A comparison of global sensitivity techniques and sampling method

X. Sun [a], **S. Roberts** [a], **B. Croke** [a,b], **A. Jakeman** [b]

[a] *Mathematical Sciences Institute, Australian National University, Canberra*
[b] *Fenner School of Environment and Society, Australian National University, Canberra*
Email: Xifu.Sun@anu.edu.au

Abstract: Inspired by Tarantola et al. (2012), we extend their analysis to include the Latin hypercube and Random sampling methods. In their paper, they compared Sobol' quasi-Monte Carlo and Latin supercube sampling methods by using a V-function and variance-based sensitivity analysis. In our case we compare the convergence rate and average error between Sobol', Latin hypercube, and Random sampling methods from the Chaospy library, keeping everything else the same as in their paper. We added the Random sampling method to test if the other two sampling methods are indeed superior. The results from our code confirm the results of their paper, where Sobol' has better performance than Latin hypercube sampling in most cases, whilst they both have higher efficiency than is achieved with Random sampling.

In addition we compared the explicit forms of 'Jansen 1999' total effects estimator used in Tarantola et al. (2012) with the 'Sobol' 2007' estimator, again keeping sample sizes and the test function the same. Results confirm that the 'Jansen 1999' estimator is more efficient than 'Sobol' 2007'. The presentation will also include the Morris sampling method and other test functions to further test efficiency among all the sampling methods on different cases.

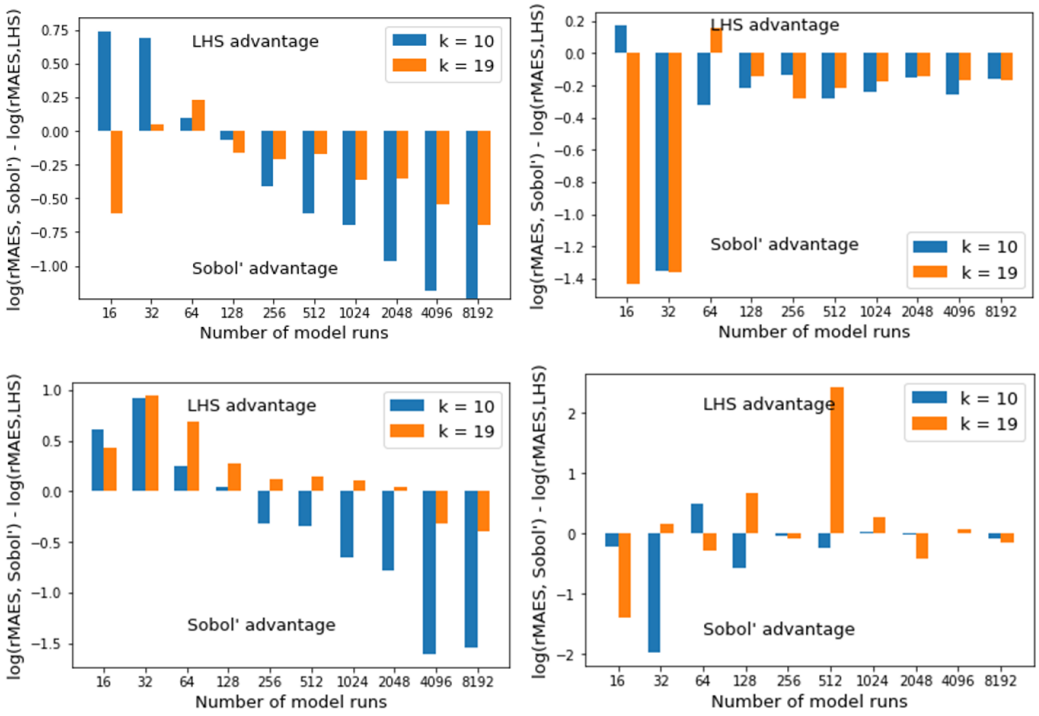

Figure 1. Comparison of rMAES value for Sobol' and Latin Hypercube. Top left is for type A1-1, top right is for type A2, bottom left is for type B, and bottom right is for type C.

Keywords: *Sobol', Latin hypercube, Random sampling, global sensitivity analysis, variance based, total effects estimator*

1 INTRODUCTION

The goal of sensitivity analysis is to study how the uncertainty of model output can be attributed to the uncertainty of input as well as the interaction between inputs. Nowadays, many models are very complex and expensive to run, so it is extremely useful to identify any important or unimportant inputs through various sensitivity analysis techniques before actually running the models. After we choose the desired technique, we want to know how big the sample size should be enough to get relatively small statistical error. Due to this concern, we need to choose the sampling method with best efficiency or least time-consuming, and the choices are based on the need of end users. In Tarantola et al. (2012), they used variance-based sensitivity analysis on V-function to compare the efficiency of randomised Sobol' QMC design and Latin Supercube method based on C^{++} library. Inspired by their paper, we want to extend their experiment to Python and compare three different sampling methods: Sobol', Latin Hypercube, and Python Random methods. In addition to the three sampling methods, Saltelli et al. (2010) summarised different forms for first order and total order effects, and they claimed that 'Jansen 1999' is better compared to 'Sobol' 2007' on total order effects sensitivity indices estimator. We will also test this result in our Python code environment. The outline of this paper is as follows: in Section 2, we will give a brief introduction of variance based sensitivity analysis; in Sections 3 and 4, we will summarise the sampling methods that we choose to use, and we will introduce the basic idea of original paper's sampling strategies, test function, and test cases; in Section 5, we will show the results on the comparison of different sampling methods and of two different total effect sensitivity indices estimators.

2 VARIANCE-BASED SENSITIVITY ANALYSIS

Sobol' first mentioned the expansion into summands of different dimensions in Sobol' (1969). Later, Sobol' generalised the expansion theorem, and then he proceed to bring up the concept of "freezing unessential variables" in Sobol' (1993). Homma and Saltelli (1996) called this "freezing" concept as "Sobol' sensitivity indices of the first order" and further improved it into "the total effect index". As most models are extremely complex, it is impossible to get the exact values for the indices for such models, since these indices are obtained through numerical means and will always have a finite confidence interval. Because of this reason, there are many discussions based how we should estimate those indices, and readers can get more information about various approaches to approximate indices from Sobol' (1993), Homma and Saltelli (1996), Saltelli et al. (2010), Jansen (1999), and Sobol' (2007). In our paper, we will use the estimators based on Tarantola et al. (2012). For readers' convenience, we will briefly summarise the idea of first and total effect sensitivity indices here. The variance-based sensitivity analysis starts with the variance decomposition methods.
We define a function $f(X)$ as

$$f(X) = f(X_1, X_2, \ldots, X_n),$$

where the domain is a n-dimensional unit cube $\Omega = \{X|0 \leq x_i \leq 1|i = 1 \ldots n\}$. Then we can write the decomposition of function f as:

$$f = f_0 + \sum_{i=1}^{n} f_i + \sum_{1\leq i\leq j\leq n}\sum f_{ij} + \cdots + f_{12\ldots n},$$

where $f_i = f_i(X_i)$, $f_{ij} = f_{ij}(X_i, X_j)$, and j represents the same index as i. Next, we write the decomposition form of variance of $f(X)$, $V(f)$, as

$$V(f) = \sum_{i=1}^{n} V_i + \sum_{1\leq i\leq j\leq n}\sum V_{ij} + \cdots + V_{12\ldots n}. \tag{1}$$

Divide $V(f)$ on both sides of above equation (1), we get

$$1 = \sum_{i=1}^{n} S_i + \sum_{1\leq i\leq j\leq n}\sum S_{ij} + \cdots + S_{12\ldots n}.$$

We call the S_i term as the Sobol' sensitivity indices of the first order, and it is defined as

$$S_i = \frac{V_i}{V(f)} = \frac{V_{X_i}(E_{X_{\sim i}}(f|X_i))}{V(f)}.$$

The notation $X_{\sim i}$ means that all X variables except the ith variable. The approximation of S_i's numerator and denominator are known as

$$V_{X_i}(E_{X_{\sim i}}(f|X_i)) \approx \frac{1}{N}\sum_{j=1}^{N} f(B)_j(f(A_B^{(i)})_j - f(A)_j),$$

and

$$V(f) \approx \frac{1}{N}\sum_{j=1}^{N} f^2(A)_j - \frac{1}{N}\sum_{j=1}^{N} f(A)_j f(B)_j.$$

Similarly, the total effects' S_{Ti} is defined as

$$S_{Ti} = \frac{E_{X_{\sim i}}(V_{X_i}(f|X_{\sim i}))}{V(f)},$$

where

$$E_{X_{\sim i}}(V_{X_i}(f|X_{\sim i})) \approx \frac{1}{2N}\sum_{j=1}^{N}(f(A)_j - f(A_B^{(i)})_j)^2. \tag{2}$$

Matrices A and B are two $N \times k$ sampling matrices generated by sampling methods, and $A_B^{(i)}$ is a matrix where the ith column comes from B and the rest is from A. Moreover, $(A)_j$ denotes the jth row of matrix A. We follow the same idea of sampling design in Section 3.2 of Tarantola et al. (2012) but with different sampling methods. The proof of the approximation forms of S_i and S_{Ti} are fairly easy, since it only uses the definition of expected value, variance, and Monte Carlo approximation. More detailed proofs of S_i and S_{Ti} are in Homma and Saltelli (1996) and Saltelli et al. (2010).

According to Saltelli et al. (2010), the formula of total effect sensitivity indices (2) is an improved version by Jansen (1999), and its efficiency is claimed to be much better than that defined in Sobol' (2007). We will test again the efficiency of both estimators in the later section and confirm this result.

3 TEST FUNCTION AND CASES

We still use the famous benchmark test function for sensitivity analysis V-function as our test function:

$$Y = V(X_1, X_2, \ldots, X_k, a_1, \ldots, a_k) = \prod_{i=1}^{k} v_i$$

with,

$$v_i = \frac{|4X_i - 2| + a_i}{1 + a_i}.$$

The sampling points are $X_i, i = 1 \ldots k$, and we get them through the sampling designs from previous section. In addition, we use coefficients, $a_i, i = 1 \ldots k$, to control whether the corresponding input is a dominated input or not. We also follow the same five types of coefficients, and we briefly summarise them as below:

- A1-1 : $a_1 = a_2 = 0, a_{i>2} = 6.52$
- A1-2 : $a_{k-1} = a_k = 0, a_{i<k-1} = 6.52$
- A2 : $a_1 = \cdots = a_5 = 0, a_{i>5} = 6.52$
- B : $a_i = 6.52, i = 1, \ldots, k$
- C : $a_i = 0, i = 1, \ldots, k$.

Since we know the exact results of first-order effects for all the types, we can calculate how strong the interaction is for each type manually. We use the exact results of 10 dimensions as an example for here. Type B has the weakest interaction among all the types, where the interaction accounts for approximately 3% of the output variance. Type A1-1 and A1-2 both have the same amount of interaction which is 17.32%, while type A2 has 49.21%. Type C has the strongest interaction which is approximately 80%. As we increase the dimensions, the interaction will become slightly stronger. Please refer to Sobol' (1993) for more details about the test function and Kucherenko et al. (2011) for the classification of types of coefficients. The exact results of test function are used by this paper can be found in Appendix A of Tarantola et al. (2012).

4 SIMULATION RESULTS

We use the same sample size N as in Tarantola et al. (2012): 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192; however, as we mentioned in the previous section, the Sobol' quasi-Monte Carlo sample generator in Chaospy library can only generates up to 40 dimensions, so we limited the dimension to 10 and 19 because of the sampling design. We replicate each experiment 100 times and then calculate the average. In their paper, they used four indicators AES, AEST, MAES, and MAEST to represent the error of the sensitivity estimates. The indicators AES and AEST are called the mean absolute error of each S_i and each S_{Ti}, and the indicators MAES and MAEST are the corresponding average of AES and AEST. AES is defined as

$$AES_i = \frac{1}{R}\sum_{r=1}^{R} |S_i^{(r)} - \bar{S}_i|,$$

and MAES has the form of

$$MAES = \frac{1}{R}\frac{1}{K}\sum_{i=1}^{k}\sum_{r=1}^{R} |S_i^{(r)} - \bar{S}_i|,$$

where $\bar{S}_i$ is the exact solution and $S_i^{(r)}$ is the estimated solution. In these formulas, i still represents the index, and R is the total number of replicates. The r means which replicate that we are currently using. AEST and MAEST have exactly the same forms as AES and AEST but with S_{Ti} instead of S_i.

In our experiment, we also calculate the mean relatively error of each S_i and S_{Ti} and corresponding averages as well. The relative AES (rAES) is written as

$$rAES_i = \frac{1}{R}\sum_{r=1}^{R} \frac{|S_i^{(r)} - \bar{S}_i|}{|\bar{S}_i|},$$

and the relative MAES (rMAES) is

$$rMAES = \frac{1}{R}\frac{1}{K}\sum_{i=1}^{k}\sum_{r=1}^{R} \frac{|S_i^{(r)} - \bar{S}_i|}{|\bar{S}_i|}.$$

Through the experiments, we confirmed that Sobol' is more superior than Latin Hypercube in most cases, and they both have better performance than Python Random in average. Since we do not have enough pages to show all the figures and discuss about the similar results, so we will only talk about rMAES and rMAEST which are not included in Tarantola et al. (2012). If readers wish to see the figures of rAES and rAEST, please refer to our github page for more information. We will look at two sets of figures for each type: the first set is the comparison bar plots of rMAES of Sobol' vs. Latin Hypercube in Figure 1. The y-axis of these figures is the subtraction of log(rMAES of Sampling Method 2) from log(rMAES of Sampling Method 1), and the x-axis is the number of model runs N. If the bar is above zero, it means that sampling method 2 has advantage over sampling method 1, and vice versa. For our case, Sobol's has more advantage if the bar is below zero, and Latin Hypercube has more advantage if the bar is higher than zero. Since we can put the results from two cases with different dimensions ($k = 10, 19$) into one figure, it is easily to see how the difference of relative error changes with the change of dimensions. The second set of figures are log-log scaled plots of rMAEST vs. N for $k = 10$ in Figure 2 and $k = 19$ in Figure 3. The smaller the rMAES or rMAEST is, the smaller the statistical error is.

Type A1. Through the experiments on type A1-1 and type A1-2, we see that the results of rMAES and rMAEST of both types are quite similar, since both types have equally strong interaction. For here, we will only show the figures from type A1-1. From top left of Figure 1, we can see that Sobol' starts to show more advantages as we increase the number of model runs from $N = 128$ compared to Latin Hypercube, and this is the same for rMAEST plot. As we increase the dimensions, the performance of Sobol' is still the best at larger sample sizes, while the difference is smaller at $k = 19$ than at the case of $k = 10$ at very large sample size. for example $N = 8192$. According to the top left of Figure 2 and 3, Latin Hypercube is better than Random, but the difference is about the same across all sample sizes. As we expected, Random is the worst among all number of model runs and dimensions, but Latin Hypercube keeps the same rate of convergence as Random. At $k = 19$, Latin Hypercube's advantage over Random is decreasing at $N = 8192$, but much larger sample sizes are required to further confirm this result.

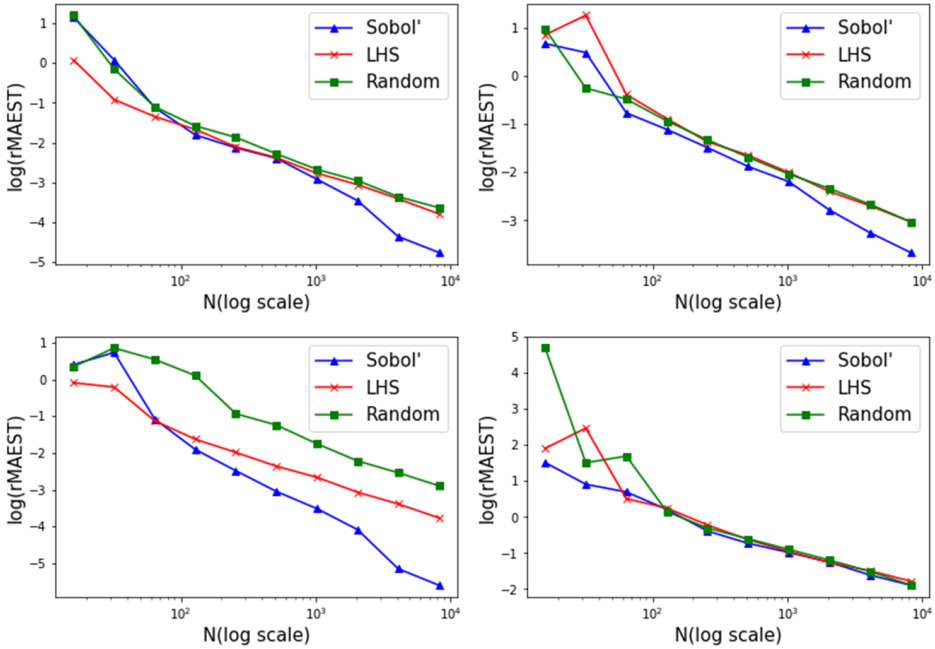

Figure 2. Comparison of rMAEST values for increasing N with log-log scale at $k = 10$. Top left is for type A1-1, top right is for type A2, bottom left is for type B, and bottom right is for type C.

Type A2. A1 has 2 important input variables, but for A2, we assign the first 5 input variables to be relatively important. Examining the rMAES plot of top right of Figure 1, we can tell that there are some random error spikes due to very small sample size for first-order sensitivity indices. With larger sample size, Sobol' provides less error for both first-order and total-effects, but the difference between Sobol' and Latin Hypercube becomes much smaller compared to type A1. On the other hand, Latin Hypercube has similar performance with Random started from $N = 128$, and we can observe this relation from top right of Figure 2 and 3. As we know that type A2 has stronger interaction between inputs than type A1, so the interaction may be one of the factors that affects the performance of sampling method, at least for Latin Hypercube in this type. We will further explore this assumption in the other types.

Type B. Type B has the weakest interaction between its inputs, since its coefficients are all zeros. According to rMAES plots of Figure 1, Latin Hypercube clearly has more advantage over Sobol' before $N = 128$ at 10 dimensions, and this advantage lasts till $N = 2048$ for 19 dimensions. Observing Figure 2 and 3, it is easily to get statistical error spikes before $N = 128$ for Sobol' and Random with 100 experiment runs; however, Latin Hypercube is surprisingly more stable than the other two sampling methods at even 19 dimensions on total-effects. Due to the error spikes, we increased the replicate to 400 for $N \leq 128$, and we got relatively reasonable result without error spikes. According to rMAEST plots, Latin Hypercube is way more superior than Random compared to the results in type A. With relatively weak interaction between inputs, Sobol' and Latin Hypercube has better performance than in type A, and Latin Hypercube is the best choice at sample size that is less than 64 for 10 dimensions or 128 for 19 dimensions among all three sampling methods.

Type C. Type C is to simulate a case where output variance is mostly attributed to the interaction between inputs. Through either rMAES or rMAEST plots, it is still hard to say which sampling method does best over a certain period of time. All three sampling methods have relatively large errors compared to previous types for either first-order or total-effects sensitivity indices. For 10 dimensions, Sobol' is slightly better than the other two sampling methods for small sample sizes; while for 19 dimensions, it is even harder to tell. With strong interaction, the influence on first-order and total-effects sensitivity indices of different sampling methods reduces. This result is more obvious at very large sample sizes where the relatively errors for three sampling methods are about the same.

'Jansen 1999' vs. 'Sobol' 2007'. As we mentioned in the section of variance-based sensitivity analysis, there are multiple attempts to approximate S_i and S_{Ti} in the history. In this section, we mainly talk about the

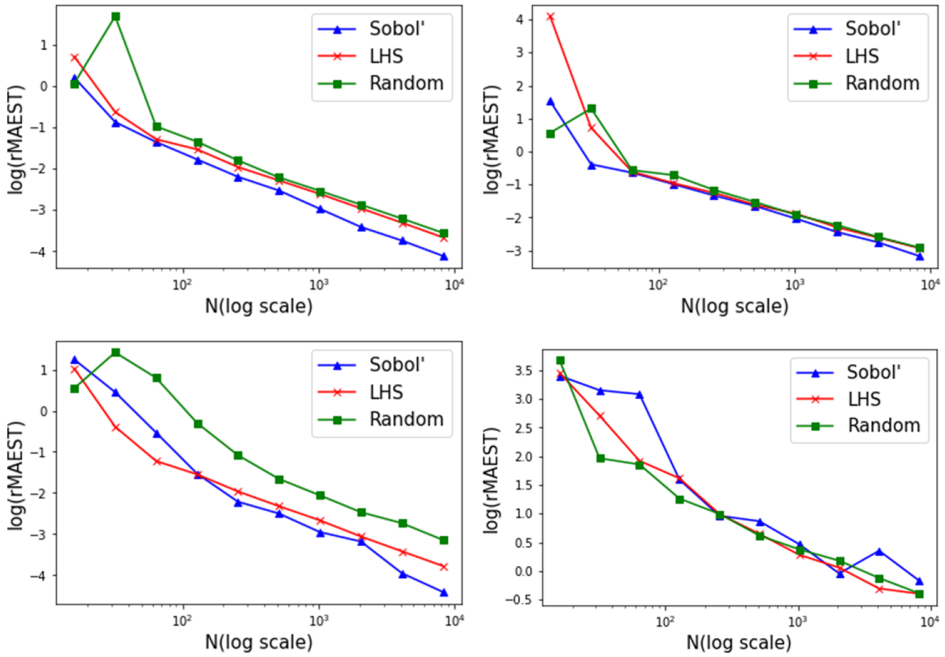

Figure 3. Comparison of rMAEST values for increasting N with log-log scale at $k = 19$. Top left is for type A1-1, top right is for type A2, bottom left is for type B, and bottom right is for type C.

estimators for S_{Ti}. Homma and Saltelli tried to estimate S_{Ti} first in 1996 Homma and Saltelli (1996)

$$E_{X_{\sim i}}(V_{X_i}(f|X_{\sim i})) = V(f) - \frac{1}{N}\sum_{j=1}^{N} f(A)_j f(A_B^{(i)})_j + f_0^2,$$

and this is called 'Homma 1996'. Later in Sobol's paper, he derived and rewrote 'Homma 1996' into a new form 'Sobol' 2007'

$$E_{X_{\sim i}}(V_{X_i}(f|X_{\sim i})) = \frac{1}{N}\sum_{j=1}^{N} f(A)_j (f(A)_j - f(A_B^{(i)})_j).$$

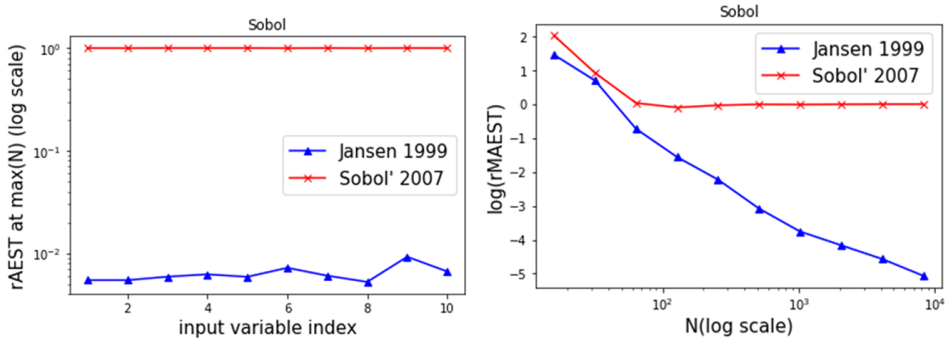

Figure 4. Comparison of 'Jansen 1999' and 'Sobol' 2007' by using type B and Sobol' sampling method. Left one is the AEST value at $N = 8192$, and the right one is log-log scale of MAEST value at increasing N.

Among all the S_{Ti} estimators, the most efficient one was proved to be 'Jansen 1999', which is the function (2) that we used in the previous section. Since 'Sobol' 2007' equivalents to 'Homma 1996', it was showed

that 'Jansen 1999' is also better than 'Sobol' 2007'. The proof can be found in Sobol' (2001). With the code we used above, we can use experiments to compare the efficiency between 'Sobol' 2007' and 'Jansen 1999'. For both estimators, We use Sobol' sampling method and Type B coefficients at 10 dimensions for relatively weak interaction, and we keep the test function the same. We compare the results of rAEST at $N = 8192$ and rMAEST plots to see how the different estimator affects the statistical errors. From Figure 4, we can clearly see that 'Jansen 1999' has far less statistical errors compared to 'Sobol' 2007'. In the log-log scaled rMAEST plot, 'Sobol' 2007' has a little difference with 'Jansen 1999' at small sampling size, but from $N = 64$ to $N = 8192$, the difference on error starts to increase dramatically. We also tested the difference of 'Sobol' 2007' and 'Jansen 1999' on other type of coefficients and sampling methods, and we got similar results. It is worth to notice that 'Sobol' 2007' has a relatively flat period from $N = 128$, and we will investigate further about the reason behind this result. Through these experiments, we confirmed that 'Jansen 1999' definitely performs better and has higher efficiency than 'Sobol' 2007'.

5 CONCLUSIONS

We extend the experiment of Tarantola et al. (2012), and we further compared the average of relative mean absolute error of each S_i and S_{Ti} between three sampling methods – Sobol', Latin Hypercube, and Random. We confirmed that Sobol' performs better than Latin Hypercube for very large sample sizes started from around $N = 256$ for most cases, but Latin Hypercube has some surprisingly good results at very small sample sizes compared to Sobol' and Random. We can see that Sobol' and Latin Hypercube are superior than Random for most cases. It is also reasonable to say that the interaction is a strong factor for the influence of different sampling method. The advantages of different sampling methods start to diminish when more than 50% of output variance are due to interaction, and it is viable to choose the method with fastest calculation speed under this situation. After the comparison between three sampling methods, we also confirmed that 'Jansen 1999' is more efficient than 'Sobol' 2007' through the same experiment environment.

We aim to bring Morris method into this comparison experiment as well, and we will compare the relative errors between all four methods. Since Tarantola et al. (2012) is based on global sensitivity analysis, we can also test the sampling methods for local sensitivity analysis. We used variance-based sensitivity analysis in this paper, but there are still many other sensitivity analysis methods that we can explore. We wish to use hydrology models as test functions, and our goal is to to find the most or relatively appropriate sensitivity analysis method/sampling method for different hydrology models.

REFERENCES

Antonov, I. A. and V. Saleev (1979). An economic method of computing lpτ-sequences. *USSR Computational Mathematics and Mathematical Physics 19*(1), 252–256.

Feinberg, J., L. H. (2015). Chaospy: An open source tool for designing methods of uncertainty quantification. *Journal of Computational Science 11*, 46–57.

Homma, T. and A. Saltelli (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety 52*(1), 1–17.

Jansen, M. J. (1999). Analysis of variance designs for model output. *Computer Physics Communications 117*(1-2), 35–43.

Kucherenko, S., B. Feil, N. Shah, and W. Mauntz (2011). The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety 96*(4), 440–449.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*(2), 239–245.

Saltelli, A., P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola (2010). Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications 181*(2), 259–270.

Sobol', I. M. (1969). Multidimensional quadrature formulas and haar functions [in russian].

Sobol', I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments 1*(4), 407–414.

Sobol', I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation 55*(1), 271–280.

Sobol', I. M. (2007). Global sensitivity analysis indices for the investigation of nonlinear mathematical models [in russian]. *Matematicheskoe Modelirovanie 19*(11), 23–24.

Tarantola, S., W. Becker, and D. Zeitz (2012). A comparison of two sampling methods for global sensitivity analysis. *Computer Physics Communications 183*(5), 1061–1072.