transpose operator. To estimate the Sobol' index $S_k$, $k \in \{1, \ldots, d\}$, a new vector $\widetilde{Y}_k$ is constructed by re-arranging $Y$ with the permutation $\pi_k$:

$$\widetilde{Y}_k = \left(Y^{\pi_k(1)}, \ldots, Y^{\pi_k(n)}\right)^T = \left(f(X^{\pi_k(1)}), \ldots, f(X^{\pi_k(n)})\right)^T.$$

Remark that:

$$X^{\pi_k(j)} = \left(X_1^{\pi_k(j)}, \ldots, X_{k-1}^{\pi_k(j)}, X'^j_k, X_{k+1}^{\pi_k(j)}, \ldots, X_d^{\pi_k(j)}\right).$$

All the coordinates have been re-sampled except the $k$-th one which has been frozen. $S_k$ is then estimated by replacing $Y$ and $Y_I$ in Formula (4) by $\widetilde{Y}_k$ and $Y'$. With this approach, known as replication procedure, the computation cost for all the first-order Sobol' indices has been reduced to $2 \times n$ (evaluations on $X$ and $X'$ only). There are various choices for the two replicated designs $\mathbf{X}$, $\mathbf{X'}$. In [4], $\mathbf{X}$ and $\mathbf{X'}$ are composed with iid rows. In [5], the authors propose the use of Latin Hypercube Sampling (LHS) insuring a better exploration of the input space. We recall below the definition of a LHS:

**Definition 1 (Latin Hypercube Sampling).** Denote by $\Pi_n$ the set of all the permutations on $\{1, \ldots, n\}$ and let $\pi_1, \ldots, \pi_d$ be $d$ independent random variables uniformly distributed on $\Pi_n$. We say that $(L_i^j)_{1 \leq j \leq n, \ 1 \leq i \leq d}$ is a Latin Hypercube if:

$$L_i^j = \pi_i(j), \ \forall i \in \{1, \ldots, d\}, \forall j \in \{1, \ldots, n\}.$$

We denote by $\mathcal{LH}(n, d)$ the set of all $n \times d$ Latin Hypercubes. Let $U_i^j$ be independent random variables uniformly distributed on $[0, 1]$ and independent of the $\pi_i$. $\mathbf{X} = (X_i^j)_{1 \leq j \leq n, \ 1 \leq i \leq d}$ is a randomized Latin Hypercube Sampling (LHS) if:

$$X^j = \left(X_1^j = \frac{L_1^j - U_1^j}{n}, \ldots, X_d^j = \frac{L_d^j - U_d^j}{n}\right) \tag{5}$$

Using Definition 1 above, permuting the entries of each column of a LHS provides a new LHS. As a result, two LHS can be used in place of $\mathbf{X}$ and $\mathbf{X'}$ to estimate all first-order Sobol' indices.

*2.2.2. Replication procedure for closed second-order indices*

For the estimation of closed second-order Sobol' indices, the previous idea of applying column-wise permutations to replicate the first design is not enough. Indeed, to estimate second-order Sobol' indices, a way to "freeze" each subset of two variables has to be found. A solution was proposed by Tissot and Prieur in [5]. It relies on the use of Orthogonal Arrays. The introduction of Orthogonal Arrays probably goes back to Kishen [15] and was further extended by Rao [16]. Let us consider the following definition [17, Definition 1.1]:

**Definition 2 (Orthogonal Array (OA)).** A $N \times d$ array $A$ with values from a set $S$ of cardinality $q$ is said to be an Orthogonal Array with $q$ levels, strength $t$ ($0 \leq t \leq d$) and index $\lambda$ if every $N \times t$ sub-array of $A$ contains each $t$-tuple based on $S$ exactly $\lambda$ times as a row. The Orthogonal Array $A$ satisfies $N = \lambda q^t$. It is denoted by $OA_\lambda(q, d, t)$.

Here, the space $S$ is identified as the Galois field of order $q$, denoted by $GF(q)$, where $q$ is a prime number or prime power number ($q = p^\alpha$, $p$ prime and $\alpha \in \mathbb{N}$). For the rest of the paper, once an OA is constructed its levels are substituted by $1, \ldots, q$ where $q$ corresponds to the number of points into which each input is discretized. For the construction of an Orthogonal Array $OA_\lambda(q, d, t)$ we invite the reader to consult the different constructions proposed in [17, Theorem 3.1, Lemma 6.12].

The strategy proposed by Tissot and Prieur [4] allows to estimate all closed second-order Sobol' indices with two designs each of size $q^2$. The creation of these two designs relies on the construction of an Orthogonal Array $OA_1(q, d, 2)$ denoted by $A = (A_i^j)_{1 \leq j \leq q^2,\ 1 \leq i \leq d}$, $A_i^j \in \{1, \ldots, q\}$. The first design $\mathbf{X} = (X_i^j)_{1 \leq j \leq q^2,\ 1 \leq i \leq d}$, is a randomized OA constructed as follows (see [5] for further details):

$$X^j = \left( X_1^j = \frac{A_1^j - U_1^{A_1^j}}{q}, \ldots, X_d^j = \frac{A_d^j - U_d^{A_d^j}}{q} \right) \tag{6}$$

where the $\{U_i^j\}_{1 \leq j \leq q^2,\ 1 \leq i \leq d}$ are independent random variables uniformly distributed on $[0, 1]$ and independent from $A$. The construction of the second design relies on the replication of $A$. This can be achieved through the use of the following definition:

**Definition 3 (replicated Orthogonal Array).** Let $L \in \mathcal{LH}(q,d)$. Let $A$ be an Orthogonal Array $OA_\lambda(q,d,2)$. Define $A'$ as follows:

$$A'^{j}_{i} = L^{A^{j}_{i}},\ j \in \{1,\dots,q^2\},\ i \in \{1,\dots,d\}$$

Then $A'$ is called an Orthogonal Array $OA_\lambda(q,d,2)$ replicated from $A$. Denote by $\Diamond$ this operation: $A' = \Diamond(A,L)$.

By applying Definition 3, a second Orthogonal Array $A'$ is replicated from $A$. The second design $\mathbf{X'} = (X'^{j}_{i})_{1\le j\le q^2,\ 1\le i\le d}$ is constructed using (6) with $A'$ in place of $A$ and with the same random variables $\{U^{j}_{i}\}_{1\le j\le q^2,\ 1\le i\le d}$:

$$X'^{j} = \Big(X'^{j}_{1} = \frac{A'^{j}_{1} - U_1^{A'^{j}_{1}}}{q}, \dots, X'^{j}_{d} = \frac{A'^{j}_{d} - U_d^{A'^{j}_{d}}}{q}\Big).$$

Let $\mathbf{Y}$ and $\mathbf{Y'}$ be the associated vectors of model outputs:

$$\mathbf{Y} = \Big(f(X^1),\dots,f(X^{q^2})\Big)^T,\ \mathbf{Y'} = \Big(f(X'^1),\dots,f(X'^{q^2})\Big)^T,$$

the estimation of the Sobol' index $\underline{S}_{k,l}$ , $(k,l) \in \{1,\dots,d\}^2$, relies on a specific ordering of $\mathbf{Y}$ and $\mathbf{Y'}$. Denote by $A_k$ the $k$-th column of $A$. The Sobol' index $\underline{S}_{k,l}$, $k,l \in \{1,\dots,d\}^2$, is obtained by re-arranging the values of $\mathbf{Y}$ (resp. $\mathbf{Y'}$) such that $A_k \times A_l$ (resp. $A'_k \times A'_l$) is sorted in ascending lexicographic order. The two resulting vectors are then used in formula (4) in place of $\mathbf{Y}$ and $\mathbf{Y}_I$ to estimate $\underline{S}_{k,l}$. This procedure allows to estimate all closed second-order Sobol' indices at a total cost of $2 \times q^2$ evaluations of the model.

**Remark 1.** In this section, the constructions of designs $\mathbf{X}$ and $\mathbf{X'}$ for either first- or second-order Sobol' indices estimation are only valid when dealing with variables $X_1,\dots,X_d$ independent and uniformly distributed on $[0,1]$. However, this construction can be generalized to other non-uniform distributions. Denote by $F_1,\dots,F_d$ the cumulative distribution functions of $X_1,\dots,X_d$. Denote by $\mathbf{X}_g$ and $\mathbf{X'}_g$ the two designs constructed for the general case, they are defined as follows:

$$\mathbf{X}^{j}_{g} = (F_1^{-1}(X^{j}_{1}),\dots,F_d^{-1}(X^{j}_{d}))$$

$$\mathbf{X'}^{j}_{g} = (F_1^{-1}(X'^{j}_{1}),\dots,F_d^{-1}(X'^{j}_{d}))$$

where $F_1^{-1},\dots,F_d^{-1}$ are the inverse cumulative distribution functions (quantile functions) of $X_1,\dots,X_d$. Then the estimation is performed with $\mathbf{X}_g$ and $\mathbf{X'}_g$ in place of $\mathbf{X}$ and $\mathbf{X'}$.

*2.3. Recursive procedure*

The recursive approach presented in this section consists in rendering the replication procedure recursive. This approach requires first to write down a recursive formula for the Sobol' index estimator. Recall the expression of the Sobol' index:

$$\underline{S_I} = \frac{\mathrm{Cov}[Y, Y_I]}{\mathrm{Var}[Y]} = \frac{\mathrm{E}[YY_I] - \mathrm{E}[Y]\ \mathrm{E}[Y_I]}{\mathrm{Var}[Y]} \tag{7}$$

where $Y$ and $Y_I$ are the model outputs calculated from the two replicated designs. At each step of the recursive procedure, both designs are augmented with a new set of points. Denote by $D_l$, $D'_l$ these two designs at the $l$-th step and by $n_l$ their number of points. At the $(l+1)$-th step of the recursive estimation, a new set of points, denoted by $D_{new,l+1}$, of size $m_{l+1}$ is added to $D_l$ to form $D_{l+1}$. Thus, $D_{l+1} = D_l \cup D_{new,l+1}$ and $n_{l+1} = n_l + m_{l+1}$. $D_{new,l+1}$ is then replicated (see Sections 2.2.1, 2.2.2) and its replication serves to increase $D'_l$ to form $D'_{l+1}$. At step $l$, the Sobol' index $\underline{S_I}$ is estimated by the family of recursive estimators defined as follows: $\widehat{\underline{S_I}}^{(l)} = \dfrac{\phi_l - \psi_l \xi_l}{V_l}$, where:

$$\begin{cases} \phi_l = \frac{1}{n_l} \sum_{j=1}^{n_l} Y^j Y_I^j, \\ \psi_l = \frac{1}{n_l} \sum_{j=1}^{n_l} Y^j, \\ \xi_l = \frac{1}{n_l} \sum_{j=1}^{n_l} Y_I^j, \\ V_l = \frac{1}{n_l - 1} \sum_{j=1}^{n_l} (Y^j - \psi_l)^2. \end{cases}$$

Using the recursive formula of $\phi_{l+1}$, $\psi_{l+1}$, $\xi_{l+1}$ and $V_{l+1}$, the following recursive formula is obtained:

$$\widehat{\underline{S_I}}^{(l+1)} = \frac{\phi_{l+1} - \psi_{l+1}\xi_{l+1}}{V_{l+1}} \tag{8}$$

where:

$$\begin{cases} \phi_{l+1} = n_l\phi_l + m_{l+1}\sum_{j=n_l+1}^{n_{l+1}} Y^j Y_I^j, \\ \psi_{l+1} = n_l\psi_l + m_{l+1}\sum_{j=n_l+1}^{n_{l+1}} Y^j, \\ \xi_{l+1} = n_l\xi_l + m_{l+1}\sum_{j=n_l+1}^{n_{l+1}} Y_I^j, \\ V_{l+1} = (n_l-1)V_l + n_l{\psi_l}^2 + \sum_{j=n_l+1}^{n_{l+1}}(Y^j)^2 - \frac{1}{n_{l+1}}(n_l\psi_l + m_{l+1}\sum_{j=n_l+1}^{n_{l+1}} Y^j)^2, \\ C^{l+1} = \frac{n_{l+1}-1}{n_{l+1}}. \end{cases}$$

Then, the recursive procedure relies on the construction of a nested space-filling structure design (discrete LHS for first-order indices and discrete OA for closed second-order indices). This structured design is partitioned into subsets denoted by blocks in the following. Algorithm 1 summarizes the main steps of our recursive estimation procedure. The variable *order* indicates whether first-order (value 1) or closed second-order (value 2) indices are estimated. The randomization applied in Step 3 of Algorithm 1 ensures that each point of the set $D_{new,l+1}$ is uniformly distributed in $[0,1]^d$. The method of randomization is detailed in the next section. The form of the stopping criterion is discussed in Section 4.3.

In the next section, we detail the construction of the nested space-filling structured designs for the estimation of either first-order or closed second-order Sobol' indices. In both cases, the construction ensures that at each step $l$ of the recursive procedure, the nested design $D_l$ possesses a space-filling structure.

## 3. Space filling construction of the blocks

For the estimation of first-oder indices, the nested space-filling structured design is a nested Latin Hypercube. The number of blocks partitioning the structure has to be specified beforehand. The discretization of each input is further refined while adding new blocks. For the estimation of closed second-oder indices, the nested space-filling structured design is an Orthogonal Array of strength two and index $\lambda > 1$. The number of blocks is iteratively augmented. However, the discretization of each input is the same for all blocks.

**Algorithm 1:** Recursive estimation procedure

1. $if$ ($order$ == 1): Construct the nested space-filling structured design given a maximum number of runs
2. Set: $l \leftarrow 0$, $D_0 \leftarrow \emptyset$, $\widehat{\underline{S_I}}^{(0)} \leftarrow 0$
3. $while$ ( ! stopping criterion):
   - 3.1 $if$ ($order$ == 1): Select a block of the nested space-filling structured design,
     $if$ ($order$ == 2): Create a block of the nested space-filling structured design
   - 3.2 Randomize the block to obtain $D_{new,l+1}$.
   - 3.3 Replicate $D_{new,l+1}$ using the replication procedure of Section 2.2.1 or 2.2.2
   - 3.4 $for$ each index $\underline{S_I}$:
     - 3.4.1 Compute $Y$ and $Y_I$ from $D_{new,l+1}$ and its replication.
     - 3.4.2 Evaluate $\widehat{\underline{S_I}}^{(l)}$ with (8).
   - 3.5 $D_{l+1} \leftarrow D_l \cup D_{new,l+1}$
   - 3.6 $l \leftarrow l + 1$
4. $return$ the estimated Sobol' indices

*3.1. Nested Latin Hypercube for first-order indices*

A way to augment the number of points while conserving a discrete Latin Hypercube structure has been proposed by Qian [8]. A nested Latin Hypercube is a discrete Latin Hypercube that is partitioned into sets of points referred here as blocks. Each of this block possesses a Latin Hypercube structure when projected onto a less refined grid. As an illustration, a two dimensional nested Latin Hypercube with 3 blocks is presented in Figure 1 (a). Each block represented in Figure 1 (b), (c), (d) possesses a discrete Latin Hypercube structure in their respective grid (delimited by the dark lines).

The algorithm underlying the construction of a nested Latin Hypercube can be found in [8, Section 5]. With this algorithm, the blocks constituting the nested Latin Hypercube possess at the minimum the following number

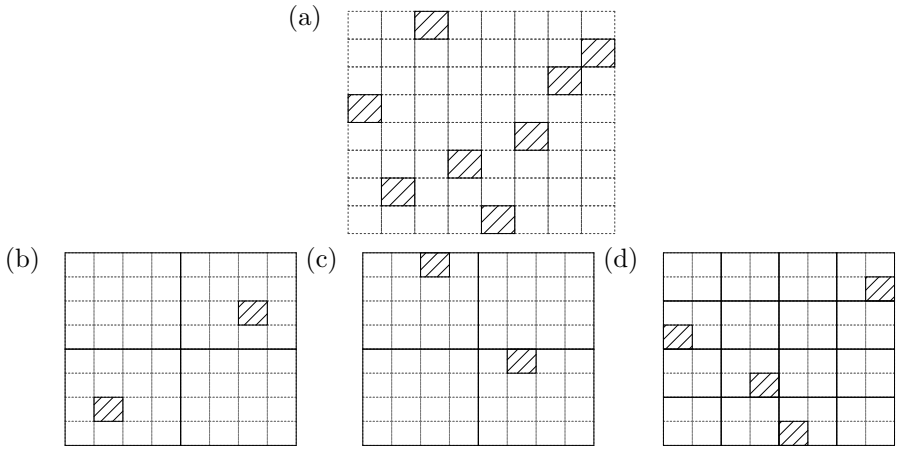



Figure 1: Final nested Latin Hypercube with 3 blocks. (a) Nested Latin Hypercube. (b) first block. (c) second block. (d) third block. (b), (c) and (d) are derived from the construction of (a).

of points: (1,1,2,$2^2$,$2^3$,...). It is important to note that the construction of a nested Latin Hypercube is not sequential, the whole structure must be constructed at once. However, the construction ensures that the concatenation of the $k$-th block with the formers still possesses a discrete Latin Hypercube structure (see [8] for details).

The randomization of a block (Step 2.2 of Algorithm 1) is performed using a formula similar to (5). As an example, the randomization of the blocks in Figure 1 would consist to sample points inside the hatched areas. We formalize thereafter this procedure. Denote by $B^l = (B_i^{j,l})_{1\leq j\leq m_l,\ 1\leq i\leq d}$ the $l$-th block of the nested Latin Hypercube. Let $m$ be the number of points of the final nested Latin Hypercube. Denote by $D_{new,l+1}$ the design resulting from the randomization of $B^l$. $D_{new,l+1}$ is defined as follows:

$$D_{new,l+1}{}^j = \Big(D_{new,l+1}{}_1^j = \frac{B_1^{j,l} - U_1^{j,B^l}}{m}, \ldots, D_{new,l+1}{}_d^j = \frac{B_d^{j,l} - U_d^{j,B^l}}{m}\Big),$$

$j = \{1, \ldots, m_l\}$, where $U_i^{j,B^l}$ are independent random variables uniformly distributed on $[0,1]$ and independent of $B^l$. $D_{new,l+1}$ is then used within Algorithm 1 to estimate all first-order Sobol' indices. The cost of the estimation equals $2 \times (\sum_{l=1}^{K} m_l)$ where $K$ is the step at which the recursive estimation has terminated.

*3.2. Orthogonal Array for closed second-order indices*

The nested space-filling structured design constructed for the estimation of closed second-order indices is an Orthogonal Array of strength two with index $\lambda \geq 1$, denoted by $OA_\lambda(q, d, 2)$. This $OA_\lambda(q, d, 2)$ can be partitioned into $\lambda$ blocks where each block has the geometric structure of an Orthogonal Array of strength two $OA_1(q, d, 2)$. We propose two procedures to construct such an $OA_\lambda(q, d, 2)$. The first one is an *accept-reject* procedure. The second one is called *algebraic* procedure and relies on results from arithmetic. Both methods consist in iteratively constructing the $\lambda$ blocks composing the $OA_\lambda(q, d, 2)$. As stated before, the discretization of the inputs is given by the first block and is not further refined with the addition of new blocks. However, the whole structure can be sequentially augmented.

To illustrate both construction methods, an example of an $OA_3(3, 3, 2)$, denoted by $A$, is presented in Figure 2. In each graph (a), (b), (c) is represented one of three blocks of $A$. Each block possesses the structure of an $OA_1(3, 3, 2)$. Each row of $A$ is associated with a sub-hypercube of the $d$-hypercube [1] . The idea of the construction reduces to progressively filling the $d$-hypercube with sub-hypercubes distinct from those already constructed.

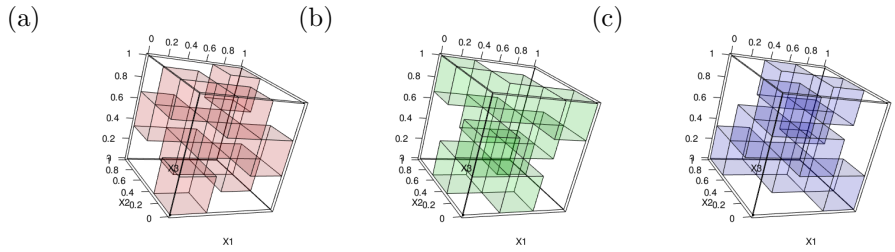



Figure 2: $OA_3(3, 3, 2)$ with 3 blocks. (a) first block. (b) second block. (c) third block.

The *accept-reject* and the *algebraic* procedures aim to construct both an $OA_\lambda(q, d, 2)$ where the rows are two by two distinct. This construction is performed iteratively during the recursive estimation procedure (Step 3.1 of Algorithm 1). For the sake of clarity, we present in Algorithm 2 a new version

[1] Consider the standard basis $\{0, \{\vec{i}, \vec{j}, \vec{k}\}\}$. Each row $(A_1^j, \ldots, A_3^j)$, $j \in \{1, \ldots, 9\}$, of $A$ is associated with the sub-hypercube $\{1/3, (A_1^j, \ldots, A_3^j)\}$ of the $d$-hypercube $[0, 1]^3$ where $1/3$ stands for the edges length of the sub-hypercube and $(A_1^j, \ldots, A_3^j)$ are the coordinates of its furthest vertex from the origin 0.

of Algorithm 1 detailing the iterative construction of the Orthogonal Array $OA_\lambda(q, d, 2)$ for the estimation of closed second-order indices. The new steps added are put in bold.

---

**Algorithm 2:** Recursive estimation procedure for closed second-order indices

---

1. **Construct $\boldsymbol{A_0}$ an $\boldsymbol{OA_1(q, d, 2)}$**
2. Set: $l \leftarrow 0$, $D_0 \leftarrow \emptyset$, $\widehat{\underline{S_I}}^{(0)} \leftarrow 0$ **and $\boldsymbol{A \leftarrow A_0}$**
3. *while* ( ! stopping criterion):
   - 3.1 Create a block of the nested space-filling structured design:
     - 3.1.1 **Construct $\boldsymbol{B}$ from $\boldsymbol{A_0}$ such that $\boldsymbol{rows(B) \cap rows(A) = \emptyset}$**
     - 3.1.2 $\boldsymbol{A \leftarrow \begin{pmatrix} A \\ B \end{pmatrix}}$
   - 3.2 Randomize the block $B$ to obtain $D_{new,l+1}$.
   - 3.3 Replicate $D_{new,l+1}$ using the replication procedure of Section 2.2.1 or 2.2.2
   - 3.4 *for* each index $\underline{S_I}$:
     - 3.4.1 Compute $Y$ and $Y_I$ from $D_{new,l+1}$ and its replication.
     - 3.4.2 Evaluate $\widehat{\underline{S_I}}^{(l)}$ with (8).
   - 3.5 $D_{l+1} \leftarrow D_l \cup D_{new,l+1}$
   - 3.6 $l \leftarrow l + 1$
4. *return* the estimated Sobol' indices

---

The Orthogonal Array $OA_\lambda(q, d, 2)$ iteratively constructed is denoted by $A$ in Algorithm 2. it is constructed by adding a new block $B$ to an initial Orthogonal Array $A_0$ at each step of the recursive estimation procedure. Each block $B$ added possesses the structure of an Orthogonal Array $OA_1(q, d, 2)$. Hence, the randomization of a block $B$ (Step 3.2 of Algorithm 2) reduces to use equation (6) to obtain a randomized OA. This randomized OA corresponds to the new sets of points $D_{new,l+1}$. With this construction, the number of new points added equals $q^2$.

The process is repeated until the stopping criterion is met. The form of

the stopping criterion is discussed in Section 4.3. The parameter $\lambda$ of the $OA_\lambda(q, d, 2)$ constructed corresponds to the total number of blocks $B$ added. The *accept-reject* and the *algebraic* procedures differ on the way the block $B$ is constructed at Step 3.1.1 of Algorithm 2. We detail below this construction for each method.

*Method 1: accept-reject.* It uses the application $\Diamond$ defined in Definition 3. Variant 1 details Step 3.1.1 of Algorithm 2 when using the *accept-reject* method. This step tests if the new block $B$ and $A$ have no common rows. This test may be computationally expansive when the dimension $d$ of the input space dimension is small ($d \leq 4$).

---

**Variant 1:** Step 3.1.1 of the *accept-reject* method

---

3.1.1 choose $L \in \mathcal{LH}(q, d)$
construct $B = \Diamond(A_0, L)$
*if* $rows(B) \cap rows(A) = \emptyset$ accept $B$
*else* discard $B$ and start again

---

*Method 2: algebraic method.* Let $C$ be the set defined as follows:

$$C = \Big\{ g = (0, 0, g_3, \ldots, g_d) \mid \forall i \geq 3,\ g_i \in GF(q) \Big\} \ \subsetneq\ GF(q)^d.$$

Denote by ${A_0}^j$ the $j$-th row of $A_0$. Variant 2 details Step 3.1.1 of Algorithm 2 when using the *algebraic* method. This method has the advantage of not relying on a computationally expensive comparison of rows. Furthermore, the maximum value taken by $\lambda$ is known and equals $q^{d-2}$ (consequence of Proposition 1 thereafter).

---

**Variant 2:** Step 3.1.1 of the *algebraic* method

---

3.1.1 choose $g \in C$
construct $B = gA_0 = \Big\{ g + {A_0}^j \mid \forall j \in \{1, \ldots, q^2\} \Big\}$
$C \leftarrow C \setminus \{g\}$

---

For both methods, the resulting Orthogonal Array $A$ constructed is an $OA_\lambda(q, d, 2)$. For the *accept-reject* method, this result is a direct consequence of Definition 3. For the *algebraic* method, this results comes from the following proposition:

**Proposition 1.** Consider $A_0$ an Orthogonal Array $OA_1(q, d, 2)$ based on the Galois field $GF(q)^d$, we have the following results:

i) $\forall g \in GF(q)^d$, $gA_0$ is an $OA_1(q, d, 2)$
ii) $\forall g, g' \in C$, such that $g \neq g'$, $gA_0 \cap g'A_0 = \emptyset$. In other words, the sets $\{gA_0\}$ form a partition of $GF(q)^d$.

*Proof.* i) Let $g = (g_1, \ldots, g_d) \in GF(q)^d$. Consider $A_{0k}, A_{0l}$ two columns of $A_0$. Denote by $E$ the group $(GF(q), +)$. Since $g_k E \times g_l E$ is isomorph to $E \times E$, the 2-tuples $(A_{0k}^j + g_k, A_{0l}^j + g_l)$ obtained after addition are all two by two distinct.

ii) The proof can be found in [18] where an Orthogonal Array is regarded as a "systematic linear code".

□

From Proposition 1, the Orthogonal Array $A$ constructed by the *algebraic* method is an $OA_\lambda(q, d, 2)$ whose rows are two by two distinct. Furthermore, as a consequence of ii), the maximum number of blocks $\lambda$ on can construct using the *algebraic* method equals the cardinality of $C$, that is $q^{d-2}$. If this maximum value is reached, the rows of $A$ form a partition of the coordinate space $GF(q)^d$.

The cost of the recursive estimation of all closed second-order indices equals $2 \times K \times q^2$ where $q$ refers to the levels of the initial Orthogonal Array $A_0$ and $K$ is the step at which the recursive replication procedure has terminated.

In the next section, the space-filling properties of the designs $D_l$ augmented at each step of the recursive replication procedure are studied. The properties of the design $D_l$ constructed for the estimation of first-order indices are compared to those of low discrepancy sequences such as the Sobol' sequence.

## 4. Space-filling properties and applications

### *4.1. Sobol' sequence*

Low discrepancy sequences are sets of points sampled so as to approximate as close as possible a uniform distribution. These sequences are known to achieve both uniformity and regularity properties. Here we focus on the low discrepancy sequence introduced by Sobol' and denoted by Sobol' sequence. The construction of this sequence can be found in [6, Section 5.4.1]. The Sobol' sequence is strongly related to the concept of $(s_1, \ldots, s_d)$-equidistribution in base 2. This notion is defined as follows:

**Definition 4 (Equidistribution).** Let $s_1, \ldots, s_d$ be nonnegative integers and $s = s_1 + \cdots + s_d$. A set of $n = 2^k$ points is $(s_1, \ldots, s_d)-$equidistributed in base 2 if every elementary interval of the form:

$$J(r) = \prod_{l=1}^{d} \left[ \frac{r_l}{2^{s_l}}, \ \frac{r_l + 1}{2^{s_l}} \right)$$

where $0 \leq r_l \leq 2^{s_l}, \ l = 1, \ldots, d$, contains $2^{k-s}$ points of the set.

Denote by $v_j$ the $j$-th point of the Sobol' sequence. The Sobol' sequence is a $(t, s)$-sequence which means that each subset $v_{m2^k}, \ldots, v_{(m+1)2^k-1}$, where $m \geq 0$, of points of the sequence is $(s_1, \ldots, s_d)$-equidistributed in base 2 whenever $s \leq k - t$. $t$ is the $t - value$ of the sequence. As an illustration, consider the

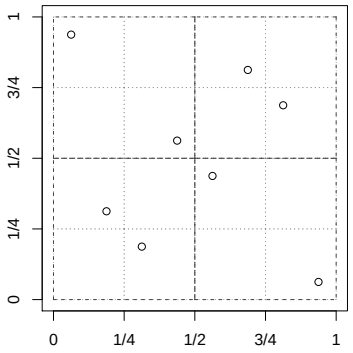



Figure 3: points $v_8$ to $v_{15}$ of the 2-dimensional Sobol' sequence, we have 2 points in each elementary interval $J(r)$.

subset of points $v_8$,$v_9$,. . . ,$v_{15}$ of the 2-dimensional Sobol' sequence represented

in Figure 3 . These points are obtained by fixing $m = 1$ and $k = 3$. This set of 8 points is $(1, 1)$-equidistributed in base 2, that is each elementary interval $J(r)$ contains 2 points of the set.

In the recursive replication procedure, at each step the design $D_l$ is augmented by adding a new set of points $D_{new,l+1}$. For the estimation of first-order Sobol' indices, $D_{new,l+1}$ possesses an LHS structure. As an alternative, a Sobol' sequence could be used. Being a $(t, s)$-sequence, the Sobol' sequence can be partitioned into subsets of $2^k$ points, $k \geq 1$, where each subset possesses $(s_1, \ldots, s_d)$-equidistribution properties. Thus, each subset can be seen as a new set of points $D_{new,l+1}$ augmenting the Sobol' sequence. Sobol' sequences are known to perform better than LHS for numerical integration. However, if used within the recursive replication procedure, the Sobol' sequence has to be replicated. The replication proposed in Section 2.2.1 does not ensure that the equidistribution properties of the subsets of the sequence are preserved.

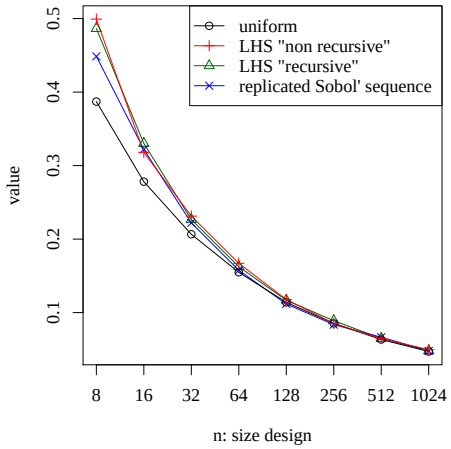
In the next section, we compare the space-filling properties of the nested design $D_l$ to those of a replicated Sobol' sequence. In addition, we compare the properties of $D_l$ to those of the replicated designs proposed by Tissot and Prieur [5]: LHS and randomized OA (equation (6)).

*4.2. Space-filling properties*

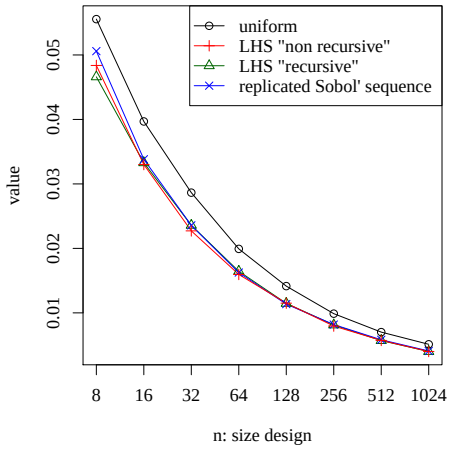
We propose to compare first the properties of the designs used for the estimation of first-order Sobol' indices: (i) uniform design, (ii) LHS "non recursive", (iii) LHS "recursive" and (iv) replicated Sobol' sequence. Design (ii) refers to the LHS used in [5]. Design (iii) refers to the design $D_l$ used in our recursive replication procedure. Prior to its replication, the Sobol' sequence is scrambled using the method proposed by Owen in [19]. The uniform design serves as a base of comparison.

Three criteria are selected to study the properties of these four designs: the mindist [20], the emst (euclidean minimal spanning tree [21]) and the $L^2$ star discrepancy [22]. The mindist criterion returns the minimum of the distances between all pair of points of a design. It can be interpreted as follows: the higher the value, the more regular the scaterring of design points. The emst criterion can be interpreted using a $(\mu, \sigma)$ graph, graph (c) of Figure 4, called interpretation graph. $\mu$ stands for the mean of the tree edges lengths, $\sigma$ for the standard deviation of the tree edges lengths.

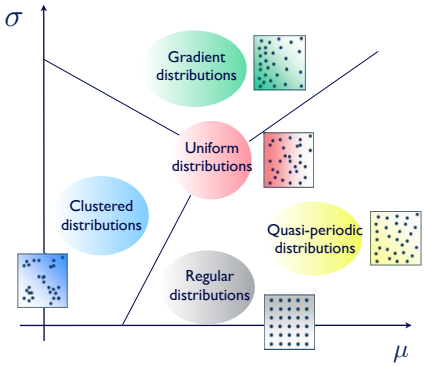
A value of the emst criterion is represented as a point in the $(\mu, \sigma)$ graph. The uniform distribution is used as a reference. A design having a higher value for $\mu$ and a smaller value for $\sigma$ than those of a uniform design is more regular. Mindist and emst criteria provide together a good estimation of the regularity properties of a design. The $L^2$ star discrepancy criterion measures the uniformity property of a design. The smaller the value, the more uniform is the design.



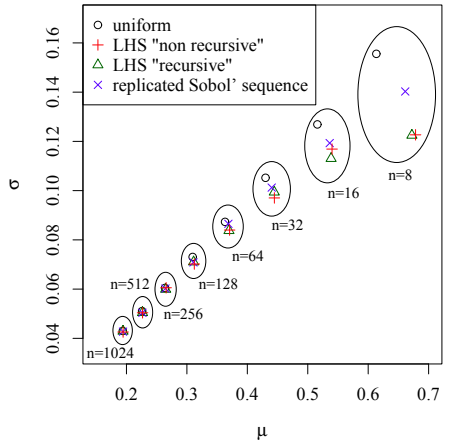

(a) mindist




(b) $L^2$ star discrepancy




(c) interpretation graph of emst




(d) emst

Figure 4: Averaged results of mindist, emst and star discrepancy criteria over 100 repetitions for different sizes $n$ of the designs used for the estimation of first-order indices.

Figure 4 shows the results obtained with each of the three criteria. The results are averaged over $r = 100$ repetitions. The inputs space dimension $d$ equals 5. The comparison is made for the following sizes $n$ of each design: $(2^3, 2^4, \ldots, 2^{10})$. For the LHS "recursive" these sizes correspond to those of design $D_l$ when iteratively adding a new set of points. For the replicated Sobol' sequence these sizes correspond to those of the design obtained when iteratively adding a new subset of points of the sequence. Both the "non recursive" and "recursive" LHS give similar results for the three criteria,. Furthermore, both designs give better results than the replicated Sobol' sequence for the first values of $n$.

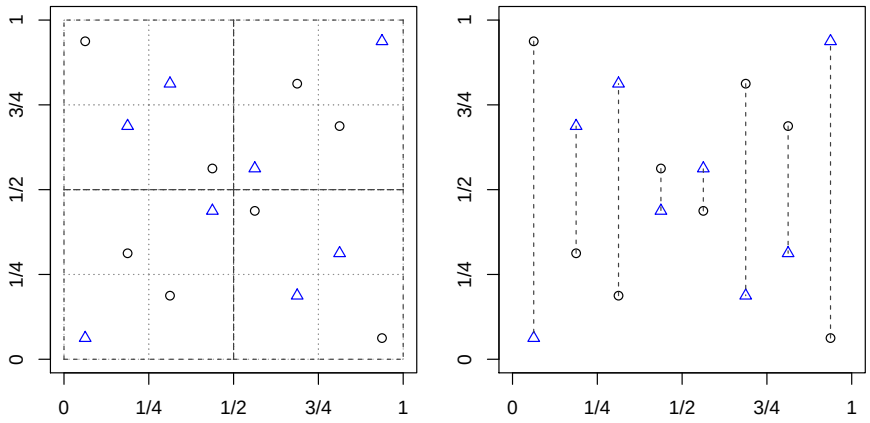
The main conclusion is that the "recursive" LHS possesses regularity and uniformity properties similar to those of the "non recursive" LHS. Hence, in terms of space-filling properties of the designs, there is no drawback to render the replication procedure recursive. The results for the replicated Sobol' sequence are not better than those of the "recursive" LHS, they are even slightly worse for small values of $n$. This justifies our choice of a "recursive" LHS over a replicated Sobol' sequence for the recursive replication procedure.

**Remark 2.** Most of the time, the replicated Sobol' sequence obtained using the procedure of Section 2.2.1 does not possess the same equidistribution properties of the ones of the original sequence. To overcome this problem, a proper procedure to replicate the Sobol' sequence can be designed. This new procedure satisfies the following conditions:

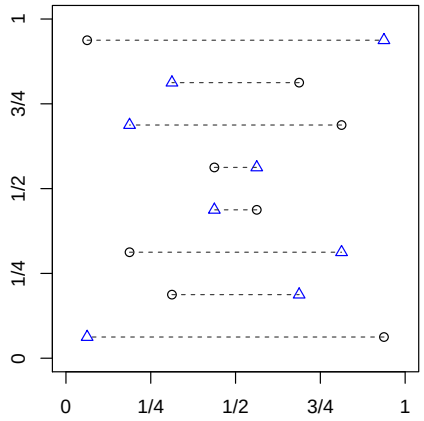
i) The Sobol' sequence and its replication have the same one-dimensional marginal values.
ii) The equidistribution properties of the original Sobol' sequence are preserved.

An illustration of the result this new approach is proposed in Figure 5. Graph (a) shows that both the Sobol' sequence and its replication are $(1, 1)$-equidistributed thus satisfying condition ii). Graphs (b) and (c) show that both the Sobol' sequence and its replication possess the same one-dimensional marginal values thus satisfying condition i). A generic algorithm for this new replication procedure of Sobol' sequence will be the concern of a futur work.

A second comparison is carried out between designs used for the estimation of closed-second order indices: (i) uniform design, (ii) OA "non-recursive", (iii) *accept-reject* and (iv) *algebraic*. Design (ii) refer to the randomized OA used in [5] and (i) is a uniform design. Designs (iii) and (iv)




(a) $(1,1)$-equidistribution properties. (b) First marginal values alignment.

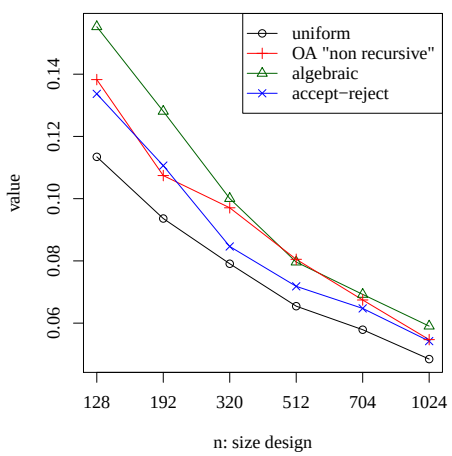



(c) Second marginal values alignment.

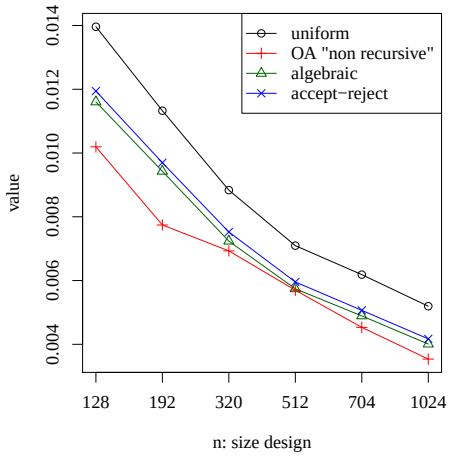
Figure 5: Sobol' sequence and its replication using the new approach. The points represent the Sobol' sequence. The triangles represent the replicated Sobol' sequence.

refers to the design $D_l$ constructed with either the *accept-reject* or the *algebraic* method. The same three previous criteria are used: mindist, emst and $L^2$ star discrepancy. The results are averaged over $r = 100$ repetitions. For the sake of visualization, results for the following sizes $n$ of the designs are represented: $(3 \times 8^2, 5 \times 8^2, 8 \times 8^2, 11 \times 8^2, 15 \times 8^2, 18 \times 8^2)$.

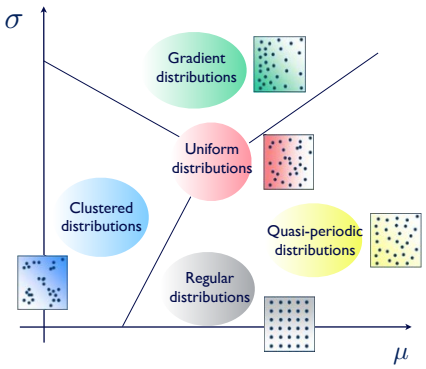
Figure 6 shows the results obtained with each of the three criteria. The *algebraic* design gives better results for both mindist and discrepancy crite-



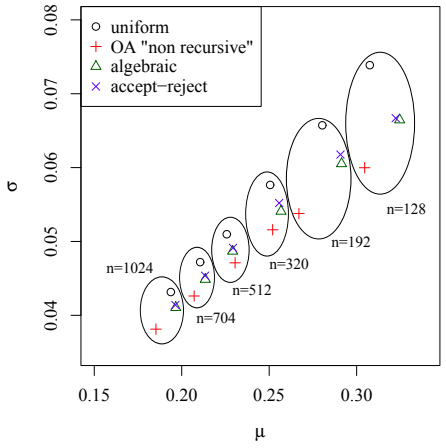

(a) mindist




(b) $L^2$ star discrepancy




(c) interpretation graph of emst




(d) emst

Figure 6: Averaged results of mindist, emst and star discrepancy criteria over 100 repetitions for different sizes $n$ of the designs used for the estimation of closed second-order indices.

ria than the *accept-reject* design. The OA "non-recursive" is the one giving the best results. For the emst criterion, the OA "non-recursive" performs the best. The *accept-reject* and the *algebraic* designs show similar results on this criterion. The main conclusion is that the *algebraic* design possesses regularity and uniformity properties overall slightly better than those of the *accept-reject* design. The OA "non-recursive" gives better results than those

two designs. This difference can be explained by the lack of progressive discretization of the inputs for both the *algebraic* and the *accept-reject* method. As a conclusion, the nested designs used in the recursive replication procedure possess slightly worse space-filling properties than the replicated designs proposed by Tissot and Prieur [5] for the estimation of closed second-order indices. However, these discrepancies are largely offset by the possibility to perform a recursive estimation of the indices.

In the next section, the recursive replication procedure is applied to a standard test function. This procedure is compared to the classical replication procedure of Tissot and Prieur [5].

*4.3. Application to test functions*

*Stopping criterion.* The recursive replication procedure is carried out until a stopping criterion is reached. At each step $l$ of the procedure, the following quantity is evaluated:

$$r_I{}^{(l)} = \left|\left|\widehat{\underline{S_I}}^{(l)} - \widehat{\underline{S_I}}^{(l-1)}\right|\right|,$$

where $||.||$ denotes the absolute value function. $r_I{}^{(l)}$ is an absolute difference between two successive estimations of $\underline{S_I}$. The stopping criterion we proposed is composed of two conditions $c_1$ and $c_2$. The first condition $c_1$ reads as follows:

$$\forall\ I \in J : r_I{}^{(l-l_0)} < \epsilon,\ r_I{}^{(l-l_0-1)} < \epsilon, \ldots, r_I{}^{(l)} < \epsilon$$

where $J$ equals either $\{1, \ldots, d\}$ or $\{(i,j) \in \{1, \ldots, d\}^2;\ i < j\}$ depending on whether first- or closed second-order Sobol' indices are estimated and $l_0 > 0$ is an integer. Condition $c_1$ tests if all quantities $r_I{}^{(l)}$ are smaller than a tolerance $\epsilon$ on $l_0$ consecutive steps. The second condition $c_2$ is defined as follows:

$$c_2 = l > L_{max}$$

where $L_{max}$ is a maximum number of iterations. The parameters $\epsilon$, $l_0$ and $L_{max}$ have to be properly set.

*Bratley et al. function.* In the following, the recursive replication procedure and the classical replication procedure are both applied to the test function introduced by Bratley *et al.* [23] and defined as follows:

$$f(X_1, \ldots, X_d) = \sum_{i=1}^{d} (-1)^i \prod_{k=1}^{i} X_k \ .$$

where $X_1, \ldots, X_d$ are independent random variables uniformly distributed on $[0, 1]$. Both first- and closed second-order Sobol' indices of the function are estimated with each method. Both methods are repeated $r = 100$ times to get a sample of estimated indices. We choose $d = 5$ for the input space dimension. Since $f$ has an analytical expression, theoretical values of the Sobol' indices can be precisely calculated through symbolic integrals evaluations.

The recursive procedure stops when one of the two conditions of the stopping criterion is satisfied. When the first condition $c_1$ is fulfilled, the recursive procedure stops at a step $K$. Thus, the $r$ repetitions of the procedure can be decomposed as a vector $(r_1, \ldots, r_K, \ldots, r_{L_{max}})$ where $r_K$ denotes the number of time the recursive procedure has stopped at the $K$-th step and $r_{L_{max}}$ is the maximum number of steps given by condition $c_2$. Denote by $r_\alpha$ the median of $(r_1, \ldots, r_{L_{max}})$ and $\alpha$ the corresponding step. To have a fair comparison, $\underline{S}_I$ is also estimated $r$ times with the classical replication procedure using a design whose size equals the one of the design used in the recursive replication procedure at the $\alpha$-th step.

For the estimation of first-order indices, we consider the context where a limited number of evaluations points is available as it is often the case in industrial applications. Therefore, a small value for $L_{max}$ is selected to highlight that the recursive replication procedure can perform as well as the classical one for a restricted budget of evaluation points. The parameters of the stopping criterion for the recursive procedure are set as follows: $\epsilon = 0.15$, $l_0 = 2$ and $L_{max} = 9$. The nested Latin Hypercube used to augment the designs of the recursive replication procedure is constructed to obtain the following sizes of the design $D_l$ at each step of the recursion: $(2^2, 2^3, 2^4, \ldots, 2^9)$.

Figure 7 shows barplots representation of the $r_K$ obtained. We observe that condition $c_2$ is only reached one third of the time.

Figure 8 shows the boxplots representation of the estimates for the two replication procedure: recursive (right boxplots) and classical (left boxplots). The true values of indices $\underline{S}_4$ and $\underline{S}_5$ are identical.

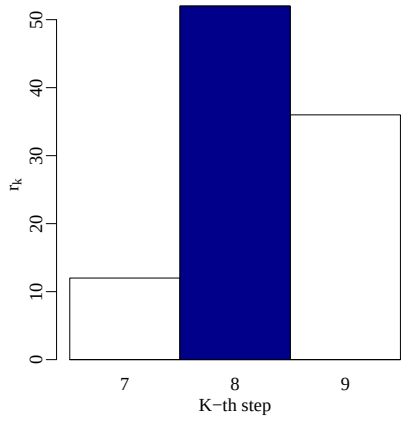



Figure 7: Distribution of the $r_K$ for the estimation of first-order indices. The bar associated to the step $\alpha$ is colored in black.

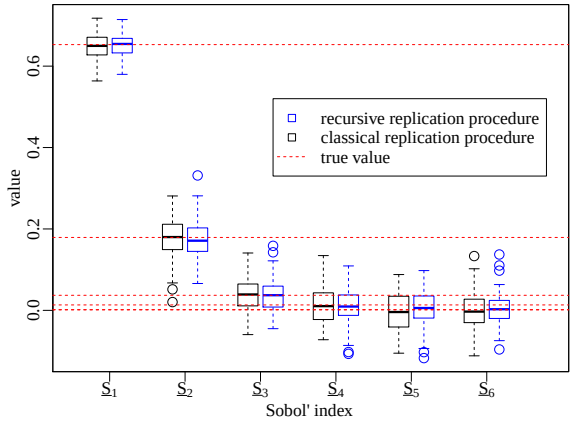


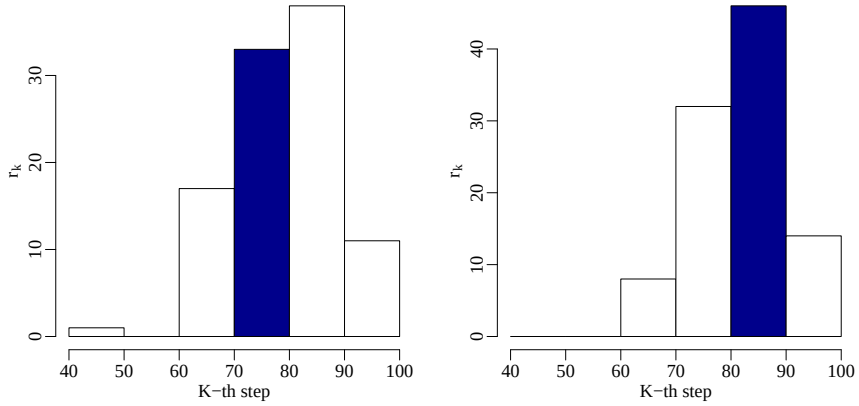

Figure 8: Boxplots of first-order Sobol' indices estimated $r = 100$ times with both the recursive and the classical replication procedures. For each index $\underline{S}_I$, the right boxplot refers to the recursive replication procedure, the left boxplot refers to the classical replication procedure. The dotted horizontal lines refer to the true values of the indices.

The two methods give overall similar results. Hence, there is no drawback to render the replication procedure recursive for the estimation of first-order indices. Furthermore, the recursive replication procedure shows that it is possible to decrease even more the number of simulations by adopting a sequential approach. One can calculate the gain in terms of number of evaluations of our sequential approach. This gain corresponds to the ratio of the

maximum number of evaluations $r_{L_{max}}$ divided by the iteration at which the recursive replication procedure stopped. For this example the median gain equals $9/8 = 1.125$ and the maximum gain equals $9/7 = 1.29$.

For the estimation of closed second-order Sobol' indices, the parameters of the stopping criterion for the recursive procedure are set as follows: $\epsilon = 3 \times 10^{-3}$, $l_0 = 3$ and $L_{max} = 100$. The Orthogonal Array of strength two $OA_\lambda(q, d, 2)$ used to augment the designs of the recursive replication procedure is constructed by setting $q = 8$ and $\lambda = 100$. Thus, the following sizes of $D_l$ are obtained: $(8^2, 2 \times 8^2, 3 \times 8^2, \ldots, 100 \times 8^2)$. Both the *accept-reject* method and the *algebraic* method are used to construct the Orthogonal Array. The estimations obtained with the recursive procedure using both constructions are compared to those obtained while using the classical replication procedure.

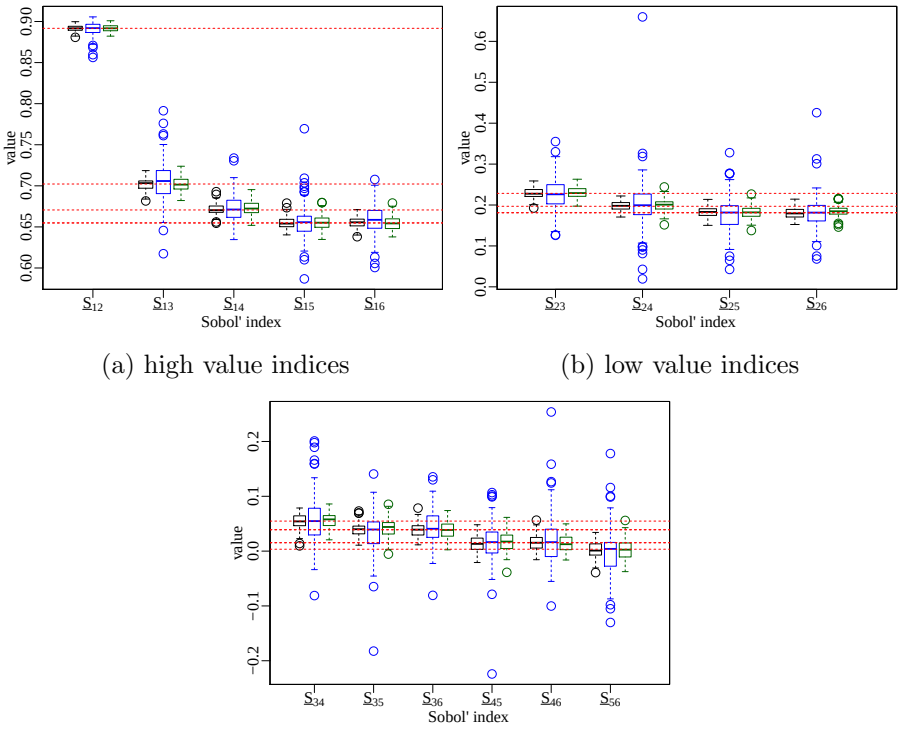
Figure 9 shows barplots representation of the $r_K$ obtained when applying the recursive replication procedure $r$ times using either the *algebraic* construction or the *accept-reject* construction. Looking at Figure 9, the recursive replication procedure finishes at earlier steps when using the *algebraic* construction.




(a) *algebraic* method (b) *accept-reject* method

Figure 9: Distribution of the $r_K$ when the recursive replication procedure is applied with either (a) the *algebraic* construction or (b) the *accept-reject* construction. For each graph, the bar associated to the median step $\alpha$ is colored in black.

Figure 10 gives the boxplots representation of the estimates obtained with: the recursive procedure using either the *algebraic* construction (middle boxplots) or the *accept-reject* construction (right boxplots) and the classical replication procedure (left boxplots). The main observation is that the




(a) high value indices

(b) low value indices

(c) near zero value indices

Figure 10: Boxplots of closed second-order Sobol' indices estimated $r = 100$ times with both recursive and classical replication procedures. For each index $\underline{S}_I$, the left boxplot refers to the classical replication procedure, the boxplot in the middle (resp. on the right) refers to the recursive replication procedure using the *algebraic* (resp. *accept-reject*) construction. The horizontal dotted lines refer to the true values of the indices.

recursive replication procedure using the *algebraic* construction shows more variability in the estimates than the two others. This observation is emphasized for graphs (b) and (c) of Figure 10 corresponding to Sobol' indices with low values. However, this variability observed is mostly due to the *algebraic* construction itself stopping at earlier steps than the *accept-reject* construction. The results obtained with the recursive procedure using the *accept-reject* construction are overall similar to those obtained with the clas-

sical replication procedure.

As for the case of first-order indices, one can calculate the gain of our recursive approach in terms of number of evaluations. Table 1 gives the gain of our method for each quartile of the vector $(r_1, \ldots, r_{L_{max}})$. The recursive

| quartile | construction | value | gain= $\frac{r_{L_{max}}}{\text{value}}$ |
|---|---|---|---|
| $r_{1/4}$ | *algebraic* | 73 | 1.37 |
| | *accept-reject* | 76 | 1.32 |
| $r_{1/2}$ | *algebraic* | 80 | 1.25 |
| | *accept-reject* | 82 | 1.22 |
| $r_{3/4}$ | *algebraic* | 87 | 1.15 |
| | *accept-reject* | 88 | 1.14 |

Table 1: Gain of the recursive replication procedure using either the *algebraic* or the *accept-reject* construction. The gain is calculated in terms of number of evaluations for each quartile $(r_{1/4}, r_{1/2}, r_{3/4})$ of the vector $(r_1, \ldots, r_{L_{max}})$.

replication procedure shows that it is possible to decrease even more the number of simulations by adopting a sequential approach for the estimation of closed second-order indices while conserving roughly the same precision. However, as stated before, there is a computational price to pay induced by the *accept-reject* construction. When the input space dimension is small ($d \leq 4$), it is harder to find new blocks, thus the *algebraic* construction should be preferred to the *accept-reject* one. At the opposite, when the input space dimension is high, new blocks are easier to find, thus the *accept-reject* construction should be used as it gives more accurate results.

## Conclusion

In this paper we proposed a new approach rendering the replication procedure recursive to estimate first-order or closed second-order Sobol' indices. We introduced a recursive formula for the Sobol' index estimator. The recursive procedure presented consists in augmenting the two replicated designs with new sets of points. Through the construction of nested space-filling structured designs a randomization of these sets of points was performed (Step 3.2 of Algorithm 1). For the case of closed second-order indices, two

methods were proposed to construct the nested space-filling structured design: an *algebraic* method and an *accept-reject* method. Our recursive replication procedure was compared to the classical replication procedure of Tissot and Prieur [5]. The comparison focused on the space-filling properties of the designs and on the precision of the Sobol' indices estimates.

The replicated designs proposed in [5] are known to be highly efficient in terms of number of simulations. Yet the results in this paper showed that it is still possible to decrease the number of simulations by adopting a sequential procedure based on a recursive method of estimation. More precisely, the nested designs proposed here gave roughly the same precision on sensitivity indices as the replicated designs used in [5] in $N$ simulations. But with a random number of simulations bounded by $N$ and of a much smaller expectation. Furthermore, the space-filling properties of the nested designs constructed were on average as good as the one of the replicated designs used in [5].

For the case of first-order indices, the nested design used could be improved by considering Sobol' sequences replicated using an approach that preserve their equidistribution properties (see Remark 2). This will the object of a future work. For the case of closed second-order indices, the variability in the results showed by the recursive replication procedure while using the *algebraic* construction could be reduced by further working on the set $C$ (Section 3.2 Variant 2). In our case, the set $C$ was filled with elements $g$ chosen at random. A more deterministic choice of the $g$ could lead to a better exploration of the input space.

## Acknowledgments

This work is supported by the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020).

## References

[1] I. M. Sobol', Sensitivity indices for nonlinear mathematical models, Mathematical Modeling and Computational Experiment 1 (1993) 407–414.

[2] A. Saltelli, Making best use of model evaluations to compute sensitivity indices, Computer Physics Communications 145 (2) (2002) 280–297, doi: 10.1016/S0010-4655(02)00280-1.

[3] M. D. McKay, W. J. Conover, R. J. Beckman, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245, doi: 10.2307/1271432.

[4] T. A. Mara, O. R. Joseph, Comparison of some efficient methods to evaluate the main effect of computer model factors, Journal of Statistical Computation and Simulation 78 (2) (2008) 167–178, doi: 10.1080/10629360600964454.

[5] J. Y. Tissot, C. Prieur, A randomized orthogonal array-based procedure for the estimation of first- and second-order Sobol' indices, J. Statist. Comput. Simulation 85 (2014) 1358–1381, doi: 10.1080/00949655.2014.971799.

[6] C. Lemieux, Monte Carlo and quasi-Monte Carlo sampling, Springer Ser. Statist., new york: springer edn., 2009.

[7] A. B. Owen, Orthogonal arrays for computer experiments, integration and visualization, Statist. Sinica 2 (1992) 280–297.

[8] P. Z. G. Qian, Nested Latin hypercube designs, Biometrika 96 (4) (2009) 957–970, doi:10.1093/biomet/asp045.

[9] P. Z. G. Qian, B. Tang, C. F. J. Wu, Nested space-filling designs for computer experiments with two levels of accuracy, Stat. Sinica 19 (2009) 287–300.

[10] P. Z. G. Qian, M. Ai, C. F. J. Wu, Construction of nested space-filling designs, Ann. Stat. 37 (6A) (2009) 3616–3643, doi:10.1214/09-AOS690.

[11] A. Dey, On the construction of nested orthogonal arrays, Australas. J. Combin. 54 (2012) 37–48.

[12] W. Hoeffding, A class of statistics with asymptotically normal distributions, Annals of Mathematical Statistics 19 (3) (1948) 293–325.

[13] A. Janon, T. Klein, A. Lagnoux, M. Nodet, C. Prieur, Asymptotic normality and efficiency of two Sobol' index estimators, ESAIM Probab. Stat. 18 (2014) 342–364, doi:10.1051/ps/2013040.

[14] H. Monod, C. Naud, D. Makowski, Uncertainty and sensitivity analysis for crop models, chap. 3, Elsevier, 55–100, 2006.

[15] K. Kishen, On latin and hyper-graeco cubes and hypercubes, Current Science 11 (3) (1942) 98–99.

[16] C. R. Rao, Hypercubes of strength "d" leading to confounded designs in factorial experiments, Bulletin of the Calcutta Mathematical Society 38 (3) (1946) 67–78.

[17] A. S. Hedayat, N. J. A. Sloane, J. Stufken, Orthogonal Arrays: Theory and Applications, Springer Series in Statist., new york: springer edn., 1999.

[18] D. R. Stinson, J. L. Massey, An Infinite Class of Counterexamples to a Conjecture Concerning Nonlinear Resilient Functions, J. Cryptology 8 (1995) 67–173, doi:10.1007/BF00202271.

[19] A. B. Owen, Randomly permuted (t;m;s)-nets and (t;s) sequences, in: H. Niederreiter, P. J. S. Shiue (Eds.), Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer-Verlag, New York, 1995.

[20] M. E. Jonshon, L. M. Moore, D. Ylvisaker, Minmax and maximin distance designs, J. Statist. Plann. Inference 26 (2) (1990) 131–148, doi: 10.1016/0378-3758(90)90122-B.

[21] J. Franco, O. Vasseur, N. Corre, M. Sergent, Minimum spanning tree: A new approach to assess the quality of the design of computer experiments, Chemom. Intell. Lab. Syst. 97 (2) (2009) 164–169, doi: 10.1016/j.chemolab.2009.03.011.

[22] W. J. Morokoff, R. E. Caflisch, Quasi-random sequences and their discrepancies, SIAM J. Sci. Comput. 15 (16) (1994) 12511279.

[23] P. Bratley, B. L. Niederreiter, Implementation and tests of low-discrepancy sequences, ACM Trans. Model. Comput. Simul. 2 (3) (1992) 195–213, doi:10.1145/146382.146385.

**Acronyms and Symbols**

| | |
|---|---|
| $\subsetneq$ | (strict) inclusion symbol |
| $\subset$ | inclusion symbol |
| $\|x\|$ | cardinality of a set $x$ |
| $x^T$ | transpose of $x$ |
| $\Pi_n$ | set of all the permutations on $\{1, \ldots, n\}$ |
| $\mathcal{LH}(n, d)$ | set of all $n \times d$ discrete Latin Hypercubes |
| $OA_\lambda(q, d, t)$ | Orthogonal Array of index $\lambda$, levels $q$ and strength $t$ |
| $GF(q)$ | Galois field of order $q$ |
| $\Diamond$ | operator symbol |
| $F$ | cumulative distribution function |
| $F^{-1}$ | quantile function |
| $\|\|.\|\|$ | absolute value function |
| $\underline{S_I}$ | closed Sobol' index of order $I$ |
| $\widehat{\underline{S_I}}$ | estimator of $\underline{S_I}$ |
| $d$ | inputs space dimension |
| $\mathbb{R}$ | real coordinate space |
| $(\Omega, \mathcal{A}, \mathbb{P})$ | probability space |
| $P_X$ | distribution function of a random variable $X$ |
| $\mathbb{L}^2(P_X)$ | space of square integrable functions |
| E | expectation symbol |
| Var | variance symbol |
| Cov | covariance symbol |
| $\mathbb{N}$ | set of positive integers |