Toward a more robust variance-based global sensitivity analysis of model outputs

C. Tong

October 16, 2007

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Toward a More Robust Variance-based Global Sensitivity Analysis of Model Outputs *

Charles Tong

Lawrence Livermore National Laboratory

Livermore, CA 94550-0808

Abstract

Global sensitivity analysis (GSA) measures the variation of a model output as a function of the variations of the model inputs given their ranges. In this paper we consider variance-based GSA methods that do not rely on certain assumptions about the model structure such as linearity or monotonicity. These variance-based methods decompose the output variance into terms of increasing dimensionality called "sensitivity indices", first introduced by Sobol' [25]. Sobol' developed a method of estimating these sensitivity indices using Monte Carlo simulations. McKay [13] proposed an efficient method using replicated Latin hypercube sampling to compute the "correlation ratios" or "main effects", which have been shown to be equivalent to Sobol's first-order sensitivity indices. Practical issues with using these variance estimators are how to choose adequate sample sizes and how to assess the accuracy of the results. This paper proposes a modified McKay main effect method featuring an adaptive procedure for accuracy assessment and improvement. We also extend our adaptive technique to the computation of second-order sensitivity indices. Details of the proposed adaptive procedure as wells as numerical results are included in this paper.

1   Introduction

Sensitivity analysis (SA) studies how variations of a model output describing certain (for example, physical, biological, or social) processes can be accounted for by variations in the control or model parameters (collectively called input factors or input parameters). In the

context of the present discussion, we restrict ourselves to sensitivity analysis of deterministic simulation models, which give identical results when presented with the same set of parameter values. Sensitivity analysis is increasingly recognized as an important tool for model building and validation. In general, sensitivity analysis is useful for all processes where it is important to know which input factors mostly contribute to output variability.

Sensitivity analysis methods are generally classified as either local or global. Local SA methods compute or approximate the partial derivatives of model outputs with respect to individual input factors at some nominal settings. Global SA, on the other hand, studies the effects of input variations on model outputs in the entire allowable ranges of the input space. Saltelli *et al.* [24, 27] have defined global SA methods by two properties:

1. The inclusion of influence of scales and shapes of the probability density functions for all inputs; and

2. The sensitivity estimates of individual inputs are evaluated while varying all other inputs (multi-dimensional averaging).

In this paper we are primarily concerned with global SA methods which can generally be decomposed into four steps:

1. Define credible ranges and distributions of input factors,

2. Create a sample of input factor values,

3. Evaluate the model for each sample point, and

4. Estimate the effect of each input factor on the model output.

Global SA methods can further be classified as either qualitative or quantitative. For applications with large number of input factors (tens to hundreds), the "curse of dimensionality" dictates that the computational cost for quantitative global SA becomes insurmountable. The purpose of qualitative SA studies is to identify (as opposed to quantify) the most important input factors using a relatively inexpensive set of simulation experiments, a process called "parameter screening". The goal is to enable the quantitative SA studies to focus on the smaller subset of most important input factors.

Quantitative SA methods, which apportion the output variability to individual input variabilities, typically require large number of simulation runs. When simulation models themselves are computationally intensive, the computational cost of quantitative SA may become prohibitive. To make quantitative SA more tractable, response surface modeling (not within the scope of this paper) is often used to construct inexpensive surrogates in place of the original simulation models.

Among the quantitative SA methods, variance-based methods have received the most attention. The main idea of the variance-based methods is to evaluate the variance components for all of the individual or groups of input factors. By decomposing the model function

into a sum of elementary functions, Sobol' [25] has shown that a decomposition of the model output variance is possible (for independent input factors). These variance components are called Sobol' indices, and can be used for any complex model functions. When the model is purely linear, the Sobol' indices are equivalent to the standardized regression coefficient in classical analysis. For models with K inputs, the number of Sobol' indices is $2^K - 1$. In practice, only the first and second-order Sobol indices are estimated. For large K, Homma and Saltelli [6] proposed the the "total sensitivity indices" which can be computed by using Monte Carlo simulations or the extended Fourier Amplitude Sampling Test (FAST) method.

This paper focuses on efficient and accurate methods for computing the first- and second-order sensitivity indices. Specifically, McKay's [13] main effect analysis is an efficient method for computing the first-order sensitivity indices. However, a difficulty when applying this method is the determination of a suitable sample size to achieve sufficient accuracy. One often resorts to very large samples to ensure sufficient accuracy. Here, we propose an improved McKay main effect analysis with an adaptive accuracy assessment and improvement capability. We also propose an efficient method for computing the second-order sensitivity indices using replicated orthogonal arrays and the corresponding second-order sensitivity indices. Again, an adaptive refinement technique is used to facilitate accuracy assessment and improvement.

In Section 2 we provide a brief introduction to variance-based sensitivity analyses. Section 3 gives details of McKay's main effect analysis. Section 4 proposes improvements to McKay's method for accuracy assessment and improvement. Section 5 presents an efficient method based on replicated orthogonal arrays for computing second-order sensitivity indices. Section 6 describes an adaptive strategy similar to the improved main effect analysis for computing the second-order sensitivity indices. Numerical results are interspersed in Section 4 and 6. Finally, a brief summary will be given in Section 7.

2   Variance-based Sensitivity Measures

Let $Y = F(X)$ be a mathematical model that maps a set of K input parameters $X \in \Re^K$ to a scalar output Y. Let $E(Y)$ and $V(Y)$ denote the mean and variance of the distribution of Y given probability distributions of X. A sensitivity measure for a given input X_i can be obtained by assuming a complete knowledge of the true value of X_i and assessing the effect of this knowledge on the output variance. To do this, we fix X_i at $X_i = X_i^*$ and compute the corresponding conditional variance $V(Y|X_i = X_i^*)$. Since this complete knowledge of X_i^* is in general not available, we compute, $E(V(Y|X_i))$, which is the average of the conditional variances given the probability distribution of X_i. Intuitively, $E(V(Y|X_i))$ measures the variance of Y when X_i is known, and so $V(Y) - E(V(Y|X_i))$ (the added variance due to the variability of X_i) is a reasonable indicator to quantify the importance of input X_i. This indicator is equivalent to the statistical quantity called *variance of conditional expectation*

(or VCE) via the following variance decomposition property:

$$V(Y) = V(E(Y|X_i)) + E(V(Y|X_i)). \tag{1}$$

The first term on the right hand side of this relation is the variance of conditional expectation (VCE), conditioned on X_i; and the second term is the error or residual term. Here $V(E(Y|X_i))$ measures the variability in the conditional expected value of Y as the input X_i takes on different values. The residual term represents the variability in Y not accounted for by the input X_i.

McKay defined the correlation ratio [13] (or main effect) by normalizing the VCE's with $V(Y)$:

$$\eta^2(X_i) = \frac{V(E(Y|X_i))}{V(Y)}. \tag{2}$$

A high correlation ratio implies that X_i is important in influencing the output variability. If all input factors are uncorrelated and there are no multi-way interactions, the sum of the correlation ratios is 1.

In [25], Sobol' derived a first-order sensitivity index and his derivation is based on the decomposition of $Y = F(X)$ into a sum of terms of increasing dimensionality:

$$F(X_1, X_2, \cdots, X_K) = F_0 + \sum_i F_i(X_i) + \sum_{i<j} F_{ij}(X_i, X_j) + \cdots + F_{12\cdots,K}(X_1, X_2, \cdots, X_K) \tag{3}$$

where the integral of every term over any of its own input variables is zero. Sobol' showed that, when all inputs are orthogonal to each other, this decomposition is unique and that $V(Y)$ is the sum of the variances of each term in the decomposition:

$$V(Y) = \sum_i V_i + \sum_{i<j} V_{ij} + \sum_{i<j<k} V_{ijk} + \cdots + V_{12\cdots K} \tag{4}$$

where V_i is the variance of F_i, V_{ij} is the variance of F_{ij}, and so on. The total number of terms for K inputs is thus $2^K - 1$. The V_i's can be shown to be equivalent to McKay's correlation ratios by the following relationship:

$$V_i = V(Y)\eta^2(X_i) = V(E(Y|X_i)).$$

Similarly, V_{ij}'s are the (pure) two-way interactions such that

$$V_{ij} = V(E(Y|X_i, X_j)) - V(E(Y|X_i)) - V(E(Y|X_j)).$$

In the event that the inputs are correlated, the above relationships no longer hold. However, variance-based measures are still useful sensitivity indicators. Input correlation will not be covered in this paper.

4

3 Main Effect Analysis

Main effects (or sensitivity indices) can be computed by directly evaluating the K integrals for the K inputs. McKay [13] proposed a more efficient estimation method based on the use of a single replicated Latin hypercube sampling (r-LHS) design for all K inputs. It should be noted that even with this efficiency improvement the main effect analysis is still very expensive requiring a substantial number (for example, thousands) of function evaluations. For models that are themselves expensive to evaluate, a common strategy to make main effect analysis feasible is to first create a response surface model (also called surrogate model, meta-model, or emulator) and perform subsequent analyses on the substantially less expensive approximate model.

In the r-LHS design, each X_i takes on distinct values $X_{ij}, j = 1, \cdots, S$ where S is the number of levels (or symbols). These values are to be replicated R times in total so that the final design has $N = SR$ sample points.

Based on this design, the mean and variance of Y can be estimated by, for any i in $\{1, \cdots, K\}$,

$$\bar{Y} = \frac{1}{SR} \sum_{j=1}^{S} \sum_{r=1}^{R} Y^{(r)}(X_i = X_{ij}), \tag{5}$$

and

$$V(Y) = \frac{1}{SR} \sum_{j=1}^{S} \sum_{r=1}^{R} \left[Y^{(r)}(X_i = X_{ij}) - \bar{Y} \right]^2, \tag{6}$$

respectively, where $Y^{(r)}(X_i = X_{ij})$ is the output corresponding to $X_i = X_{ij}$ in the r-th replication. (that is, the R replications amount to keeping input i at some fixed value and varying all others). The estimator of the conditional expectation for $X_i = X_{ij}$ is given by

$$\bar{Y}(X_i = X_{ij}) = \frac{1}{R} \sum_{r=1}^{R} Y^{(r)}(X_i = X_{ij}) \tag{7}$$

Finally, the formula for the variance of conditional expectation (VCE) is given by:

$$\text{VCE}(X_i) = \frac{1}{S} \sum_{j=1}^{S} \left[\bar{Y}(X_i = X_{ij}) - \bar{Y} \right]^2 - \frac{1}{SR^2} \sum_{j=1}^{S} \sum_{r=1}^{R} \left[Y^{(r)}(X_i = X_{ij}) - \bar{Y}(X_i = X_{ij}) \right]^2, \tag{8}$$

and the correlation ratio for input i can be computed by normalizing $\text{VCE}(X_i)$ with the output variance. A variant of the VCE is the biased VCE which is defined as:

$$\text{VCE}_b(X_i) = \frac{1}{S} \sum_{j=1}^{S} \left[\bar{Y}(X_i = X_{ij}) - \bar{Y} \right]^2. \tag{9}$$

The correlation ratio is a useful estimator for input importance for general models. In addition, the scatter plots (the K plots, each with respect to individual inputs) from the replicated Latin hypercube samples provide useful visual information on how the output
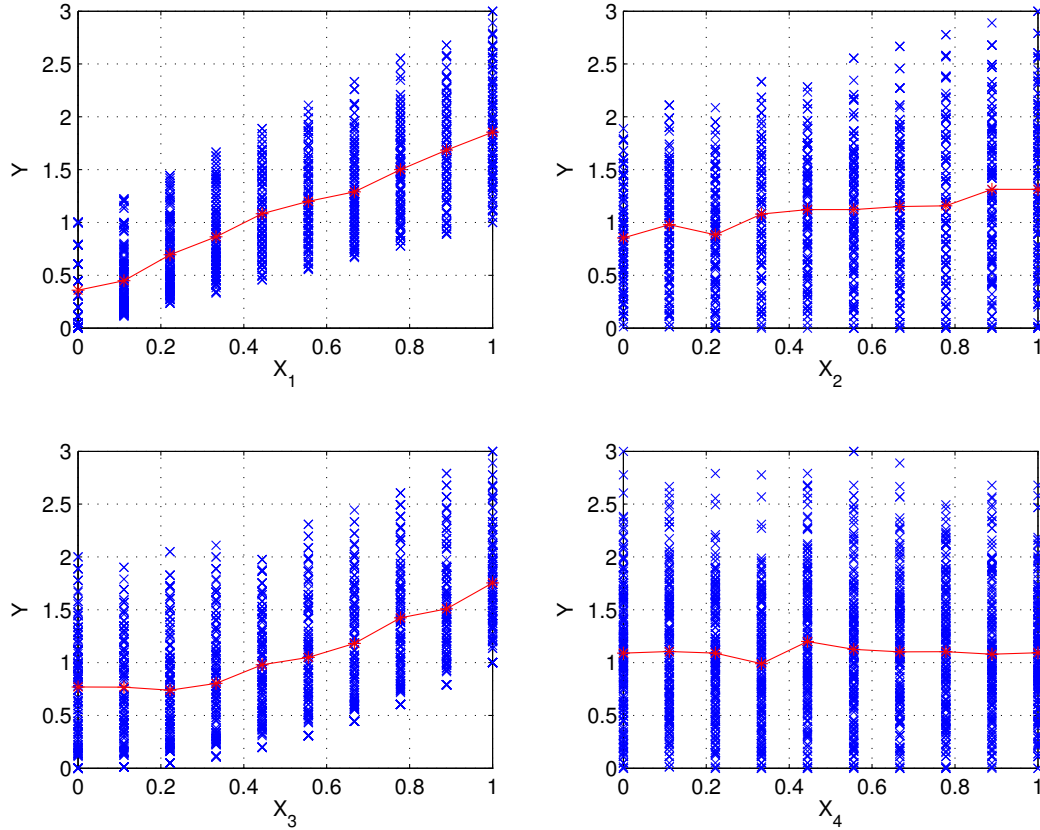
5

Figure 1: An example scatter plot from replicated Latin hypercube sampling

behaves as X_i takes on different values X_{ij} (for example, by inspecting the line joining the means of all levels, namely, X_{ij}'s in the X_i scatter plot.) In addition, when R is sufficiently large (say, > 50), the parameter space is sufficiently explored so that the envelops encompassing the output data in the scatter plots give additional qualitative information about parameter interactions (that X_i is interacting with some other inputs), although they offer no additional information on which other inputs X_i interacts with (It requires two-way interaction analysis described in Section 4 to quantify pair-wise interactions). An example is given in Figure 1 where the function is: $Y = X_1 + X_1 * X_2 + X_3^2$ with four inputs $X_i, i = 1, 2, 3, 4; X_i \in [0, 1]$. Here, we observe in the scatter plots for X_1 and X_2 that the envelopes enclosing the data points are not uniform, indicating that these two input factors have interactions with other inputs. This observation agrees well with the example function. Furthermore, we observe that X_3 is nonlinear and X_4 has negligible effect on the output.

4 An Improved Main Effect Analysis

To create a replicated Latin hypercube sample, both S (number of levels) and R (number of replications) have to be specified (such that $N = SR$) by users. In [14], McKay investi-

gated the variability of correlation ratio estimates as a function of sampling variability and concluded that sufficiency of the sampling design (specifically, S and R) is very important to achieve the desired precision. Specifically, large N may be needed to adequately estimate the correlation ratios. In addition, if the biased correlation ratio estimator is used, large bias may result when R is small. Saltelli *et al.* [24] recommended that S should be larger than R to give good accuracy. Despite this recommendation, it should be noted that the adequacy of a sampling design is model dependent and thus not generally known a-priori. In this section we propose a more robust main effect analysis to address this issue.

Our improved main effect analysis is based on an iterative procedure consisting of an adaptive sampling scheme and an accuracy assessment tool to monitor the convergence of the correlation ratios. Our adaptive sampling scheme borrows from our earlier work on refinement of stratified designs [32]. Our improved method currently considers only adaptively increasing S (by a factor of 2 per refinement) for accuracy improvement while keeping R fixed. To offset the effect of bias [14], we use a moderate sized R and also the unbiased correlation ratio estimator.

In the rest of this section, we first show how to adaptively refine a replicated Latin hypercube design. We will then describe the iterative procedure utilizing this adaptive sampling scheme. A few examples will be given to study the effectiveness of this improved method.

4.1 Refinement for Replicated Latin Hypercube

We first denote a replicated Latin hypercube by an 3-tuple LH(N,K,S) where N, K and S are the sample size, number of input parameters, and number of symbols or levels, respectively. The number of replications can be recovered by $R = N/S$. We begin with a fixed R (for example, $R = 50$) and an initial S (for example, $S = 4$). The basic idea in the refinement algorithm follows two major steps. The first step involves refining each grid cell (in a K-dimensional grid with S partitions in each dimension) into an 2^K subgrid. Then for each cell that already contains a sample point, a LH($2, K, 2$) (with size 2, K inputs, and 2 levels) containing the existing sample point is created for the grid cell. The refined sample can be shown to preserve its property as a replicated Latin hypercube. A selective random permutation is then applied to the newly created sample points to improve the statistical property of the entire refined sample while leaving the original sample points unchanged. The detailed refinement algorithm (*Algorithm RefineLH*) consists of the following steps (given an initial replicated LH sample matrix Z):

Pattern reconstruction: Transform the sample matrix Z (an $N \times K$ matrix) to the corresponding LH pattern matrix A by (S is the current number of levels and R is the number of replications)

$$A(i,j) = \lceil (Z(i,j) - L_j)/\delta \hat{X}_j) \rceil, \quad i = 1, \cdots, N; j = 1, \cdots, K,$$

where L_j and U_j are the lower and upper bound of input j, and $\delta \hat{X}_j = (U_j - L_j)/S$.

Replication separation: Partition A into R individual LH pattern matrices $A_m, m = 1, \cdots, R$ (each A_m is an $S \times K$ matrix). Then, for each A_m,

Level refinement: Form another pattern matrix B_m (called base pattern matrix) from A_m by

$$B_m(i, j) = (\lceil A_m(i, j) \rceil - 1) * 2.$$

New sample insertion: Create the new pattern matrix $\tilde{A}_m$: for each row i of B,

1. Form a new LH pattern matrix C_i of size $2 \times K$.

2. Set $C_i \leftarrow C_i + [1 \quad 1]^T B_m(i)$,

3. Permute C_i to have one row matching $A_m(i)$ (by first exchanging entries of row 1 of C_i with entries in the same column so that row 1 matches $A_m(i)$).

4. Load $\tilde{A}_m$ row $2 \times (i - 1) + 1$ to row $2 \times i$ with C_i.

Sample randomization: Perform random permutation to each column of $\tilde{A}_m$ but only to the newly created rows.

Sample concatenation: Append all $\tilde{A}_m, m = 1, \cdots, R$ matrices to form the final $\tilde{A}$ pattern matrix.

Sample Generation: Map the pattern matrix (which has number of levels $= 2S$ now) to the new sample matrix $\tilde{Z}$ by scaling and translation with respect to the input ranges.

$$\tilde{Z}(i, j) = \tilde{A}(i, j) * \delta X_j + L_j + \epsilon(i, j)$$

where $\epsilon(i, j)$ is a small random perturbation and its value depends on $\tilde{A}(i, j)$ to preserve the replicated LH property.

An example of refining a LH sample is given in [32].

4.2   An Adaptive Algorithm for Main Effect Analysis

The refinement technique can be used in an iterative procedure to improve the accuracy of main effect analysis. The algorithm is as follow:

1. Select an initial replicated LH sample with sample size $N_0 = S_0 R$. Prescribe a precision $0 < \epsilon < 1$. Set Iteration $= 0$.

2. Set Iteration $=$ Iteration $+ 1$. Then evaluate the model using the current sample.

3. Use the sample inputs and outputs to compute the VCE's.

4. If Iteration > 1, do the following: for each VCE(X_i), compute the error e_i by finding the difference between the current and the last VCE(X_i); else set $e_i = \epsilon$.

5. If $\max e_i < \epsilon$, terminate.

6. *Apply the Refine algorithm* to create the refined LH sample. Then go to step 2.

An alternative termination criterion for this procedure can be a prescribed maximum number of model evaluations. Using this criterion, the main effect analysis should give not only the $\text{VCE}(X_i)$'s, but also the estimated error bounds.

4.3   Numerical Results

In this section we demonstrate the effectiveness of our modified main effect algorithm on two test examples- one monotonic and one non-monotonic functions.

4.3.1   A Monotonic Test Problem

The first test problem is the monotonic Sobol' function [24] given by:

$$Y = \exp\left(\sum_{j=1}^{6} b_j X_j\right) - I_6 \tag{10}$$

where $b_1 = 1.5, b_2 = b_3 = b_4 = b_5 = b_6 = 0.9$,

$$I_6 = \prod_{j=1}^{6} \frac{e^{b_j} - 1}{b_j},$$

and X_j is uniformly distributed in $[0, 1]$. The true correlation ratios for X_1 is 0.287 and 0.1057 for $X_j, j = 2, \cdots, 6$.

We simulate our iterative algorithm 100 times, each with an initial S of 4 and $R = 50$. Figure 2 shows the convergence history of the 6 correlation ratios as a function of S. Due to the randomness in the initial LH design and subsequent refinements, each of the 100 simulations goes through a different convergence path. The blue 'x' in the plots are actual correlation ratios computed at different refinment levels. We observe firstly from the plots that all simulations exhibit similar paths converging to the true values as S is increased through refinement. In general, the spread of the correlation ratios shrinks as S is increased, demonstrating that larger sample sizes increase the confidence of the estimations. The reason that some envelopes expand a little initially is that the many sample point duplications due to the initial number of levels being too small ($S = 4$) limit the spread of the results.

4.3.2   A Non-monotonic Test Problem

The second test problem is the Ishigami function [24]:

$$Y = sin(X_1) + 7sin^2(X_2) + 0.1X_3^4 sin(X_1) \quad X_i \in [-\pi, \pi], i = 1, 2, 3 \tag{11}$$
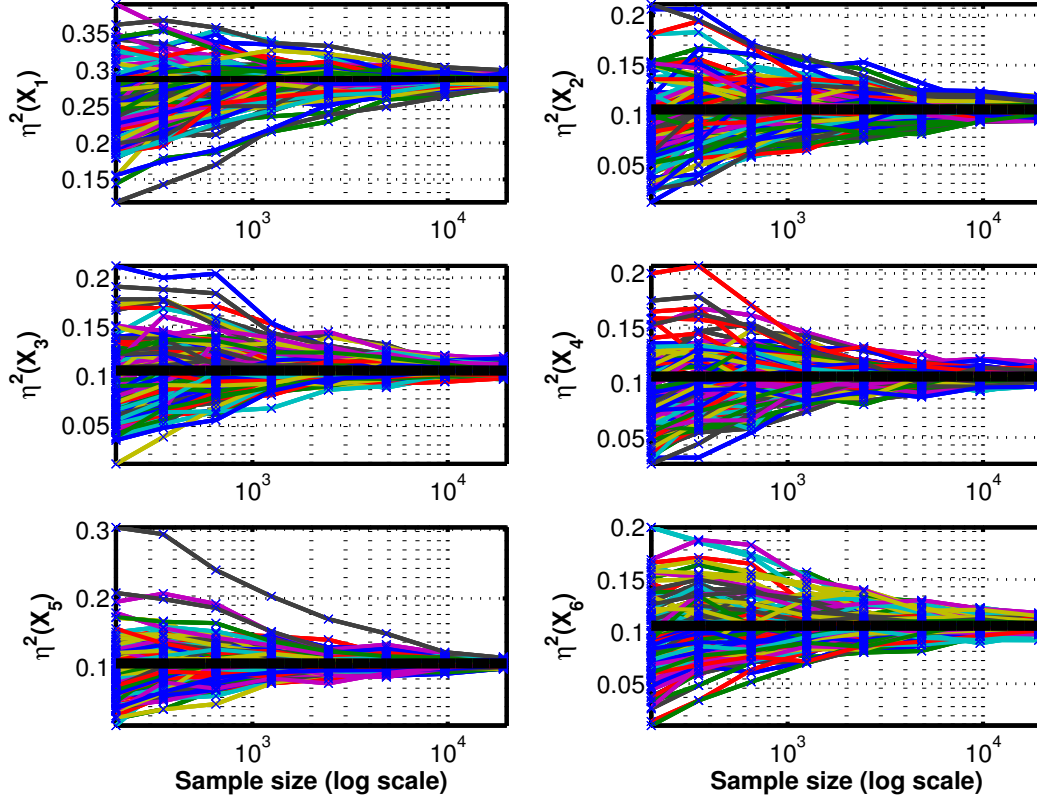
9

Figure 2: Sobol' function: convergence history for the η^2's (black horizontal lines- true values)

which has the following statistics

$$\bar{Y} = 3.5; \quad V(Y) = \pi^4/50 + \pi^8/1800 + 1/2 + 49/8 \approx 13.8445$$

$$\eta(X_1) = 0.3139; \quad \eta(X_2) = 0.4424; \quad \eta(X_3) = 0.0$$

Again, We simulate our iterative algorithm 100 times, each with an initial S of 4 and $R = 50$. Figure 3 shows the convergence history of the 3 correlation ratios as a function of S. Again, we observe that the correlation ratios converge to their true values as S is increased through refinement. We again observe that in general the spread of the correlation ratios in general shrinks as S is increased.

5   Two-way Interaction Analysis

In this section we extend the idea for main effect analysis to two-way interaction studies for uncorrelated inputs. In this case, we employ the following relationship
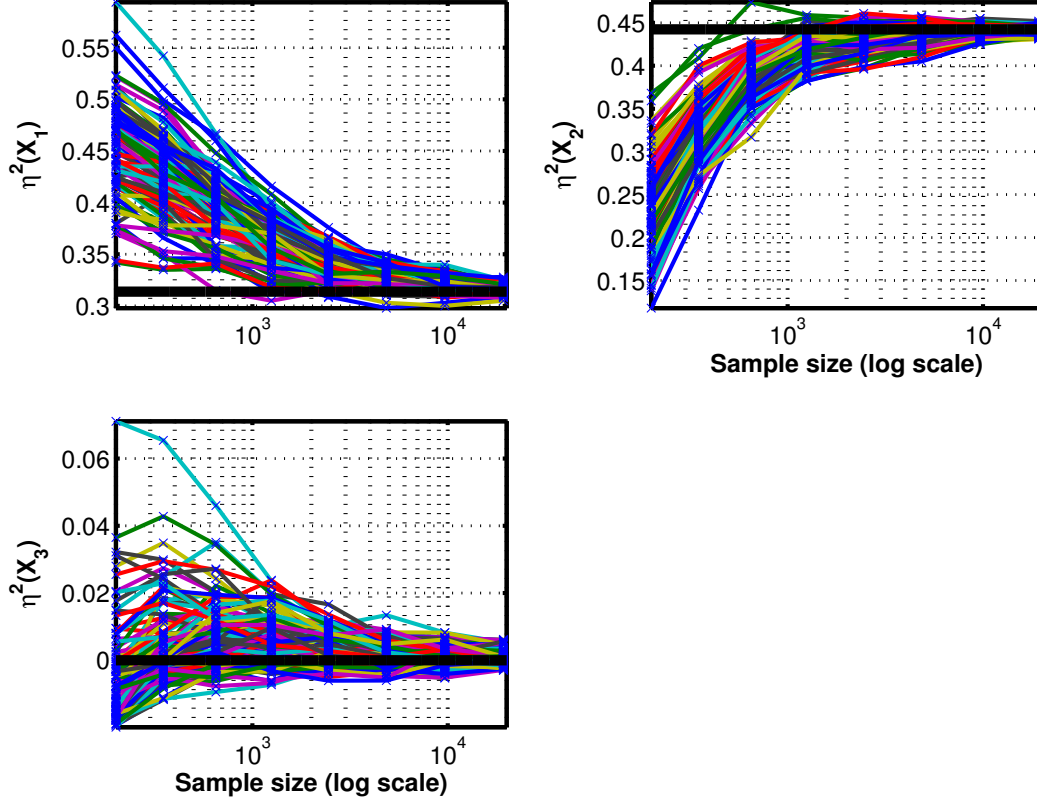
$$V(Y) = V(E(Y|X_i, X_k)) + E(V(Y|X_i, X_k)) \tag{12}$$

10

Figure 3: Ishigami function: convergence history for the η^2's (black horizontal lines- true values)

where X_i and X_k are two distinct inputs under consideration. The first term on the right hand side is the variance of the conditional expectation $\text{VCE}(X_i, X_k)$ of Y, conditioned on X_i and X_k. Again, the second term is the error or residual term measuring the estimated variance of Y by fixing X_i and X_k. In addition, the correlation ratio for the input pair (X_i, X_k) is

$$\eta^2(X_i, X_k) = V(E(Y|X_i, X_k))/V(Y). \tag{13}$$

A high correlation ratio shows that X_i and X_k taken together are important contributors to the output variability. The variance due to the interaction term alone is defined as

$$V(X_i, X_k) = V(E(Y|X_i, X_k)) - V(E(Y|X_i)) - V(E(Y|X_k)). \tag{14}$$

$V(X_i, X_k)$ can be computed using many different techniques, for example, by directly evaluating the corresponding integral. Here we illustrate its evaluation with the use of replicated orthogonal array sampling. Using orthogonal array design with a strength of 2, X_i and X_k take on values $X_{ij}, j = 1, \cdots, S$ and $X_{kl}, l = 1, \cdots, S$ where S is the number of symbols (or levels). Based on this design, the mean and variance of Y can be estimated by,

11

for any i and k in $\{1, \cdots, K\}, \quad i \neq k,$

$$\bar{Y} = \frac{1}{S^2 R} \sum_{j=1}^{s} \sum_{l=1}^{S} \sum_{r=1}^{R} Y^{(r)}(X_i = X_{ij}, X_k = X_{kl})), \tag{15}$$

and

$$V(Y) = \frac{1}{SR} \sum_{j=1}^{S} \sum_{l=1}^{S} \sum_{r=1}^{R} \left[Y^{(r)}(X_i = X_{ij}, X_k = X_{kl}) - \bar{Y} \right]^2, \tag{16}$$

where $Y^{(r)}(X_i = X_{ij}, X_k = X_{kl})$ is the output corresponding to $X_i = X_{ij}$ and $X_k = X_{kl}$ in the r-th replication (that is, keeping the two inputs at some fixed values and varying all others). The variance estimator for the expectation conditioned on $X_i = X_{ij}$ and $X_k = X_{kl}$ is

$$\bar{Y}(X_i = X_{ij}, X_k = X_{kl}) = \frac{1}{R} \sum_{r=1}^{R} Y^{(r)}(X_i = X_{ij}, X_k = X_{kl}) \tag{17}$$

To approximate the variance of conditional expectation $\text{VCE}(X_i, X_k)$, we use

$$\text{VCE}(X_i, X_k) = \frac{1}{S^2} \sum_{j=1}^{S} \sum_{l=1}^{S} \left[\bar{Y}(X_i = X_{ij}, X_k = X_{kl}) - \bar{Y} \right]^2 - \tag{18}$$

$$\frac{1}{S^2 R^2} \sum_{j=1}^{S} \sum_{l=1}^{S} \sum_{r=1}^{R} \left[Y^{(r)}(X_i = X_{ij}, X_k = X_{kl}) - \bar{Y}(X_i = X_{ij}, X_k = X_{kl}) \right]^2,$$

and the two-way correlation ratio for input pair (i, k) is obtained by normalizing $\text{VCE}(X_i, X_k)$ with the output variance $V(Y)$. Again, we can also compute the corresponding biased estimator by ignoring the second term in the above equation.

Finally, we arrive at the following pure two-way interaction effect

$$V(X_i, X_k) = \text{VCE}(X_i, X_k) - \text{VCE}(X_i) - \text{VCE}(X_k) \tag{19}$$

where $\text{VCE}(X_i)$ and $\text{VCE}(X_i)$ can be obtained from the main effect analysis.

This same idea can be applied to the analysis of higher order interaction. For example, to analyze 3-way interaction, a replicated orthogonal array design of strength 3 can be used together with the corresponding formulas for computing the variance of conditional expectations.

5.1 An Improved Two-way Interaction Analysis

Our improved two-way interaction analysis is based on an iterative procedure consisting of an adaptive orthogonal array sampling scheme (based on our earlier work in [32]) and an accuracy assessment tool (similar to the one in our improved main effect analysis) to monitor the convergence of the correlation ratios. As opposed to replicated Latin hypercube designs which have a sample size growth factor of ≈ 2 per refinement, the sample size growth factor

for orthogonal arrays is $O(K^2)$. Therefore, our improved procedure is less practical than the improved main effect analysis for large K (for example, $K > 5$).

In the rest of this section, we first present the refinement algorithm for orthogonal arrays. We will then describe how to embed this refinement algorithm in the iterative procedure. A few examples will be given to study the effectiveness of our improved method.

5.2 Refinement for Replicated Orthogonal Arrays

We first denote a replicated orthogonal array by an 4-tuple OA(N,K,S,t) where N, K, S and t are the sample size, number of parameters, number of symbols or levels, and strength, respectively. The number of replications can be recovered by $R = N/(S^2)$. We begin with a fixed R (for example, $R = 50$) and an initial S (the minimum S depends on K). The basic idea in the refinement algorithm is similar to that of the Latin hypercube and it consists of the following two steps: (1) refine each grid cell (in a K-dimensional grid with S partitions in each dimension) into an S^K subgrid; and (2) for each grid cell that already contains a sample point, an OA(S^2, K, S, t) including the existing sample point is created. The refined sample can be shown to preserve its property as a replicated orthogonal array. A selective random permutation is then applied to the newly created sample to improve the statistical property of the refined sample while leaving the original sample points unchanged. The refinement algorithm (*Algorithm RefineOA*) consists of the following steps:

Pattern reconstruction: same as in *Algorithm RefineLH*.

Replication separation: same as in *Algorithm RefineLH*.

Level refinement: For each pattern matrix $A_m, m = 1, \cdots, R$, for another pattern matrix B_m (called base pattern matrix) from A_m by

$$B_m(i, j) = (\lceil A_m(i, j) \rceil - 1) * 2.$$

New sample insertion: Create the new pattern matrix $\tilde{A}_m$: for each row i of B,

1. Form a new OA pattern matrix C_i with OA($S^2, K, S, 2$).
2. Set $C_i \leftarrow C_i + [1 \quad 1]^T B_m(i)$,
3. Permute C_i to have one row matching $A_m(i)$ (by first exchanging entries of row 1 of C_i with entries in the same column so that row 1 matches $A_m(i)$).
4. Load $\tilde{A}_m$ row $S^2 \times (i - 1) + 1$ to row $S^2 \times i$ with C_i.

Sample randomization: same as in *Algorithm RefineLH*.

Sample concatenation: same as in *Algorithm RefineLH*.

Sample Generation: same as in *Algorithm RefineLH*.

An example of refining an OA sample is given in [32].

5.3 An Adaptive Algorithm for Two-way Interaction Analysis

The OA refinement technique can be used in an iterative procedure to improve the accuracy of interaction analysis. The algorithm is as follow:

1. Select an initial replicated OA sample with sample size $N_0 = S_0^2 R$. Prescribe a precision $0 < \epsilon < 1$. Set Iteration $= 0$.

2. Set Iteration $=$ Iteration $+ 1$. Then evaluate the model using the current sample.

3. Use the sample inputs and outputs to compute the VCE's.

4. If Iteration > 1, do the following: for each $\text{VCE}(X_i, X_k)$, compute the error e_{ik} by finding the difference between the current and the last $\text{VCE}(X_i, X_k)$; else set $e_{ik} = \epsilon..$

5. If $\max e_{ik} < \epsilon$, terminate.

6. *Apply the Refine algorithm* to create a refined OA sample. Then go to step 2.

5.4 Numerical Results

In this section we demonstrate the effectiveness of our modified interaction analysis on two test examples- one monotonic and one non-monotonic functions.

5.4.1 A Monotonic Test Problem

The first test problem is the following polynomial function given by:

$$Y = X_1 + X_1 X_2 + X_3 X_4^3 \tag{20}$$

where X_j is uniformly distributed in $[0, 2]$.

We simulate our iterative algorithm 100 times, each with an initial S of 2 and $R = 50$. Figure 4 shows the convergence history of the 3 two-parameter correlation ratios as a function of $N = S^2 R$. Again, because of the randomness in the initial orthogonal array design and subsequent refinements, each of the 100 simulations goes through a different convergence path. We again observe that the correlation ratios of all 100 simulations converge to their true values as S is increased through refinement. In addition, the spread of the correlation ratios shrinks as S is increased, showing again that larger sample sizes increase the confidence of the estimations.

5.4.2 A Non-monotonic Test Problem

The second test problem is the Ishigami function [24]:

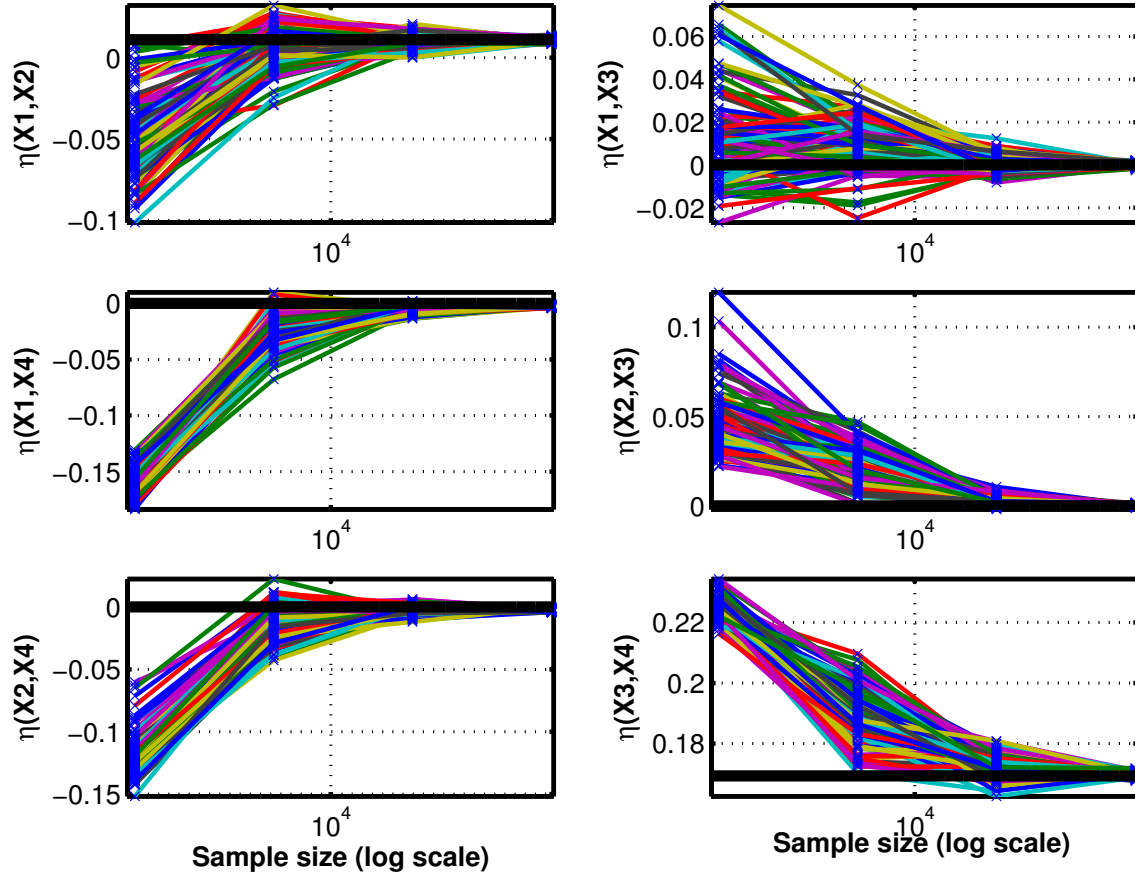$$Y = sin(X_1) + 7sin^2(X_2) + 0.1X_3^4 sin(X_1) \quad X_i \in [-\pi, \pi], i = 1, 2, 3$$

14

Figure 4: Polynomial function: convergence history for the η^2's (black horizontal lines- true values)

Once again we simulate our iterative algorithm 100 times, each with an initial S of 2 and $R = 50$. Figure 5 shows the convergence history of the 3 two-way correlation ratios as a function of S. Again, the same trends are observed as before.

6   Summary

In this paper we propose robust first- and second-order variance-based methods for global sensitivity analysis. Specifically, the use of refinement techniques in stratified sampling methods such as Latin hypercube and orthogonal array together with the corresponding analyses has enabled the accuracy assessment and improvement of the correlation ratios. We have demonstrated the effectiveness of these methods through a few numerical examples.
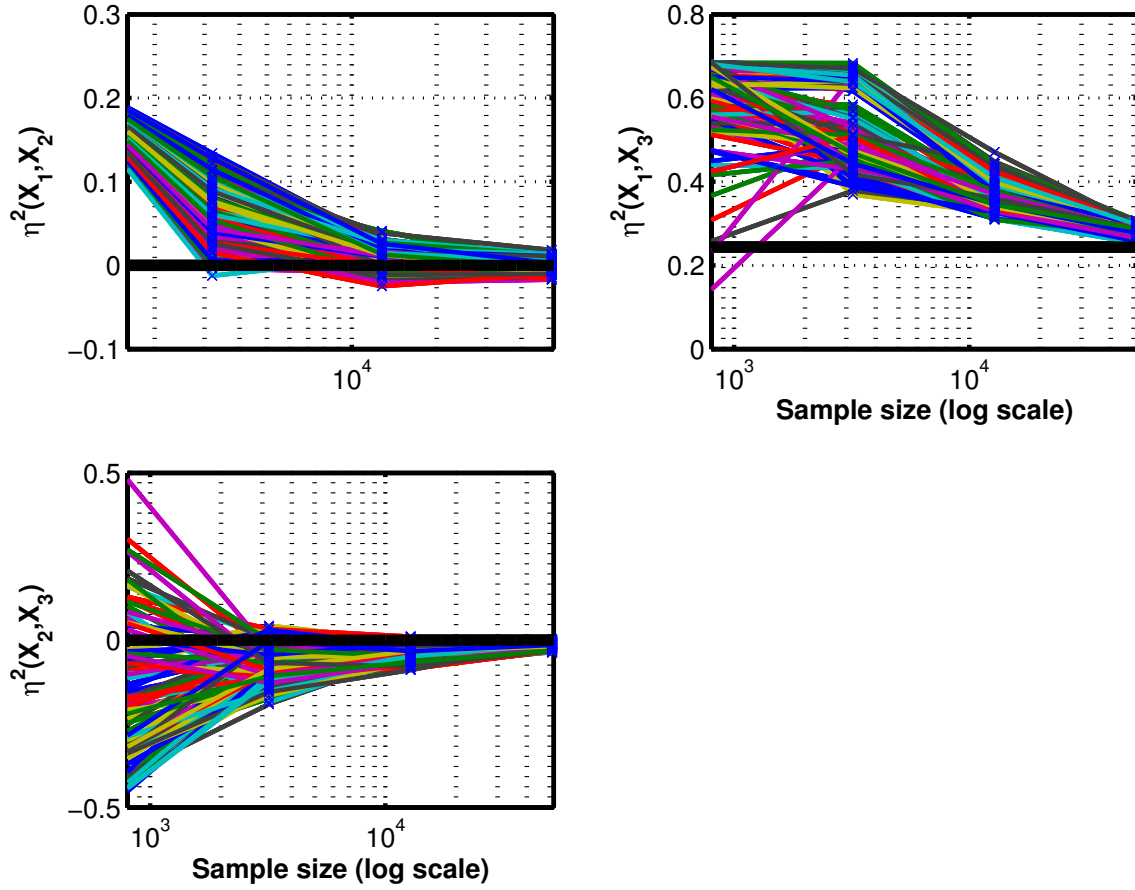
Figure 5: Ishigami function: convergence history for the η^2's (black horizontal lines- true values)

References

[1] F. Campolongo and R. Braddock, *The use of graph theory in the sensitivity analysis of the model output: a new screening method,* Reliability Engineering and System Safety 64 (1997), pp. 1-12.

[2] F. Campolongo, S. Tarantola, and A. Saltelli, *Tackling quantitatively large dimensionality problems,* Computer Physics Communications 117 (1999) pp. 75-85.

[3] J. H. Friedman, *Multivariate adaptive regression splines, Annals of Statistics* 19.1, 1-141, 1991.

[4] A. S. Hedayat, N. J. a. Sloane, and John Stufken, *Orthogonal Arrays: Theory and Applications,* Springer Series in Statistics, 1999.

[5] J. C. Helton and F.J. Davis, *Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems,* Reliability Engineering and System Safety, Vol. 81, No. 1, pp. 23-69, 2003.

[6] T. Homma and A. Saltelli, *Importance measures in global sensitivity analysis of non linear models,* Reliability Engineering and System Safety 52 (1996), pp. 117.

[7] T. Ishigami and T. Homma, *An importance quantification technique in uncertainty analysis for computer models,* in Proceedings of the ZSUMA 90, University of Maryland, USA, 1990. p. 398403.

[8] R. L. Iman and W. J. Conover, *A Distribution-free Approach to Inducing Rank Correlation Among Input Variables,* Commun. Statist. Simula. Computa., Vol 11, pp. 311-334, 1982.

[9] J. Jacques and C. Lavergne, *Sensitivity Analysis for Model with Correlated Inputs,* Reliability Engineering and System Safety 91 (2006), Issue 10-11, pp 1126-1134.

[10] J. R. Koehler and A. B. Owen, *Computer Experiments.*

[11] T. Kolda, *Revisiting Asynchronous Parallel Search for Nonlinear Optimization,* Technical Report SAND2004-8055, Sandia National Laboratories, Livermore, California, February 2004.

[12] T. J. Lorenzen and V. L. Anderson, *Design of Experiments : A No-Name Approach,* Marcel Dekker, Inc. 1993.

[13] M. D. McKay, *Evaluating Prediction Uncertainty,* Los Alamos National Laboratory Technical Report NUREG/CR-6311, LA-12915-MS, 1995.

[14] M. D. McKay, *Nonparametric Variance-based Methods of Assessing Uncertainty Importance,* Reliability Engineering and System Safety 57(1997), pp. 267-279.

[15] M. McKay, R. Beckman, W. Conover, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,* Technometrics, 21(2):239-245, 1979.

[16] M. McKay, M. A. Fitzgerald, and R. J.Beckman, *Sample Size Effects when Using R^2 to Measure Model Input Importance,* LANL technical report.

[17] M. D. Morris, *Factorial Sampling Plans for Preliminary Computational Experiments,* Technometrics, 21(2), pp. 239-245, 1991.

[18] M. D. Morris, *Private communication via email.*

[19] R. H. Myers and D. C. Montgomery, *Response Surface Methodology,* Second Edition, Wiley Series in Probability and Statistics, 2002.

[20] A. B. Owen, *Orthogonal Arrays for Computer Experiments, integration, and visualization,* Statist. Sinica 2, pp. 439-452, 1992.

[21] R. L. Plackett and J. P. Burman, *The Design of Optimum Multifactorial Experiments,* Biometrika, 33, pp 305-325.

[22] M. Ratto, S. Tarantola and A. Saltelli, *Estimation of Importance Indicators for Correlated Inputs,* Proceedings of the European Conference on Safety and Reliability, ESREL 2001, E. Zio, M. Demichela, N. Piccinini (eds.), torino, 16-20 September 2001, vol. 1, pp. 157-164.

[23] A. Saltelli, *Making best use of model evaluations to compute sensitivity indices,* Comput. Phys. Commun. 145 (2002), pp. 280297.

[24] A. Saltelli, K. Chan, E. M. Scott (editors), *Sensitivity Analysis,* Wiley Series in Probability and Statistics, 2000.

[25] I. M. Sobol, *Sensitivity estimates for nonlinear mathematical models,* Mathematical modeling and computational experiments, 1993.

[26] A. Saltelli, S. Tarantola, and K. Chan, *A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output,* Technometrics, Vol. 41, No. 1, pp. 39-55, 1999.

[27] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice,* Wiley, 2004.

[28] M. Stein, *Large Sample Properties of Simulations Using Latin Hypercube Sampling,* Technometrics, 29(2), pp. 143-151.

[29] B. Tang, *Latin Hypercubes and Supersaturated Designs,* Dissertation, Dept. of Statistics and Acturial Science, University of Waterloo, 1992.

[30] B. Tang, *Orthogonal Array-based Latin Hypercubes,* J. Amer. Statist. Assn. 88, 1392-1397, 1993.

[31] S. Tarantola, *Quantifying uncertainty importance when Inputs are correlated,* Foresight and Precaution, edited by Cottam, Harvey, Pape and Tait (eds), pp. 1115-1120, 2000.

[32] C. Tong, *Refinement Strategies for Stratified Sampling Methods,* Reliability Engineering and System Safety 91 (2006), Issue 10-11, pp 1257-1265.

[33] C. Tong and F. Graziani, *A Global Sensitivity Analysis Methodology for Multi-physics Applications,* LLNL Technical Report UCRL-TR-227800, 2007.