

Optimized HSIC-based tests for sensitivity analysis: application to thermal-hydraulic simulation of accidental scenario on nuclear reactor

Reda El Amri, Amandine Marrel

▶ **To cite this version:**

Reda El Amri, Amandine Marrel. Optimized HSIC-based tests for sensitivity analysis: application to thermal-hydraulic simulation of accidental scenario on nuclear reactor. 2021. cea-03193912

HAL Id: cea-03193912
https://cea.hal.science/cea-03193912

Preprint submitted on 9 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimized HSIC-based tests for sensitivity analysis: application to thermal-hydraulic simulation of accidental scenario on nuclear reactor

Reda El Amri[a,*], Amandine Marrel[a]

[a] *CEA, DES, DER, F-13108 Saint-Paul-Lez-Durance, France*

Abstract

Physical phenomena are commonly modeled by numerical simulators. Such codes can take as input a high number of uncertain parameters and it is important to identify their influences on the outputs via a Global Sensitivity Analysis (GSA). However, these codes can be time consuming which prevents a GSA based on the classical Sobol' indices, requiring too many code simulations. This is all the more true as the number of inputs is important. To address this limitation, we consider recent advances in dependence measures, focusing on the Hilbert-Schmidt independence criterion (HSIC). In this framework, this paper proposes new goal-oriented algorithms to optimize the permuted HSIC-based tests for screening and ranking purposes.

HSIC-based tests are built upon a sample of inputs/output of the studied model (simulator) and relies on the estimation of a *p-value* under independence hypothesis. These p-values can be estimated either by an asymptotic approximation (for large sample size) or by permutation method. However in the latter case, a brute approach with a large number of permutations can be prohibitive in practice, especially when the testing procedure is repeated a large number of times. To overcome this, we propose several strategies to greedy estimate the p-value, according to the final goal of GSA. Three sequential permuted tests are thus proposed: screening-oriented, ranking-oriented and ranking-screening-oriented. These algorithms are tested and compared on analytical examples. The performances in terms of accuracy, efficiency and time saving are clearly demonstrated. Their use is then illustrated on a nuclear engineering use case simulating an intermediate-break loss-of-coolant accident on a pressurized water reactor. A convergence study, made computationally tractable by the optimized algorithms, is carried out to assess the convergence of the results, according to the sample size.

Keywords: Sensitiviy analysis, Hilbert-Schmidt Independence Criterion (HSIC), Independence tests, Sequential permutations.

*Corresponding author

Email address: mohamed-reda.elamri@cea.fr (Reda El Amri)

1. Introduction

As part of safety studies for nuclear reactors, computation codes (or numerical simulators) are fundamental tools for understanding, modelling and predicting physical phenomena. These tools can take a large number of input parameters, characterizing the studied phenomenon or related to its physical and numerical modelling. The information related to some of these parameters is often limited or uncertain, this can be due to the lack or absence of data, measurement or modelling errors, or a natural variability of the parameters. These input parameters, and consequently the simulator output, are thus uncertain. This is referred to as uncertainty propagation. It is therefore important not only to consider the nominal values of the inputs, but also to take into account their uncertainties and their effects on the output. Alongside uncertainty propagation, a sensitivity analysis can be conducted. The Sensitivity Analysis (denoted SA) aims at determining how the variability of the input parameters affects the value of the output or the quantity of interest [1]. It thus allows to identify and perhaps quantify, for each input parameter or group of parameters, its contribution to the variability of the output. In a simplified way, SA can have two main goals:

- to prioritize input parameters by order of influence on the output variability, this is referred to as *ranking*;
- to separate the inputs into two groups: those which mostly influence the output uncertainty and those whose influence can be neglected. This input splitting is known as *screening*.

Whatever its purpose, SA results provide valuable information for the impact of uncertain inputs, the comprehension of the model and the underlying physical phenomenon. They can also be used for various purposes: reducing uncertainties by targeting the characterization efforts on the most influential inputs, simplifying the model by setting non-influential inputs to reference values, or validating the model with respect to the modeled phenomenon. These issues explain the amount of recent studies on statistical tools and methods for sensitivity analysis. One of the most commonly used SA methods in industrial applications is based on a decomposition of the output variance [2, 3]: each term of the decomposition represents the contribution share of an input or a group of inputs to the output variance. As a result of this approach, Sobol's indices are obtained. However, these easy-to-interpret indices have several practical drawbacks: a very costly estimate in terms of the number of code simulations (several tens to hundreds of thousands) and partial information provided by the variance towards to the notion of probabilistic dependence. To overcome these limitations, other approaches based on dependence measures have recently been proposed by [4] for GSA, then deeply studied in [5, 6] and just recently by

[7]. The interest in these methods is explained by their theoretical and practical advantages, which are described below, and the promising results obtained in several industrial applications. The work presented in this article focus on the Hilbert-Schmidt Independence Criterion, denoted HSIC [8], which generalizes the notion of covariance between two random variables and thus makes it possible to capture a very wide spectrum of forms of dependence between the variables. Moreover, as illustrated by [5], HSIC indices also have the advantage of having a low estimation cost (in practice a few hundred code simulations). HSIC-based statistical independence tests can also be built to determine in a robust and objective way if an input is significantly influential on the output. Thus, depending on the study case and the objectives, HSIC measures can be used either to screen or rank inputs by order of influence on the output.

Asymptotic version [9] but also non-asymptotic one [5, 10] of HSIC-based tests exist. The non-asymptotic extension based on permutation method makes it possible to deal with simulation samples of smaller size (a few tens to hundreds). However, some practical limitations remain in the use of permutation-based HSIC tests. More precisely, a large number of permutations (several thousands) is recommended to ensure the convergence of the estimated p-value of each independence test (test of dependence between each input and the output). The resulting computation cost can therefore become significant when:

- the number of inputs is large (one or more hundreds);
- the SA procedure is repeated a large number of times. For example, this happens if the input laws are uncertain and their uncertainty is taken into account in a second-level SA, as proposed by [11]. SA procedure can also be repeated according to a variable parameter: for instance, if we consider target or conditional[1] SA [12, 7]. In this latter case, we may wish to vary the threshold value;
- the HSIC-test is included in an additional outer loop of bootstrap sampling, in order to take into account the sampling variance and control the convergence according to the number of simulations (sample size).

In all these cases, a brute approach with a large number of permutations can therefore be prohibitive, especially when the sample size is intermediate (and not enough large to guarantee the validity of the asymptotic framework and use asymptotic HSIC tests). To overcome this limitation, the work presented in this paper aims at developing an efficient methodology and associated operational tool in order to optimize the number of permutations in HSIC-based tests, with regard to the final objective of GSA. The resulting benefit on numerical examples and industrial applications is also illustrated and quantified.

[1] Target and conditional SA focus on a restricted domain of the studied phenomenon, most often defined by exceeding a threshold value. In this framework, target SA aims at measuring the influence of the inputs on the occurrence of the threshold being exceeded, while conditional SA evaluates the influence of the inputs on the output within this critical domain only, ignoring what happens outside.

The paper is organized as follows. In Section 2, we detail the HSIC and associated SA measures, we also describe the HSIC-based independence tests. Section 3 introduces the proposed methodology to optimize the number of permutations and sequentially well estimate the p-values of HSIC-based independence tests. The new proposed algorithms are then tested on analytical examples in Section 4. Finally, in Section 5, a further application is proposed, on a use case simulating an accidental scenario on a nuclear pressurized water reactor.

2. HSIC: General principle and use for independence test

First we introduce a few notations. Throughout the rest of this paper, the numerical model is represented by the relation:

$$\begin{aligned} \mathcal{M} : \mathcal{X} &\longrightarrow \mathcal{Y} \\ \mathbf{X} &\longmapsto \mathcal{M}(\mathbf{X}) = Y, \end{aligned}$$

where $\mathbf{X} = (X_1, \ldots, X_d)^\top$ and Y are respectively the d uncertain inputs and the uncertain output[2], evolving in measurable spaces respectively denoted $\mathcal{X} = \times_{i=1}^{d} \mathcal{X}_i$ where $\mathcal{X}_i \subset \mathbb{R}$ and $\mathcal{Y} \subset \mathbb{R}$. As part of the probabilistic approach, the d inputs are assumed to be continuous and independent random variables with known densities. These densities are respectively denoted $p_{X_1}, \ldots, p_{X_d}$. Finally, $p_{\mathbf{X}} = \prod_{i=1}^{d} p_{X_i}$ denotes the density of the random vector $\mathbf{X}$. As the model $\mathcal{M}$ is not known analytically, a direct computation of the output probability density as well as dependence measures between $\mathbf{X}$ and Y is impossible. Only observations (or realizations) of $\mathcal{M}$ are available. It is therefore assumed in the following that we have a n sample of inputs and associated outputs $(\mathbf{X}^{(m)}, Y^{(m)})_{1 \leq m \leq n}$ where $Y^{(m)} = \mathcal{M}(\mathbf{X}^{(m)})$ for $m = 1, \ldots, n$.

2.1. HSIC definition and estimation

As mentioned in Section 1, the HSIC measure between an input X_i and the output Y is based on a generalization of the notion of covariance between these random variables. For this, we associate to X_i a universal Reproducing Kernel Hilbert-Schmidt space (RKHS) $\mathcal{F}_i$ composed of functions mapping from $\mathcal{X}_i$ to $\mathbb{R}$ and defined by the *characteristic kernel* function k_i (see, e.g. [13] for a complete bibliography on RKHS spaces). The same transformation is associated to Y, considering the universal RKHS $\mathcal{G}$ and the kernel function k. To do so, we define a (possibly nonlinear) mapping $\Phi_i : \mathcal{X}_i \to \mathcal{F}_i$ from each $\mathcal{X}_i$ to a feature space $\mathcal{F}_i$ (and an analogous map $\psi : \mathcal{Y} \to \mathcal{G}$). Each feature map is composed of a set of feature functions $\Phi_i(x) = (\phi_{i,j}(x))_j$ and can be of finite or infinite dimension. And the most important, the feature map is linked to the positive definite kernel function via the following relationship:

$$k_i(x, x') = \langle \Phi_i(x), \Phi_i(x') \rangle_{\mathcal{F}_i} \quad \text{and} \quad k(y, y') = \langle \psi(y), \psi(y') \rangle_{\mathcal{G}}, \tag{1}$$

[2] or any quantity of interest derived from the output(s).

where $\langle \cdot, \cdot \rangle_{\mathcal{F}_i}$ and $\langle \cdot, \cdot \rangle_{\mathcal{G}}$ denote the inner scalar product over $\mathcal{F}_i$ and $\mathcal{G}$ respectively. The kernels k_i and k are associated uniquely with respective reproducing kernel Hilbert spaces $\mathcal{F}_i$ and $\mathcal{G}$. We may now define a cross-covariance between the feature maps of X_i and Y, associated to the probability density function p_{X_iY} of (X_i, Y):

$$\mathbb{COV}(\Phi_i(X_i), \psi(Y)) = \mathbb{E}_{X_iY}[\Phi_i(X_i) \otimes \psi(Y)] - \mathbb{E}_{X_i}[\Phi_i(X_i)] \otimes \mathbb{E}_Y[\psi(Y)]$$

where $\otimes$ denotes the tensor product. Then, the crossed-covariance operator $C_{X_iY}[.]$ generalizes the notion of covariance between X_i and Y. Finally, the HSIC is defined as the Hilbert-Schmidt norm of the operator C_{X_iY} (as demonstrated in [14]):

$$||C_{X_iY}||_{HS} = \sum_{l,m} |\langle\ u_l, C_{X_iY}[v_m]\rangle_{\mathcal{F}_i}|^2,$$

where $(u_l)_{l\geq 0}$ and $(v_m)_{m\geq 0}$ are orthonormal bases of, respectively, $\mathcal{F}_i$ and $\mathcal{G}$. Thanks to the kernel trick, [8] show that the HSIC measure can be written, in an equivalent manner, using expected values of kernels:

$$\begin{aligned}\mathrm{HSIC}_{\mathcal{F}_i,\mathcal{G}}(X_i, Y) =& \mathbb{E}[k_i(X_i, X_i')k(Y, Y')] + \mathbb{E}[k_i(X_i, X_i')]\mathbb{E}[k(Y, Y')] \\ &- 2\mathbb{E}[\mathbb{E}[k_i(X_i, X_i')|X_i]\mathbb{E}[k(Y, Y')|Y]],\end{aligned}$$

where (X_i', Y') is an independent and identically distributed (iid) copy of (X_i, Y). Note that the dependence on RKHS $\mathcal{F}_i, \mathcal{G}$ is often omitted, as in the following, to adopt the simplified notation $\mathrm{HSIC}(X_i, Y)$. We note that the nullity of this measure is not always equivalent to the independence between X_i and Y. This propriety depends on the RKHS associated to X_i and Y. In particular, it has been proven that when the kernels k_i and k belong to the specific class of characteristic kernels then,

$$\mathrm{HSIC}(X_i, Y) = 0 \iff X_i \perp\!\!\!\perp Y, \tag{2}$$

$\perp\!\!\!\perp$ meaning that X_i and Y are independent (see, e.g. [15, 16] for a detailed overview on the subject).

For GSA purposes, the nullity of the measure indicates that X_i does not influence Y. This property is crucial for testing dependence, as it will be described in Section 2.2. A most commonly used characteristic kernel for real variable is the Gaussian kernel which is defined for a pair of variables $(\mathbf{z}, \mathbf{z}') \in \mathbb{R}^q \times \mathbb{R}^q$ by:

$$\ell_\lambda(\mathbf{z}, \mathbf{z}') = \exp(-\lambda||\mathbf{z} - \mathbf{z}'||_2^2),$$

where λ is the *bandwidth parameter* of the kernel ℓ_λ and $||.||_2$ is the Euclidean norm in $\mathbb{R}^q$. Usually, one uses in practice $\lambda = 1/\sigma^2$ with σ^2 being the empirical variance of the sample of the considered variable Z.

To conclude on HSIC, it remains to deal with its estimation in practice. From a n-sample $(X_i^{(m)}, Y^{(m)})_{1\leq m\leq n}$ of (X_i, Y), The V-estimator of the measure HSIC is proposed in [8]:

$$\widehat{\mathrm{HSIC}}(X_i, Y) = \frac{1}{n^2}\mathrm{Tr}(L_i H L H), \tag{3}$$

where the Gram matrices L_i and L are defined by $L_i = (k_i(X_i^{(l)}, X_i^{(m)}))_{1\leq l,m\leq n}$ and $L = (k(Y^{(l)}, Y^{(m)}))_{1\leq l,m\leq n}$, and $H = (\delta_{lm} - 1/n)_{1\leq l,m\leq n}$, with δ_{lm} is the Kronecker operator.

Remark 1. *Several methods based on the use of HSIC measures are developed for GSA. For ranking purpose, [4] defines normalized sensitivity indices. These indices classify the inputs $X_1, \ldots, X_d$ by order of influence on the output Y. They are defined for all $i \in \{1, \ldots, d\}$ by:*

$$R^2_{HSIC,i} = \frac{\mathrm{HSIC}(X_i, Y)}{\sqrt{\mathrm{HSIC}(X_i, X_i)}\sqrt{\mathrm{HSIC}(Y, Y)}}. \tag{4}$$

Thanks to the property given by Eq. (2) with characteristics kernels, independence tests can be built upon HSIC and offer a mathematical rigorous framework for GSA screening, as described in the next section.

2.2. Statistical independence tests based on HSIC: asymptotic and non-asymptotic versions

In a screening context, the objective is to separate the input parameters into two sub-groups, the significant ones and the non-significant ones. For this, HSIC can be used to conduct a statistical hypothesis test. For a given input X_i, it aims at testing the null hypothesis "$(\mathcal{H}_0^i)$: X_i and Y are independent", against, its alternative "$(\mathcal{H}_1^i)$: X_i and Y are dependent". Since the nullity of HSIC (with characteristic kernels) is equivalent to the independence between X and Y, testing independence is equivalent to test:

$$(\mathcal{H}_0^i) : \mathrm{HSIC}(X_i, Y) = 0 \qquad \text{versus} \qquad (\mathcal{H}_1^i) : \mathrm{HSIC}(X_i, Y) > 0.$$

The statistic estimator $\widehat{\mathcal{T}}_i = n \times \widehat{\mathrm{HSIC}}(X_i, Y)$ is then a natural choice to test independence between X_i and Y. The probability of being wrong under the null hypothesis $(\mathcal{H}_0^i)$ is generally called *first-kind error* or *level of test*[3] and denoted α. Theoretical and practical control of the level of independence tests is possible and generally set at a threshold of 5% or 10%. By contrast, there is currently no theoretical or practical control of the second-kind error.

The test can be defined in an equivalent way using the p-value which is defined as the probability that, under $(\mathcal{H}_0^i)$, the test statistic (in this case, $\widehat{\mathcal{T}}_i$) is greater than or equal to the observed value on the data $\widehat{\mathcal{T}}_{i,obs} = n \times \widehat{\mathrm{HSIC}}(X_i, Y)_{obs}$. Therefore, the p-value is defined by:

$$p_{val,i} = \mathbb{P}_{(\mathcal{H}_0^i)}(\widehat{\mathcal{T}}_i \geq \widehat{\mathcal{T}}_{i,obs}).$$

Finally, the null hypothesis $\mathcal{H}_0^i$ is rejected if $p_{val,i} < \alpha$, which means here that the input X_i is significantly influential.

[3] Rigorously, the *level* of the test is an upper bound of the first-kind error.

To carry out the test, it remains to compute $p_{val,i}$ under $\mathcal{H}_0^i$. Unfortunately, the law of the statistic $\widehat{\mathcal{T}}_i$ under $\mathcal{H}_0^i$ is not theoretically known and its estimation depends on the framework in which we are placed. Under asymptotic convergence (i.e., when n is sufficiently large), [9] have proved that the asymptotic law of the $\widehat{\mathcal{T}}_i$ estimator can be approached by a Gamma distribution.

Outside the asymptotic framework, i.e., when n is rather small (e.g., lower than thousand), the Gamma approximation cannot be used anymore. In this case, non-asymptotic versions of the tests [5, 10] based on permutation-method offers a suitable and relevant alternative. For this, B independent and uniformly distributed random permutations $\mathbf{Y}_{[1]}, \ldots, \mathbf{Y}_{[B]}$ of the output sample $\mathbf{Y} = \{Y^{(1)}, \ldots, Y^{(n)}\}$ are generated. The initial input sample $\mathbf{X}_i$ where $\mathbf{X}_i = \{X_i^{(1)}, \ldots, X_i^{(n)}\}$ is associated to each $\mathbf{Y}_{[b]}$ with $b \in \{1, \ldots, B\}$:

$$\begin{bmatrix} X_i^{(1)} & Y^{(1)} \\ \vdots & \vdots \\ X_i^{(n)} & Y^{(n)} \end{bmatrix} \rightarrow \left(\begin{bmatrix} X_i^{(1)} & Y_{[b]}^{(1)} \\ \vdots & \vdots \\ X_i^{(n)} & Y_{[b]}^{(n)} \end{bmatrix} \right).$$

By doing so, B iid n-size samples $((\mathbf{X}_i, \mathbf{Y}_{[b]})_{1 \leq b \leq B})$ of (X_i, Y) under the independence hypothesis $(\mathcal{H}_0^i)$ are simulated. Then, the HSIC is computed for each n-size sample $(\mathbf{X}_i, \mathbf{Y}_{[b]})$. A sample of iid realisations of the HSIC under $(\mathcal{H}_0^i)$ is obtained and the p-value can be estimated by the empirical estimator (Monte-Carlo approximation). The resulting permutation-based test is summarized by Algorithm 1. More details and demonstration of test properties are available in [10]. This permuted test with Monte Carlo approximation of p-value is demonstrated to be of prescribed non-asymptotic level α.

Algorithm 1 – Permutation-based independence test (for each X_i)

Require: The learning sample $(\mathbf{X}_i, \mathbf{Y})$ of n inputs/outputs $\{(X_i^{(1)}, Y^{(1)}), \ldots, (X_i^{(n)}, Y^{(n)})\}$, B and α

1: Compute $\widehat{\text{HSIC}}_{obs}(X_i, Y)$ from Eq. (3)
2: Generate B permutation-based samples $\left(\mathbf{X}_i, \mathbf{Y}_{[b]}\right)_{1 \leq b \leq B}$
3: Compute the B permutation-based estimators $\left(\widehat{\text{HSIC}}_b(X_i, Y)\right)_{1 \leq b \leq B}$ by replacing $\mathbf{Y}$ by $\mathbf{Y}_{[b]}$ in Eq. (3)
4: Estimate the p-value by Monte-Carlo estimator:

$$\hat{p}_{val,i}^B = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\widehat{\text{HSIC}}_b(X_i,Y) > \widehat{\text{HSIC}}_{obs}(X_i,Y)}$$

5: **if** $\hat{p}_{val,i}^B < \alpha$ **then**
6: **return** reject $(\mathcal{H}_0^i)$
7: **else**
8: **return** accept $(\mathcal{H}_0^i)$
9: **end if**

Therefore, HSIC can be used, through the associated independence test, for screening purpose. By controlling the level of the test α, the test provides a more robust and objective interpretation of HSIC values in order to conclude on the significant influence of an input. Moreover, beyond allowing a decision for screening purpose, the estimated p-value can also be used for ranking purpose. The p-value can be viewed as an indicator of the likelihood of the observed statistics under $(\mathcal{H}_0^i)$. The lower the p-value, the stronger $(\mathcal{H}_0^i)$ is rejected and the higher the influence of the input. Consequently, under the assumption that the same RKHS is used for all the inputs, HSIC-test p-values can be compared: the inputs can be ordered according to the estimated p-value. This approach has notably been used by [12] for a preliminary screening and ranking before building a metamodel.

As discussed in the previous subsection, the p-value of HSIC statistics can be estimated either by permutation or, in an asymptotic framework, by the approximation of the HSIC distribution with a Gamma law. In practice, it is often difficult to know if we are in the asymptotic framework and if the law convergence is ensured. These hypotheses are perfectly acceptable when the sample size n is higher than several thousands but it is really more questionable when n is equal to few hundred or one or two thousands. In this latter case, it seems more reasonable to apply permutation-based tests. However, the estimation of the p-value mainly depends on the number of permutations B. Even if the estimator is unbiased, its variance (and therefore its mean square error) depends on B. One solution can consist in considering systematically a very large value of B, e.g. $B = 5000$, to ensure convergence of the estimated p-value of each independence test between each input and the output. Since the computational complexity of each HSIC estimation is $\mathcal{O}(n^2)$, the total resulting computation cost for d inputs is $\mathcal{O}(dBn^2)$ and can therefore become significant when the number of inputs d is large. All the more so if the procedure is repeated a large number of times, as in the situations listed in Section 1 (large number of inputs, several SA, call in a bootstrap outer loop, etc.). A brute approach with a large number of permutations can therefore be prohibitive.

To overcome this problem, it is relevant to optimize the number of permutations for each estimated p-value to ensure a given convergence at a lower cost. The objective of the next section is to propose new algorithms for sequentially estimating the p-values, according to the GSA purpose.

3. Sequential permutation-based test

The permutation tests based on dependence measures presented in Section 2.2 are efficient tools for the selection of the influential input variables. However, they are considerably depending on the prior choice of the number of permutations B: a too low B does not allow sufficient convergence of p-value (and conclusion of the test) to be achieved. To get more general results on conver-

gence, let's go back to the formula of the p-value permutation-based estimator:

$$\hat{p}^B_{val,i} = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}_{\widehat{\mathrm{HSIC}}_b(X_i,Y) > \widehat{\mathrm{HSIC}}_{obs}(X_i,Y)}.$$

It can be shown that $\hat{p}^B_{val,i}$ is an unbiased estimator of the p-value $p_{val,i}$, since permutations are independent (see [10] for demonstration). The law of large numbers ensures that $p^B_{val,i}$ will converge to $p_{val,i}$ as B grows. The remaining question is how fast. The answer lies in the variance of the estimator which is:

$$\mathbb{VAR}(\hat{p}^B_{val,i}) = \frac{1}{B}\ p_{val,i}\ (1 - p_{val,i}).$$

The exact value of $p_{val,i}$ is actually not known. Anyway, considering a target coefficient of variation, we can estimate how many permutations are needed to compute a given probability $p_{val,i}$ within some prescribed accuracy. Indeed the coefficient of variation may be cast as:

$$\delta = \frac{\sqrt{\frac{1}{B}p_{val,i}(1 - p_{val,i})}}{p_{val,i}} = \sqrt{\frac{1 - p_{val,i}}{Bp_{val,i}}}.$$

From this relationship, one can conversely find the number of permutations required for a given probability and coefficient of variation

$$B = \frac{1 - p_{val,i}}{\delta^2 p_{val,i}}.$$

For instance, if we look for a p-value close to 5% (which corresponds to the usual level of the test) with a coefficient of variation of 5%, we would need approximately 7600 permutations. This can be hardly tractable even if each HSIC measure is relatively cheap to calculate (all the more so when the procedure is repeated). Consequently, it would be of great interest to have sequential process to control the convergence of the p-value estimation with regard to its interpretation for screening and ranking purposes, with the ultimate goal of optimizing the required number of permutations. Indeed, the decision whether or not to reject the hypothesis of independence can be made on the basis of the history of the estimate. Thus, based on the stagnation of this estimate, a decision can be taken much earlier. The same approach can be implemented for the ranking of the inputs. It is therefore not necessary to obtain a great precision on all the estimated p-values to converge on the screening or ranking results. Consequently, we propose in the following several approaches in order to well estimate the p-value with a reasonable number of permutations, according to the GSA purposes. In addition, this sequential estimation makes it possible to study and control in practice the convergence of the permutation-based estimation of p-values.

3.1. Stopping criterion based on screening

The sequentiality of the proposed estimation methods of the p-values leads us to define a stopping criterion. The first proposed approach relies on the quantity of interest e_i^B defined, for each input, by the test decision (screening decision):

$$e_i^B = \mathbb{1}_{\hat{p}_{val,i}^B < \alpha}, \tag{5}$$

where $\hat{p}_{val,i}^B = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}_{\widehat{\mathrm{HSIC}}_b(X_i,Y) > \widehat{\mathrm{HSIC}}_{obs}(X_i,Y)}$. The first *stopping criterion* is based (for each input) on the stagnation of e_i^B given by the following relation:

$$\forall k \in \{1, \ldots, l_0\} \ , \ e_i^B = e_i^{B-k}. \tag{6}$$

It ensures that the p-values are well estimated since the decision is unchanged during the l_0 previous steps. This first approach can be summarized by Algorithm 2.

In addition to the learning sample and the level of the test (α), the algorithm requires the specific following parameters: an initial and final number of permutations (B_{start} and B_{final} respectively), a number of permutations to be added to each iteration (B_{batch}) and a convergence parameter l_o which guarantees the stability of the estimate. Concerning the step 10 of Algorithm 2, for iteration j in the *while* loop, we update the estimation of the p-values using the previous estimate $p_{val,i}^{B_{start}+(j-1)B_{batch}}$ and the new batch of B_{batch} permutation-samples.

3.2. Stopping criterion based on ranking

The second criterion is based on the ranking of the inputs according to their p-values. For this, we define the rank vector given by

$$\mathbf{r}^B = rank\big((\hat{p}_{val,i}^B)_{i=1:d}^{\top}\big), \tag{7}$$

where $rank(.)$ is the ranking function. More precisely, $\mathbf{r}^B$ is a permutation on the set $\{1, \ldots, d\}$, which verifies that $\mathbf{r}^B(k) = i$ if and only if the variable X_i is the k-th in the ranking. The second *stopping criterion* is based on the stagnation of the ranking, i.e. $\mathbf{r}^B$, and is given by:

$$\forall k \in \{1, \ldots, l_0\} \ , \ \mathbf{r}^B = \mathbf{r}^{B-k}. \tag{8}$$

This second approach is summarized by Algorithm 3.

3.3. Stopping criterion based on both ranking and screening

Finally, the third criterion consists in checking the two previous ones in order to guarantee a good ordering and a good screening. The algorithm is similar to Algorithm 3 but with the following stopping criterion (step 10 of Algorithm 3):

$$S^B = \prod_{k=0}^{l_0} \left(\mathbb{1}_{\mathbf{r}^{B-k} = \mathbf{r}^B} \prod_{i=1}^{d} \mathbb{1}_{e_i^{B-k} = e_i^B} \right) \tag{9}$$

with $\mathbf{r}^B$ and e_i^B defined by Eq. (5) and Eq. (7) respectively. It should be noted that this criterion, more demanding to be satisfied, may require more permutations. This third approach will be denoted Algorithm 4.

Algorithm 2 – Screening-oriented sequential permutation-based independence test (for each X_i)

Require: The n-size learning sample $(\mathbf{X}_i, \mathbf{Y})$, B_{start}, B_{final}, B_{batch}, l_o and α

Compute $\widehat{\text{HSIC}}_{obs}(X_i, Y)$ from Eq. (3)
$B \longleftarrow B_{start}$
Generate B permutation-based samples $\left(\mathbf{X}_i, \mathbf{Y}_{[b]}\right)_{1 \leq b \leq B}$
Compute the B permutation-based estimators $\left(\widehat{\text{HSIC}}_b(X_i, Y)\right)_{1 \leq b \leq B}$ by replacing $\mathbf{Y}$ by $\mathbf{Y}_{[b]}$ in Eq. (3)
Estimate the p-value by Monte-Carlo estimator:

$$\hat{p}^B_{val,i} = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\widehat{\text{HSIC}}_b(X_i,Y) > \widehat{\text{HSIC}}_{obs}(X_i,Y)}$$

Compute the stopping criterion: $S_i^B = \prod_{k=1}^{l_0} \mathbb{1}_{e_i^{B-k} = e_i^B}$ with e_i^B defined by Eq. (5)
while $(S_i^B = 0$ **and** $B \leq B_{final})$ **do**
    Add a new B_{batch} iid permutation-based samples to the existing B samplings
    $B \longleftarrow B + B_{batch}$
    Update the p-value estimate $\hat{p}^B_{val,i}$ and stopping criterion S_i^B
end while
if $\hat{p}^B_{val,i} < \alpha$ **then**
    return reject $(\mathcal{H}_0^i)$
else
    return accept $(\mathcal{H}_0^i)$
end if

4. Numerical experiments and practical recommendations

4.1. Numerical experiments

In this section, the efficiency of the proposed algorithms is first assessed on two numerical examples well-known from the sensitivity analysis community. The first one is the G-Sobol function which is defined in dimension $d = 8$ and given by:

$$\mathcal{M}_1(\mathbf{X}) = \prod_{i=1}^{8} \frac{|4X_i - 2| + a_i}{1 + a_i}, \tag{10}$$

where $\forall i = 1, \ldots, d$, $a_i = \frac{i-2}{2}$ and the inputs $X_1, \ldots, X_8$ are independent random variables, following a uniform distribution over $[0, 1]$. The second analytical model is the Ishigami function defined in dimension $d = 3$ by:

$$\mathcal{M}_2(\mathbf{X}) = \sin(X_1) + 5\sin(X_2) + 0.1X_3^4 \sin(X_1), \tag{11}$$

Algorithm 3 – Ranking-oriented sequential permutation-based independence test, dealing with all inputs $\{X_1, \ldots, X_d\}$ simultaneously

Require: The n-size learning sample $(\mathbf{X}_1, \ldots, \mathbf{X}_d, \mathbf{Y})$, B_{start}, B_{final}, B_{batch}, l_o and α

for $i = 1$ to d **do**
  Compute $\widehat{\text{HSIC}}_{obs}(X_i, Y)$ from Eq. (3)
end for
$B \longleftarrow B_{start}$
Generate B permutation-based samples $\left(\mathbf{X}_1, \ldots, \mathbf{X}_d, \mathbf{Y}_{[b]}\right)_{1 \leq b \leq B}$
for $i = 1$ to d **do**
  Compute the B permutation-based estimators $\left(\widehat{\text{HSIC}}_b(X_i, Y)\right)_{1 \leq b \leq B}$ by replacing $\mathbf{Y}$ by $\mathbf{Y}_{[b]}$ in Eq. (3)
  Estimate the p-value by Monte-Carlo estimator:

$$\hat{p}^B_{val,i} = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}_{\widehat{\text{HSIC}}_b(X_i,Y) > \widehat{\text{HSIC}}_{obs}(X_i,Y)}$$

end for
Compute the stopping criterion: $S^B = \prod_{k=1}^{l_0} \mathbb{1}_{\mathbf{r}^{B-k} = \mathbf{r}^B}$ with $\mathbf{r}^B$ defined by Eq. (7)
while $(S^B_i = 0$ **and** $B \leq B_{final})$ **do**
  Add a new B_{batch} iid permutation-based samples to the existing B samplings
  $B \longleftarrow B + B_{batch}$
  Update the p-value estimate $\hat{p}^B_{val,i}$ and stopping criterion S^B_i
end while
if $\hat{p}^B_{val,i} < \alpha$ **then**
  return reject $(\mathcal{H}^i_0)$
else
  return accept $(\mathcal{H}^i_0)$
end if

Algorithm 4 – Ranking-Screening-oriented sequential permutation-based independence test (for $\{X_1, \ldots, X_d\}$)

Algorithm 3 with $S^B = \prod_{k=1}^{l_0} \left(\mathbb{1}_{\mathbf{r}^{B-k} = \mathbf{r}^B} \prod_{i=1}^{d} \mathbb{1}_{e_i^{B-k} = e_i^B} \right)$ with $\mathbf{r}^B$ and e_i^B defined by Eq. (5) and Eq. (7) respectively

where all inputs X_1, X_2 and X_3 are independent and uniformly distributed over $[-\pi, \pi]$.

To highlight the performance of the proposed algorithms, we generate 100

iid samples of inputs/output and compare the obtained results to the reference permutation-based test (Algorithm 1) performed with $B = 7600$. Note that the whole computational aspect is carried out in the $\mathbf{R}$[4] environment. We rely on the function `sensiHSIC` from the R `sensitivity` package to run the algorithms 1 to 4. The internal parameters of the sequential algorithms are set to $B_{start} = 100$, $B_{final} = 10000$ and $B_{batch} = 100$ and $l_0 = 200$. Note that the value of parameter l_0, which guarantees the stability of the estimate and avoids premature stops, will be discussed in Section 4.2. The number of permutations for Algorithm 1 is $B = 7600$. A level $\alpha = 5\%$ is set for all algorithms.

Table 1 supplies, for the two analytical models, the percentage of ranking and/or screening similar to the reference method, according to the objective of the algorithm (screening and/or ranking). More precisely, for Algorithm 2, the percentage is the one of good screening, i.e. the percentage of times where Algorithm 2 selects the same groups of influential and non-influential inputs as the reference test (Algorithm 1). For Algorithm 3, the percentage is the one of good ranking, i.e. the percentage of times where Algorithm 3 orders the inputs in the same way as the reference test. Finally, for Algorithm 4, it corresponds to the percentage of times where both screening and ranking results are similar. From this table, we note that the proposed Algorithms 2, 3 and 4 succeed very well in providing the same screening and/or ranking results as the reference algorithm, while significantly reducing the number of permutations. Indeed, the mean number of permutations required to reach each stopping criterion (written in brackets in Table 1) indicates that, in average:

- for the G-Sobol function, around $B = 300$ and $B = 700$ permutations are required to find the screening and ranking results, respectively (regardless of n);
- for the Ishighami function, around $B = 300$ are required for both targets,

against, as we recall, $B = 7600$ here for the reference algorithm. Again, we note that Algorithm 4 needs more permutations to satisfy both conditions, this increase obviously depending on the model.

Figures 1 and 2 illustrates on a single sample of size $n = 200$ the sequential estimation of p-values for the G-Sobol $\mathcal{M}_1$ and Ishigami models $\mathcal{M}_2$ respectively. The vertical lines indicate where the algorithms stop the estimation. For Algorithm 2 and the two models, process stops at $B = 300$ for all the inputs: this corresponds to two iterations of the algorithm (since $B_{batch} = 100$). More iterations are necessary for Algorithm 3 and Algorithm 4: four and one more iteration for $\mathcal{M}_1$ and $\mathcal{M}_2$, receptively. We note that for this particular example (model and sample), both algorithms stop at the same iteration. It is also recalled that, in general, it is not guaranteed to obtain a screening result with

[4]Language and opensource environment for statistical calculation and data analysis, available on the CRAN website (Comprehensive R Archive Network) : `https://cran.r-project.org/`.

Algorithm 3, similar to that of the reference solution, since Algorithm 3 is only based on ranking.

$\mathcal{M}_1$	$n = 20$	$n = 50$	$n = 70$	$n = 100$	$n = 150$	$n = 200$
Algorithm 2	100 (306)	100 (308)	100 (307)	100 (307)	100 (303)	100 (304)
Algorithm 3	98 (659)	98 (639)	100 (635)	99 (631)	100 (599)	100 (607)
Algorithm 4	100 (753)	100 (675)	100 (713)	100 (707)	100 (609)	100 (659)
$\mathcal{M}_2$	$n = 20$	$n = 50$	$n = 70$	$n = 100$	$n = 150$	$n = 200$
Algorithm 2	100 (306)	100 (303)	100 (303)	100 (303)	100 (301)	100 (301)
Algorithm 3	100 (331)	100 (323)	100 (313)	100 (301)	100 (299)	100 (299)
Algorithm 4	100 (383)	100 (343)	100 (341)	100 (301)	100 (327)	100 (335)

Table 1: G-Sobol function $\mathcal{M}_1$ (top) and Ishigami function $\mathcal{M}_2$ (bottom) – Percentage of similar screening and/or ranking obtained with sequential algorithms, compared to the results obtained from the simple permutation-based algorithm (Algorithm 1), for different sample sizes and with a level $\alpha = 5\%$ for all the tests. Written in brackets is the mean number of permutations required to reach the stopping criterion.

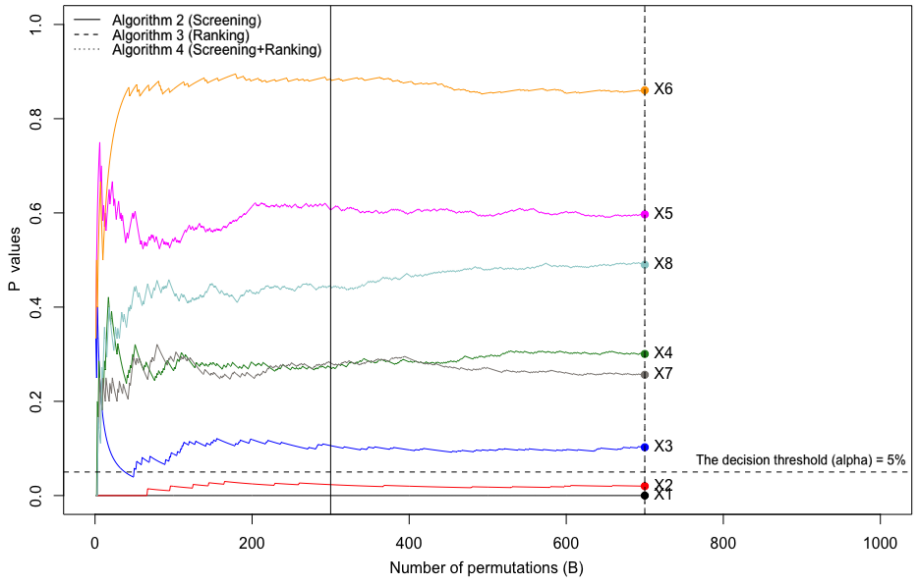


Figure 1: G-Sobol function $\mathcal{M}_1$ – Sequential estimation of the p-values according to the number of permutations B, for a given sample of $n = 200$ simulations (the vertical lines of Algorithm 3 and 4 overlap).

4.2. Sensitivity to the parameter l_0

Among the parameters of the proposed sequential algorithms, the parameter l_0 is of major importance. Indeed, this parameter is used to check the stability of the estimate. If it is too small, there is a risk of stopping the estimate while

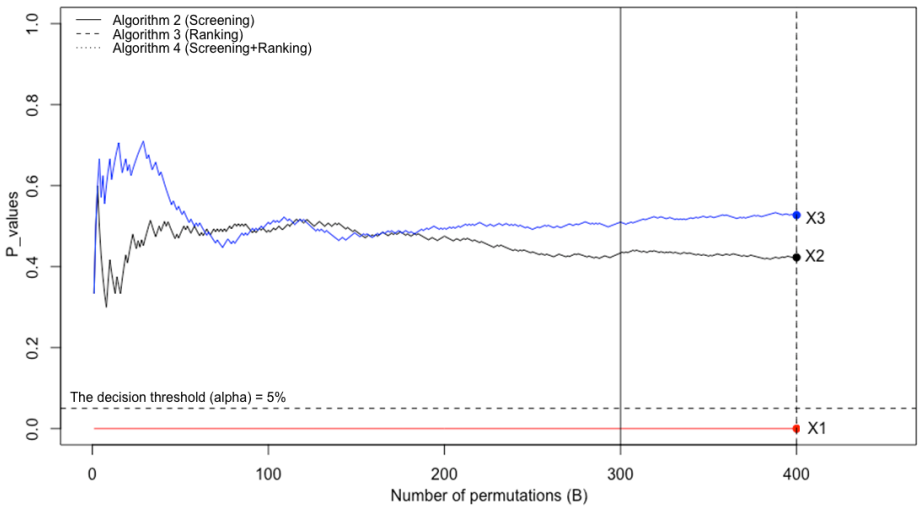


Figure 2: Ishighami function $\mathcal{M}_2$ – Sequential estimation of the p-values according to the number of permutations B, for a given sample of $n = 200$ simulations (the vertical lines of Algorithm 3 and 4 overlap).

convergence is not yet ensured. On the contrary, the greater its value, the less the gain in terms of reduction of the number of permutations B. From a practical point of view and based on our experience, we can recommend to set $l_0 = 200$, to ensure both a good ranking and a good screening while significantly optimize B. However, it is relevant to check the sensitivity of the algorithm to this parameter. For this, we consider the analytical model $\mathcal{M}_1$ and vary the parameter $l_0 \in \{50, 100\}$. The other parameters B_{start}, B_{final}, B_{batch} and α remain unchanged. The percentage of similar ranking and/or screening to the reference test is given by Table 2 (similar protocol to Table 1). Two remarks can be made. The first one is that Algorithm 2 is not sensitive (for this model) to the parameter l_0, since we can correctly classify the inputs even with a value set to 50. The second is that the other two algorithms lose efficiency for low l_0 values ($l_0 = 50$); a value of $l_0 = 100$ is at least necessary and $l_0 = 200$ offers a good compromise between a good performance guarantee and an optimal procedure. Same behaviours and remarks, not described here for the sake of brevity, are observed for $\mathcal{M}_2$.

5. Application on IBLOCA accidental use

To finish illustrating the practical benefit of the methodology, we consider here an accidental scenario on a Pressurized Water Reactor (PWR), namely the loss of primary coolant accident due to intermediate break described in [17]. A simple HSIC-based GSA was performed by [17] in order to identify the non-influential inputs and rank the influential ones. This screening and ranking step

$\mathbf{l_0 = 50}$	$n = 20$	$n = 50$	$n = 70$	$n = 100$	$n = 150$	$n = 200$
Algorithm 2	98 (202)	99 (202)	100 (201)	100 (201)	100 (201)	100 (200)
Algorithm 3	61 (295)	63 (259)	66 (273)	68 (275)	65 (256)	61 (265)
Algorithm 4	65 (328)	67 (281)	70 (280)	72 (279)	64 (267)	69 (290)
$\mathbf{l_0 = 100}$	$n = 20$	$n = 50$	$n = 70$	$n = 100$	$n = 150$	$n = 200$
Algorithm 2	99 (203)	99 (204)	100 (203)	100 (204)	100 (202)	100 (202)
Algorithm 3	94 (289)	95 (335)	92 (369)	90 (357)	95 (357)	98 (349)
Algorithm 4	95 (478)	97 (389)	93 (421)	98 (388)	94 (383)	98 (369)
$\mathbf{l_0 = 200}$	$n = 20$	$n = 50$	$n = 70$	$n = 100$	$n = 150$	$n = 200$
Algorithm 2	100 (306)	100 (308)	100 (307)	100 (307)	100 (303)	100 (304)
Algorithm 3	98 (659)	98 (639)	100 (635)	99 (631)	100 (599)	100 (607)
Algorithm 4	100 (753)	100 (675)	100 (713)	100 (707)	100 (609)	100 (659)

Table 2: G-Sobol function $\mathcal{M}_1$ – Percentage of similar screening and/or ranking obtained with sequential algorithms, compared to the results obtained from the simple permutation-based algorithm, for different sample sizes and different parameters l_0. Written in brackets is the mean number of permutations required to reach the stopping criterion.

was preliminary to the building of a metamodel for the final estimation of high-order quantiles of the output. In the following, we apply on this use-case our sequential methodology for screening and ranking with permuted HSIC-based independence tests.

5.1. Brief description of the use case

In support of regulatory work and nuclear power plant design and operation, safety analysis considers the so-called "Loss Of Coolant Accident" which takes into account a double-ended guillotine break with a specific size piping rupture. The numerical model is based on code CATHARE2 (V2.5_3mod3.1) which simulates the time evolution of physical quantities during a thermal-hydraulic transient. The model used is representative of an *Intermediate Break Loss Of Coolant Accident* (IBLOCA) [18]. The simulated accidental transient is an IBLOCA with a break on the cold leg and no safety injection on the broken leg (see [17] for more details). In this use-case, $d = 27$ scalar input variables of CATHARE2 code are uncertain. These inputs listed in Table 3 correspond to various system parameters. Within a probabilistic approach, the uncertainty on the 27 inputs is defined by probability density functions (pdf), which can be uniform, log-uniform, normal or log-normal. The output variable of interest is a single scalar which is the maximal *peak cladding temperature* (PCT) during the accident transient.

5.2. Results of global sensitivity analysis

As detailed in [17], a sample of $n = 500$ CATHARE2 simulations is available built from a Latin Hypercube Design and following the prior distributions of inputs defined in Table 3. The histogram of the obtained values for the output of interest, namely the PCT, is given by Figure 3 (temperature is in °C). A

Type of inputs	Inputs	pdf[a]	Physical models
Heat transfer in the core	X_1	$\mathcal{N}$	Departure from nucleate boiling
	X_2	$\mathcal{U}$	Minimum film stable temperature
	X_3	$\mathcal{LN}$	HTC[b] for steam convection
	X_4	$\mathcal{LN}$	Wall-fluid HTC
	X_5	$\mathcal{N}$	HTC for film boiling
Heat transfer in the steam generators U-tube	X_6	$\mathcal{LU}$	HTC forced wall-steam convection
	X_7	$\mathcal{N}$	Liquid-interface HTC for film condensation
Wall-steam friction in the core	X_8	$\mathcal{LU}$	
Interfacial friction	X_9	$\mathcal{LN}$	Steam generator outlet plena and crossover legs together
	X_{10}	$\mathcal{LN}$	Hot legs (horizontal part)
	X_{11}	$\mathcal{LN}$	Bend of the hot legs
	X_{12}	$\mathcal{LN}$	Steam generator inlet plena
	X_{13}	$\mathcal{LN}$	Downcomer
	X_{14}	$\mathcal{LN}$	Core
	X_{15}	$\mathcal{LN}$	Upper plenum
	X_{16}	$\mathcal{LN}$	Lower plenum
	X_{17}	$\mathcal{LN}$	Upper head
Condensation	X_{18}	$\mathcal{LN}$	Downcomer
	X_{19}	$\mathcal{U}$	Cold leg (intact)
	X_{20}	$\mathcal{U}$	Cold leg (broken)
	X_{27}	$\mathcal{U}$	Jet
Break flow	X_{21}	$\mathcal{LN}$	Flashing (undersaturated)
	X_{22}	$\mathcal{N}$	Wall-liquid friction (undersaturated)
	X_{23}	$\mathcal{N}$	Flashing delay (undersaturated)
	X_{24}	$\mathcal{LN}$	Flashing (saturated)
	X_{25}	$\mathcal{N}$	Wall-liquid friction (saturated)
	X_{26}	$\mathcal{LN}$	Global interfacial friction (saturated)

[a] $\mathcal{U}$, $\mathcal{LU}$, $\mathcal{N}$ and $\mathcal{LN}$ respectively stands for uniform, log-uniform, normal and log-normal probability distributions.

[b] Heat Transfer Coefficient.

Table 3: IBLOCA test case – List of the 27 uncertain input parameters, associated pdf and physical models in CATHARE2 code.

kernel density estimator of the data is also added on the plot to provide an estimator of the probability density function.

The first analysis consists in highlighting the performance of the sequential-permutation based test for a screening purpose. To do so, we compare the result obtained by Algorithm 2 to the reference solution given by the simple permutation-based test (Algorithm 1). Parameters of Algorithm 2 are set to $B_{start} = 100$, $B_{final} = 10000$, $B_{batch} = 100$ and $l_o = 200$. Algorithm 1 is performed with $B = 4000$ and a level $\alpha = 5\%$ is set for all the algorithms.

Figure 4 shows the sequential estimation of the p-values according to the number of permutations. The p-values estimated at the end of the sequential

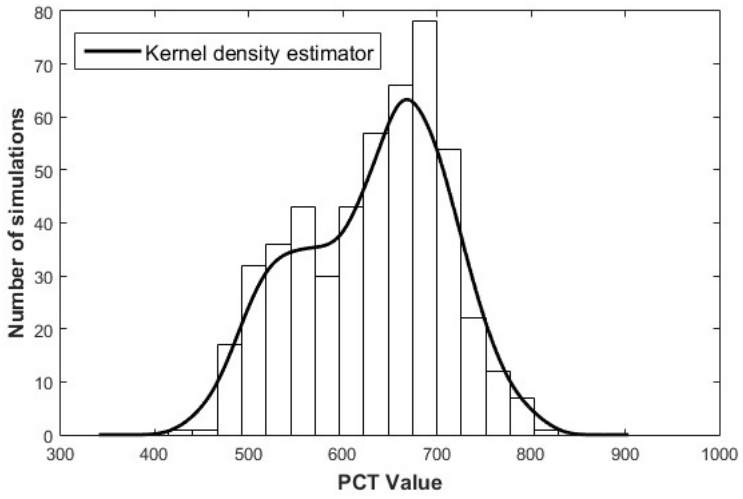


Figure 3: IBLOCA test case – Histogram of the PCT output from the sample of size $n = 500$.

process are also given by Table 4. The level of the independence tests is still set at $\alpha = 5\%$. The analysis of results (similar to those in [17]) shows a group of 8 influential variables (X_2, X_9, X_{10}, X_{12}, X_{13}, X_{14}, X_{15}, X_{21}) and another group of 19 non-influential. Same classification is obtained with the optimized algorithm and the reference one, but with a much smaller total number of permutations: around 8500 ($300 \times 26 + 700$) for Algorithm 2 against 108000 (4000×27) for the basic algorithm. The observation of convergence for Algorithm 2 shows that the number of permutations could have been further reduced for most variables by reducing B_{start}, B_{batch} and/or l_0. Moreover, a level set to $\alpha = 10\%$ could have reduced this number since the estimated p-value of X_6 is located around the decision threshold $\alpha = 5\%$ (number of permutation increased by the stabilization of X_6 p-value above the threshold). Note that, in a safety framework, it is recommended to keep an input variable with a p-value as close to the decision threshold.

Inputs	X_2	X_9	X_{10}	X_{12}	X_{14}	X_{15}	X_{21}	X_{13}	X_6
$\hat{p}_{val,\text{ Algo 2}}$	0	0	0	0	0	0	3.10^{-3}	0.019	**0.054**
$\hat{p}_{val,\text{ Algo 1}}$	0	5.10^{-4}	0	0	0	4.10^{-3}	1.10^{-3}	0.021	**0.050**

Table 4: IBLOCA test case – P-values estimated by Algorithms 1 and 2 for the influential inputs.

The second analysis consists in checking the performance of Algorithm 4 for screening and ranking. Figure 5 shows the sequential estimation of the p-values and the rank of the inputs variables. As shown in this figure, this is mainly driven by the discrimination between non-influential inputs, such as X_8 and X_{11} p-values curves or X_8 and X_{26}. We notice that the number of permutations is increased to $B = 1400$ to guarantee the convergence of both

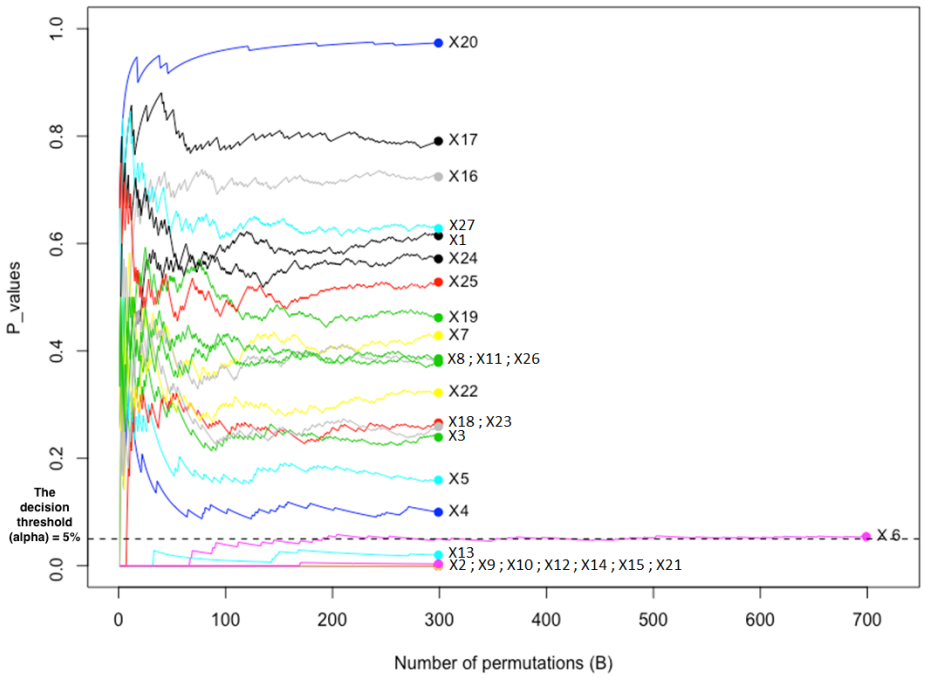


Figure 4: IBLOCA test case – Sequential estimation of p-values by Algorithm 2 (screening), according to the number of permutations.

criteria simultaneously (around a total of 37800 permutations). We also remark that all variables are similarly ranked except for X_8 and X_{11} (indicated by circles in Figure 5). We also notice that some very influential variables share the same place in the ranking (rectangle in Figure 5), this is due to the zero values of p-values estimates.

Remark 2. *When the dependence on several variables is very strong, the estimated p-values are zero. More precisely, the observed HSIC for two very influential inputs are very far in the distribution tail of HSIC statistics under independence. This occur when dependence of both inputs is clearly and strongly detected, this happening all the more as sample size is large. Consequently in that case, it is no possible to rank the variables with p-values and* R^2_{HSIC} *(Eq (4)) can then be used in addition for ranking.*

From a physical interpretation point of view, we notice the predominant influence of the interfacial friction coefficient in the horizontal part of the hot legs (X_{10}) and the interfacial friction coefficient in the steam generator inlet plena (X_{12}). As explained in [17], larger values of X_{10} in the horizontal part of the hot legs lead to larger PCT. This can be explained by the increase of vapor which brings the liquid in the horizontal part of hot legs, leading to a reduction of the liquid water return from the rising part of the U-tubes of the steam generator to

the core (through the hot branches and the upper plenum). Since the amount of liquid water available to the core cooling is reduced, higher PCT are observed. For X_{12}, its higher values lead to a greater supply (by the vapor) of liquid possibly stored in the water plena to the rest of the primary loops (then lower PCT).

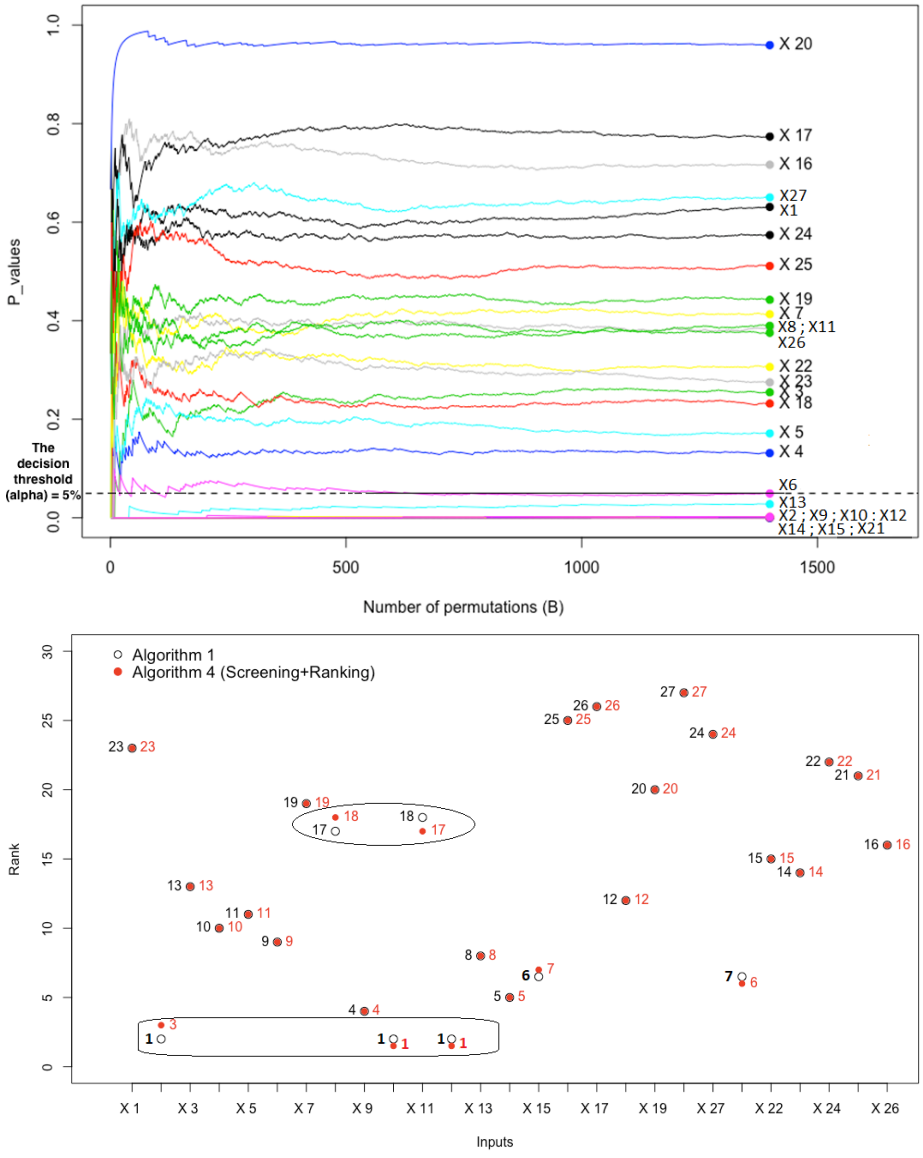


Figure 5: IBLOCA test case – Top: Sequential estimation of p-values by Algorithm 4 (screening + ranking), according to the number of permutations. Bottom: Rank of the inputs based on the p-values estimated by Algorithm 4.

The previous study was carried out on a learning sample of size $n = 500$ and some decision-making still seems indecisive, such as the classification as influential or not of the variable X_6. In addition, it is relevant to conduct convergence studies (according to n) in order to take into account the sampling uncertainties, and assess the stability and convergence of GSA results. For this, we will consider a bootstrap-based approach which is made tractable thanks to our optimized algorithms. More precisely, for different sample sizes $n \in \{100, 200, 300, 400, 500\}$, 100 n-size random samples are randomly drawn with replacement from the original data set. Then, for each sample thus generated, Algorithm 2 (screening) is performed and we store if the variable is selected or not. Figure 6 shows the selection rate of each variable as influential, as a function of the sample size n. The inputs variables can be classified into 2 groups:

- those that are detected as influential (X_2, X_{10}, X_{12} or X_{21}, e.g.) or not (X_3, X_4, X_{11} or X_{16}, e.g.) very quickly, with a high rate, even for small sample sizes. Remember that the levels of independence tests (error or first order) is set at 5%, it is therefore normal to have detection cases as influential even for non-influential variables;
- a second group of variables ($X_9, X_{13}, X_{14}, X_{15}$, e.g.) whose detection is more indecisive, significantly more variable for smaller n, and even not yet converged for $n = 500$. In particular, this is the case of X_6 where the trend of increasing detection rates suggests that this variable could be detected as significantly influential with more data. In a context of reduction of variables for a safety study, this reinforces the idea already mentioned to keep this variable.

Same convergence studies could also be done for ranking (or screening-ranking) results. Generally speaking, such a convergence study also makes it possible to assess whether a dependence is easy to detect or not, this very probably depending on the global or more local nature of the dependence.

From a computational point of view, thanks to our optimized algorithms this study could be carried out within a reasonable time. Indeed, if we look for a probability close to the level set (fixed at 5%) with a coefficient of variation of 5%, we would need hours to perform this study, as detailed in Table 5 for a Intel 2,3 GHz processor.

Sample size (n)	100	200	300	400	500
Algorithm 1	0.61	1.97	4.52	8.42	13.74
Algorithm 2	0.05	0.17	0.35	0.54	0.85

Table 5: IBLOCA test case – Running time (in hours) for a Intel 2,3 GHz processor.

This application clearly shows the gain in number of permutations, while preserving the accuracy of the results in terms of screening, ranking or both.

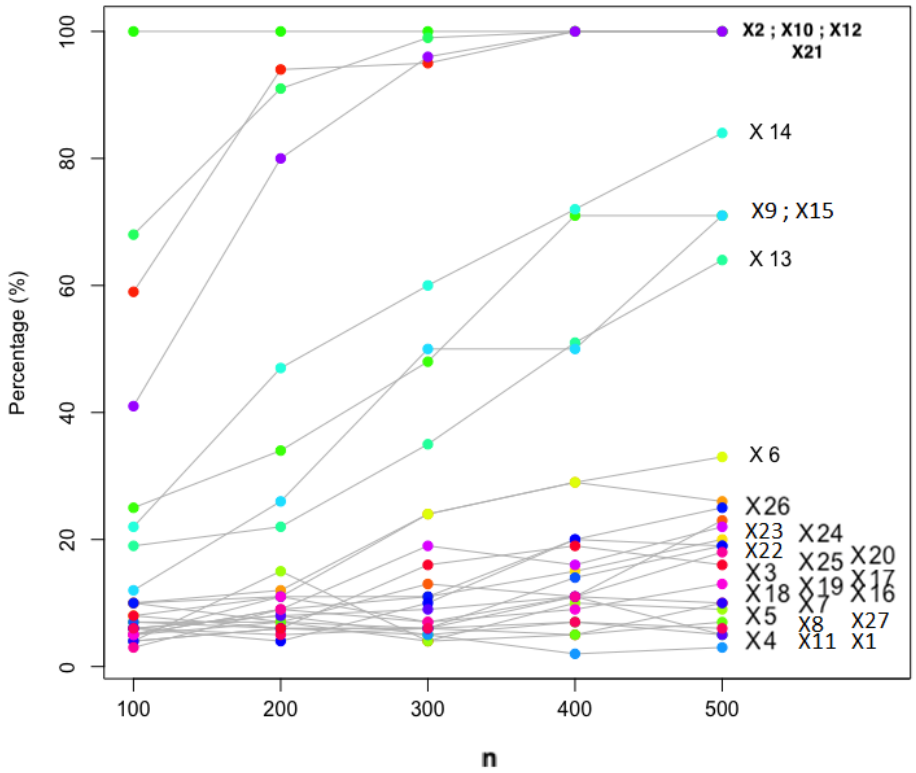


Figure 6: IBLOCA test case – Selection rate of inputs as influential, according to the sample size n.

In addition, confidence in the results obtained is also significantly increased by checking the convergence of the estimated p-values. Finally, accelerating the algorithm by optimizing permutations makes an external bootstrap loop computationally tractable: convergence studies can be performed to evaluate the convergence of results, according to the sample size.

6. Conclusion

In this paper, new goal-oriented algorithms were proposed to perform independence tests with HSIC for screening and ranking purposes in the context of global sensitivity analysis of numerical simulators. More precisely, we proposed an efficient methodology and associated operational tool in order to optimize the number of permutations in permuted HSIC-based tests. Resulting optimized tests can be applied whatever the size of sample and the number of inputs. We also quantified the resulting benefit on numerical examples and industrial application.

The permutation-tests based on HSIC are efficient and relevant statistical tools to identify which inputs significantly impact the output value and also to

rank the inputs by order influence on the output. These tests are built upon a n-size sample of inputs/output and rely on the estimation of a p-value under independence hypothesis. These p-values can be estimated either by permutation method which makes it possible to deal with simulation samples of smaller size (a few tens to one or two thousands). However, a brute (or blind) approach performed with a large number of permutations, which ensures convergence of the estimated p-value, can be prohibitive in practice when the testing procedure is repeated a large number of times. To overcome this, several strategies have been proposed to estimate this quantity in a greedy fashion, according to the final goal: screening-oriented, ranking-oriented and ranking-screening-oriented sequential permuted tests. These algorithms save a significant number of permutations and computation time (in practice, up to more than a factor 10, according to the model and the controlled criterion).

In addition to their dedicated tasks, there is a rich field of applications where our optimized procedures for permuted tests will be of great benefit. For example, practitioners may be interested in taking into account the sampling uncertainty and control the convergence according to the number of simulations in the hypothesis testing. For this, an outer loop based on bootstrap process could be used to compute confidence intervals for the p-values. Therefore, this kind of study will require a lot of permutations and such goal-oriented sequential approaches will be very useful.

Other variants of these algorithms could be envisaged according to the objectives of the sensitivity analysis and our algorithms can easily be adapted. For instance, the third algorithm can be adapted to screen the non-influential variables and rank only the influential variables, the order of the non-influential being not relevant.

Acknowledgments

The authors are grateful to Henri Geiser and Thibault Delage who performed the simulations using the CATHARE2 code. CATHARE2 code is developed under the collaborative framework of the NEPTUNE project, supported by CEA, EDF (Electricité De France), Framatome and IRSN (Institut de Radioprotection et de Sûreté Nucléaire).

References

[1] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity analysis in practice: a guide to assessing scientific models, Chichester, England (2004).

[2] W. Hoeffding, A class of statistics with asymptotically normal distribution, in: Breakthroughs in Statistics, Springer, 1992, pp. 308–334.

[3] I. M. Sobol, Sensitivity estimates for nonlinear mathematical models, Mathematical modelling and computational experiments 1 (1993) 407–414.

[4] S. Da Veiga, Global sensitivity analysis with dependence measures, Journal of Statistical Computation and Simulation 85 (2015) 1283–1305.

[5] M. De Lozzo, A. Marrel, New improvements in the use of dependence measures for sensitivity analysis and screening, Journal of Statistical Computation and Simulation 86 (2016) 3038–3058.

[6] A. Meynaoui, M. Albert, B. Laurent, A. Marrel, Aggregated test of independence based on hsic measures, arXiv preprint arXiv:1902.06441 (2019).

[7] A. Marrel, V. Chabridon, Statistical developments for target and conditional sensitivity analysis: application on safety studies for nuclear reactor, HAL Preprint available at https://hal.archives-ouvertes.fr/hal-02541142 (2020).

[8] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with hilbert-schmidt norms, in: International conference on algorithmic learning theory, Springer, 2005, pp. 63–77.

[9] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, A. J. Smola, A kernel statistical test of independence, in: Advances in neural information processing systems, 2008, pp. 585–592.

[10] A. Meynaoui, New developments around dependence measures for sensitivity analysis: application to severe accident studies for generation IV reactors, Ph.D. thesis, University of Toulouse, 2019.

[11] A. Meynaoui, A. Marrel, B. Laurent, New statistical methodology for second level global sensitivity analysis, arXiv preprint arXiv:1902.07030 (2019).

[12] A. Marrel, B. Iooss, V. Chabridon, Statistical identification of penalizing configurations in high-dimensional thermalhydraulic numerical experiments: the ICSCREAM methodology, Preprint hal-02535146 (2020).

[13] N. Aronszajn, Theory of reproducing kernels, Transactions of the American mathematical society 68 (1950) 337–404.

[14] A. Gretton, Notes on mean embeddings and covariance operators, 2019.

[15] A. Gretton, A simpler condition for consistency of a kernel independence test, arXiv preprint arXiv:1501.06103 (2015).

[16] Z. Szabó, B. K. Sriperumbudur, Characteristic and universal tensor product kernels, Journal of Machine Learning Research 18 (2018) 1–29.

[17] B. Iooss, A. Marrel, Advanced methodology for uncertainty propagation in computer experiments with large number of inputs, Nuclear Technology 205 (2019) 1588–1606.

[18] P. Mazgaj, J.-L. Vacher, S. Carnevali, Comparison of CATHARE results with the experimental results of cold leg intermediate break LOCA obtained during ROSA-2/LSTF test 7, EPJ Nuclear Sciences & Technology 2 (2016).