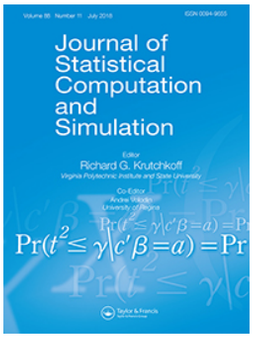

Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

Sensitivity analysis approaches to high-dimensional screening problems at low sample size

W. E. Becker, S. Tarantola & G. Deman

To cite this article: W. E. Becker, S. Tarantola & G. Deman (2018) Sensitivity analysis approaches to high-dimensional screening problems at low sample size, Journal of Statistical Computation and Simulation, 88:11, 2089-2110, DOI: 10.1080/00949655.2018.1450876

To link to this article: https://doi.org/10.1080/00949655.2018.1450876

Published online: 04 Apr 2018.

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION, 2018
VOL. 88, NO. 11, 2089–2110
https://doi.org/10.1080/00949655.2018.1450876

Taylor & Francis
Taylor & Francis Group

Sensitivity analysis approaches to high-dimensional screening problems at low sample size

W. E. Becker[a], S. Tarantola[a] and G. Deman[b]

[a]European Commission, Joint Research Centre, Ispra, Italy; [b]The Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel, Neuchâtel, Switzerland

ABSTRACT

Sensitivity analysis is an essential tool in the development of robust models for engineering, physical sciences, economics and policy-making, but typically requires running the model a large number of times in order to estimate sensitivity measures. While statistical emulators allow sensitivity analysis even on complex models, they only perform well with a moderately low number of model inputs: in higher dimensional problems they tend to require a restrictively high number of model runs unless the model is relatively linear. Therefore, an open question is how to tackle sensitivity problems in higher dimensionalities, at very low sample sizes. This article examines the relative performance of four sampling-based measures which can be used in such high-dimensional nonlinear problems. The measures tested are the Sobol' total sensitivity indices, the absolute mean of elementary effects, a derivative-based global sensitivity measure, and a modified derivative-based measure. Performance is assessed in a 'screening' context, by assessing the ability of each measure to identify influential and non-influential inputs on a wide variety of test functions at different dimensionalities. The results show that the best-performing measure in the screening context is dependent on the model or function, but derivative-based measures have a significant potential at low sample sizes that is currently not widely recognised.

ARTICLE HISTORY
Received 7 July 2017
Accepted 7 March 2018

KEYWORDS
Sensitivity analysis; screening; Sobol' indices; elementary effects; Derivative-based global sensitivity measures; G^* function; low-discrepancy sequences

AMS SUBJECT CLASSIFICATIONS
F1.1; F4.3

1. Introduction

Models are becoming more complex and computationally demanding: they may include dozens or even hundreds of inputs and analyse as many outputs. Knowledge of the inputs is often limited and, when dealing with numerous uncertain inputs, sensitivity analysis (SA) is widely employed to quantify the contribution to the uncertainty in the model output from individual inputs and groups of inputs.

One way of classifying problem types faced in sensitivity analysis is by dimensionality (the number of inputs of a model), and model run-time (the computational time required to execute a computer model for a given set of inputs) – see Figure 1. When the dimensionality and run-time of a model are both low, performing sensitivity analysis is relatively

CONTACT W. E. Becker william.becker@ec.europa.eu European Commission, Joint Research Centre, Via E. Fermi 2749, 21027, Ispra, Italy

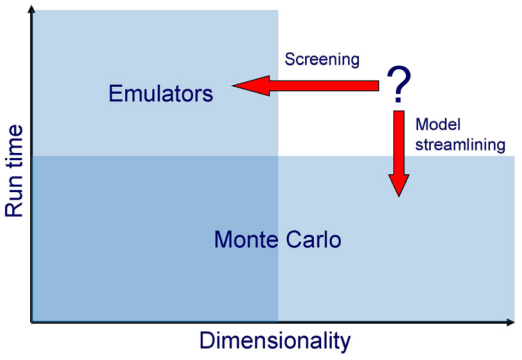

Figure 1. A classification of problems in sensitivity analysis by dimensionality and run time.

straightforward – a typical approach would be to run the model many thousands of times in order to estimate Sobol' sensitivity indices via the Monte Carlo method [1]. If the run time of the model is low, but the number of inputs is high, the Monte Carlo method is usually still appropriate since its cost scales well with dimensionality. On the other hand, if the dimensionality is low, but the run-time of the model is high, an emulator-based approach is usually adopted (also known as 'surrogate models', 'metamodels' and 'response surfaces'). This involves building a mathematical approximation of the model (typically a Gaussian Process [2,3], Polynomial Chaos Expansion (PCE) [4] or state-dependent parameter regression [5]) using a limited number of model runs as 'training data', then estimating sensitivity indices using the emulator, which can be run a great number of times for a negligible computational cost.

Although there is extensive literature on the emulator problem in sensitivity analysis, there is relatively little that deals with the very common situation where one has both a high dimensionality *and* a high run time. In such cases, fitting an emulator is generally impractical because the number of sample points required to successfully emulate the model becomes prohibitively high. Furthermore, the model cannot be run enough times for an accurate estimation of sensitivity indices.

Although some recent approaches such as 'sparse PCE' have been successfully applied in some limited cases to high-dimensional problems at low sample sizes [6], the models tested were linear with few interactions. In tests on several emulators on analytical functions, emulators were not found to provide reasonable estimations of sensitivity indices unless the sample size needed to train the emulator was dramatically increased (and were generally out-performed by sample-based measures) [7]. Thus, emulator approaches are a viable option in high dimensions only if the model or function is sufficiently 'well-behaved' (linear and few interactions), or the available sample size for training the emulator is high.

In general then, one is left with one of two approaches: either to somehow streamline the model to make it run faster, thereby bringing it within reach of Monte Carlo estimation of sensitivity indices, or to somehow reduce the dimensionality of the problem. Although the first approach can often be fruitful, it is only a viable option if one is the developer of the model; besides, the run time might already have been minimised. This paper focuses on the second strategy, which is often called 'screening', 'factor fixing' or 'freezing variables'.

Screening aims to answer the question, ' Which inputs could be fixed to an arbitrary value within their range of uncertainty such that the expected reduction in output uncertainty would be small?' [8]. Screening is usually associated with the elementary effects method, which is a sensitivity analysis approach based on finite differences [9,10], but to the knowledge of the authors there is no theoretical or empirical evidence which shows that the elementary effects method consistently performs better at low sample sizes than other measures of sensitivity. The objective of this paper is therefore to compare the performance of the absolute mean measure μ^* from the elementary effects method, on a high-dimensional problem at small sample size, against two other measures of sensitivity, the well-known Sobol' total sensitivity indices S_T [11] and a derivative-based global sensitivity measure (DGSM) ν [12,13]. In this work we discuss the links between the three measures, and additionally investigate a variation of the DGSM measure which is included in the testing. It should be emphasised that, for the reasons discussed previously, emulator-based approaches to estimating these measures are not tested in this work because they do not provide reasonable estimates at the very small sample sizes investigated. Therefore, this study focuses solely on estimation of 'sampling-based' screening measures (those that do not rely on data modelling).

Because screening and factor fixing is primarily interested in distinguishing between influential and non-influential inputs, we assess the performance of each measure in terms of the proportion of 'influential' inputs correctly identified in the test functions examined in this paper, rather than using the error between estimated and analytical values of each measure, as for example in [14]. We use analytical test functions rather than physical models because the analytical functions allow us to define a priori the set of influential and non-influential inputs, thus allowing an evaluation of the performance of the screening measures against a given pattern of importance.

In the experiments performed here, the input variables are assumed to be independent. While there is an increasing body of literature on sensitivity analysis of models with dependent inputs, dependence significantly complicates the analysis because (apart from practical considerations) the influence of an input on the output can be due to its own effect, plus effects due to other inputs with which it is correlated (or anti-correlated). This creates an ambiguity in the ranking, and results in a non-unique variance decomposition. The idea of screening correlated variables may be investigated in future work.

The remainder of this paper is organised as follows: in Section 2 the four SA measures considered in this study are briefly introduced. We refer however to the vast literature available for further details on the measures, limiting the description to the most relevant features with respect to the present topic. In Section 3, the numerical experiments are described along with the approach used to measuring screening error. In Section 4, the test functions are introduced and the results of the sensitivity measures are given with some discussion. Finally, an overview of the main findings and conclusions is in Section 5.

2. Measures of sensitivity

For all sensitivity measures described in this paper we assume that the inputs are independent. Let us denote by $x_1, x_2, \ldots, x_k$ the inputs of a test function f, defined over the unit hypercube $\mathcal{H}^k$, with y being the output of the function, such that $y = f(x_1, x_2, \ldots, x_k)$.

Further, assume that the uncertainty in the model inputs is expressed by a joint independent uniform probability distribution, $p(x_1, x_2, \ldots, x_k) = \mathcal{U}(0,1)^k$. Note that (apart from the assumption of independence), this incurs no loss of generality because any distribution can be transformed onto the unit interval. As a result of the input uncertainty, the uncertainty in the output y is correspondingly expressed as $p(y)$.

Any point $\boldsymbol{x}_j = x_{1j}, x_{2j}, \ldots, x_{kj}$ in the hypercube represents a given set of values for the k inputs, for which the output y_j can in turn be evaluated by executing the test function (we talk in this case of function evaluations, or model runs). In general, physical models cannot be expressed in closed form, so estimation of $p(y)$ and sensitivity analysis involves calculating many values of y at different $\boldsymbol{x}$. Each point sampled in the hypercube will be used for one model run and will provide one value of y; in the case of a physical model this evaluation can take very different computational time, from nanoseconds to hours, or even days. Hence, it is important, especially in the latter case, to be able to obtain accurate sensitivity measures with the minimum number of runs. In this paper we use analytical test function in place of real physical models, as this allows us to run the test function in negligible time and, above all, because we can define a priori a given importance structure to enable a proper performance assessment of the screening measures.

One approach to sensitivity analysis is to decompose the variance of $p(y)$ into portions attributable to inputs and sets of inputs:

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{i,j} + \cdots + V_{1,2,\ldots,k}, \tag{1}$$

where

$$V_i = V[E(y|x_i)]$$
$$V_{i,j} = V[E(y|x_i, x_j)] - V[E(y|x_i)] - V[E(y|x_j)]$$

and so on for the higher order terms. Here, $V(\cdot)$ denotes the variance operator, and the terms are used directly as sensitivity indices, e.g. the *first-order sensitivity index* $S_i = V_i/V(y)$ measures the contribution of the input x_i to $V(y)$, without including interactions with other inputs [1]. The sum $\sum S_i = 1$ $(i = 1, \ldots, k)$, for purely additive models whereas $\sum S_i \ll 1$ for models with strong interactions.

The variance-based measure that is used in this work is denoted S_{Ti}, and is called the *total order sensitivity index*, which is defined as [11],

$$S_{Ti} = 1 - \frac{V[E(y \mid \boldsymbol{x}_{\sim i})]}{V(y)} = \frac{E[V(y \mid \boldsymbol{x}_{\sim i})]}{V(y)}, \tag{2}$$

where $V(\cdot)$ denotes the variance operator, $E(\cdot)$ the expected value, and $\boldsymbol{x}_{\sim i}$ the set of all inputs except x_i. The total order sensitivity index measures the contribution to $V(y)$ of a given input x_i, as well as all its interactions of any order with other inputs.

In order to estimate S_{Ti}, we generate a set of N sampling points, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, suitably sampled from $\mathcal{H}^k$. This may be expressed in a design matrix $\mathbf{X}$ of N rows and k columns, which is used to estimate all the sensitivity measures here. Letting $\boldsymbol{x}_j$ and $\boldsymbol{x}_j^{(i')}$ be, respectively, a point in the input space, and a point that differs from $\boldsymbol{x}_j^{(i)}$ only in the value of x_i,

an estimator of the numerator of S_{Ti} (V_{Ti}) is as follows [15,16]:

$$\hat{V}_{Ti} = \frac{1}{2N} \sum_{j=1}^{N} |f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j)|^2. \tag{3}$$

The absolute value notation here is used for consistency with the other estimators presented here. The $\boldsymbol{x}_j$ therefore correspond to the points from $\mathbf{X}$, and the $\boldsymbol{x}_j^{(i')}$ represent random shifts in the x_i direction. Normally, this estimator is used with large values of N in order to more accurately estimate S_{Ti}. Here, we investigate its utility at very low sample sizes.

Another widely-used measure of sensitivity is the mean of absolute elementary effects, which is estimated as follows [17]:

$$\hat{\mu}_i^* = \frac{1}{N} \sum_{j=1}^{N} \frac{|f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j)|}{|x_{ji}^{(i')} - x_{ji}|}. \tag{4}$$

Here, x_{ji} denotes the ith coordinate of $\boldsymbol{x}_j$, so that the denominator of Equation (4) is equal to the difference in x_i between $\boldsymbol{x}_j$ and $\boldsymbol{x}_j^{(i')}$. The elementary effects measure is most commonly used in the screening setting [17]. Note however that it cannot be interpreted in terms of variance in the same way as S_{Ti}.

The final measure used in this study is part of a set of sensitivity measures called 'derivative-based global sensitivity measures' (DGSM). The measure is the integral of squared partial derivatives, i.e. $\nu_i = \int_{\mathcal{H}} (\partial y/\partial x_i)^2 \, \mathrm{d}\boldsymbol{x}$. This may be estimated as follows [13]:

$$\hat{\nu}_i = \frac{1}{N} \sum_{j=1}^{N} \frac{|f(\boldsymbol{x}_j^{(i'')}) - f(\boldsymbol{x}_j)|^2}{|x_{ji}^{(i'')} - x_{ji}|}, \tag{5}$$

where $\boldsymbol{x}_j^{(i'')}$ is a point that differs from $\boldsymbol{x}_j$ only by a small increment δ of x_i, in order to give an estimate of $\partial y/\partial x_i$ at each point $\boldsymbol{x}_j$. This increment, and the method of estimation of partial derivatives, has been studied in more detail in [18,19], in which it was concluded that DGSM provides the same quality of information as the Sobol' indices while being significantly less computationally demanding, as well as the fact that the computational efficiency of DGSM depends on the parameter. Methods of automatic differentiation applied to DGSM have also been proposed in [19,20], which can further increase the computational efficiency. In this work, however, we use the original approach which takes a fixed increment of $\delta = 1 \times 10^{-5}$ when sampling with respect to the unit hypercube.

Notice that the three measures share a number of similarities. The first is that they are all estimated using random samples in $\mathcal{H}^k$, with perturbations in each x_i direction. This results in a so-called radial design which consists of N 'stars' in the input space, each having a point from $\mathbf{X}$ at its centre (see Figure 2). Hence, for N radial samples, the total number of sample points (and therefore of model runs) is $N_T = N(k+1)$. In the cases of S_{Ti} and μ_i^*, the perturbations are large and random, whereas in the estimator of ν_i they are small and kept constant. Some alternative estimation techniques for these measures can reduce the number of model runs required, such as metamodelling [21] or automatic differentiation for DGSM [20].

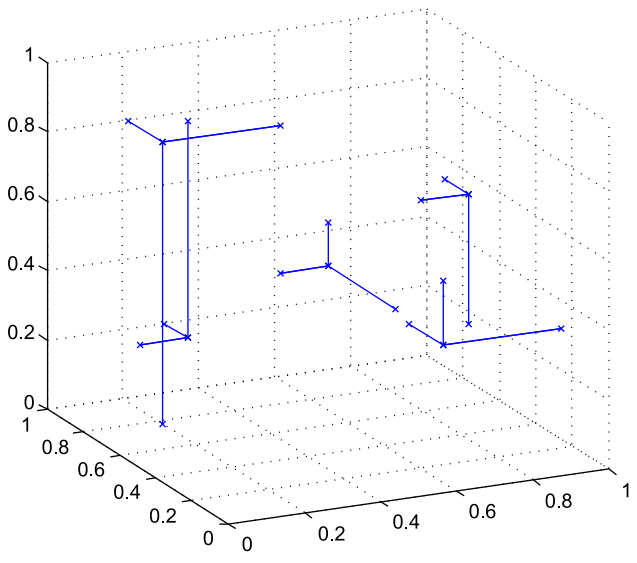

Figure 2. An illustration of a radial sampling design with large perturbations, with $N = 5$ and $k = 3$.

Next, all three measures use the difference between $f(\boldsymbol{x}_j)$ and $f(\boldsymbol{x}_j^{(i')})$ (or $f(\boldsymbol{x}_j^{(i'')})$) as a basis for measuring sensitivity – for S_{Ti} and ν_i this is the squared difference, whereas for μ_i^* it is the absolute difference. Finally, the estimators for μ_i^* and ν_i both use the difference in x_i as a denominator, whereas S_{Ti} does not use this information. In [13], it was shown that ν_i is an upper bound on V_{Ti}, such that they are related by the inequality $V_{Ti} \leq \nu_i/\pi^2$.

The similarities and differences between these three measures led us to consider a fourth measure which is a variation of ν_i, which we denote ξ_i. This is defined as $\xi_i = \int_{\mathcal{H}} |\partial y/\partial x_i| \, \mathrm{d}\boldsymbol{x}$ and is estimated as follows:

$$\hat{\xi}_i = \frac{1}{N} \sum_{j=1}^{N} \frac{|f(\boldsymbol{x}_j^{(i'')}) - f(\boldsymbol{x}_j)|}{|x_{ji}^{(i'')} - x_{ji}|}, \tag{6}$$

i.e. it uses the absolute value in the numerator, rather than the squared value. The aim here is simply to see whether the use of the absolute value, as opposed to the squared value, can help in screening in high dimensions. Note that this measure was also mentioned in [13], in which it was shown that it is the limit of the Morris measure as $N \to \infty$ and $|x_{ji}^{(i'')} - x_{ji}| \to 0$; furthermore, it was pointed out that $\xi_i \leq \sqrt{\nu_i}$, and $\nu_i \leq C\xi_i$ if $|\partial y/\partial x_i| \leq C$. The measure was also investigated in numerical experiments in [12], however, the focus was on lower dimensional problems up to $k = 10$. Here, the focus is on the screening context where dimensionality is high and sample size is low.

3. Methodology

The aim of the tests performed here is to simulate as closely as possible the setting for a screening analysis. Screening is a technique commonly employed in sensitivity analysis, usually motivated by the following two problem constraints:

(1) The number of inputs is high, i.e. there are at least around 30 variables, and possibly hundreds (or even thousands).
(2) The number of runs that can be performed is low, perhaps in the region of a few hundred at most.

The combination of these two conditions precludes the use of Monte Carlo approaches and the use of emulators, since both would require a restrictively high number of runs. In such a setting, a common approach in sensitivity analysis is to perform a two-step analysis. In the first (screening) step, a rough first sensitivity analysis is performed using a very limited number of runs and a sampling-based approach such as the elementary effects method [9]. While this does not give precise measures of sensitivity, it is enough to divide the inputs into an influential set and a non-influential set. Then a more detailed sensitivity analysis can be performed on the remaining set of influential inputs, since in many cases the dimensionality will now be reduced to a size which is feasible for either Monte Carlo methods (see, e.g. [17]), or emulators (see [22]). The idea of finding a small set of inputs that explain the majority of the output uncertainty is based on the *principle of factor sparsity*, also known as the *Pareto Principle* [23], which is an observation that in many situations, 80% of the effects are due to 20% of the causes.

Notice that in the screening step, we are not primarily interested in obtaining precise values of sensitivity measures; rather, the key objective is to split the inputs into two groups: the influential group, and the non-influential group. For this reason, to judge the success of a screening method it is not appropriate to examine the convergence of the measures to theoretical values, for example. Our approach to measuring screening performance therefore consists of a range of analytical test functions, in which the inputs can be set to be influential or non-influential, and a measure of screening error which is based on the number of influential variables incorrectly identified as being un-influential (i.e. a mis-classification). This latter is explained in more detail in Section 3.2.

3.1. Test functions

Since the success of one screening method over another is dependent on the type of model/function to be investigated, the approach taken here is to use several test functions as the basis of comparison that have different characteristics regarding dimensionality, nonlinearity, continuity and so forth. Test functions are used rather than physical models because with test functions we can set inputs to be influential or non influential by controlling parameters' values that we can set a priori. Additionally, the test functions are very cheap to evaluate, and are conceptually no different from physical models, in that they are simply a function of a number of uncertain inputs. In the case of a physical model, we would actually not know the true importance of the inputs, so it would be difficult to draw conclusions on the success of a given screening analysis.

In order to reflect a screening setting, for each function we set a certain fraction γ of the inputs to be of higher influence on the model output (the *influential set*), and the remaining fraction $1 - \gamma$ of inputs to be of lower influence (the *uninfluential set*). At a given sample size of the experimental design, each screening method attempts to correctly separate the two sets.

Our experiments are then set as follows. For a function with k inputs, let $k_{\text{high}} = \lfloor \gamma k \rfloor$, i.e. the number of inputs that belong to the influential set, and $k_{\text{low}} = k - k_{\text{high}}$, the number of inputs that belong to the uninfluential set. In each test function, the inputs are set as important or non important by selecting suitable values of the parameters a_i (see details in the next section). In other words, the first k_{high} inputs, comprising the influential set, are set to have equal and high importance, and the remainder (the uninfluential set) to have equal and low importance.

We aim to investigate a variety of function types and dimensionalities which are typically used to test the performance of sensitivity analysis methods. For this reason, we use three different test functions: one is a sum of simple polynomial functions which is a smooth nonlinear additive function; the second is a more complex nonlinear function with strong interactions and a discontinuity in the first derivative; and the last function has a near-discontinuity in its response (details can be found in Section 4. We investigate dimensionalities of $k = \{30, 50, 75, 100\}$ and change the proportion of influential variables between $\gamma = 0.2$ and $\gamma = 0.5$. Additionally, for each test function, we investigate different parameter values a_i to provide a wide range of scenarios.

3.2. A measure of screening error

In order to sort between influential and uninfluential input variables, there are two main steps. The first is to estimate a sensitivity measure for each input variable, which will result in a ranking. The second step is to somehow use a threshold value, or classification/clustering procedure, to define which sensitivity values to identify as influential and which as uninfluential. Clearly, the first step is a prerequisite for the second. Whatever classification method or threshold value is used, a necessary (but not sufficient) condition to correctly classify the variables is that the high-importance variables all have higher sensitivity scores (and therefore ranks) than all of the low-importance variables. If this is not the case, then whatever threshold value is used, there will be some classification error. Since there are many possible methods of classification and/or threshold values, the measure of screening performance used here focuses on exactly this prerequisite, as opposed to the results of a classification procedure, which would make the results dependent on which classification procedure we choose.

Following this logic, the measure of screening error proposed here is based on the number of *influential* variables that are ranked below *un-influential* variables. In the following, this is explained in more detail and links to similar measures are given.

At any given sample size, by employing one of the aforementioned screening methods, each input x_i is characterized by an estimated measure of its sensitivity, s_i (we use s_i as a generic notation for a sensitivity measure, which could be S_{Ti}, μ_i^*, ν_i or ξ_i. Accordingly, a ranking r_i can be defined for each input x_i, where ranking runs in descending order, i.e. $r_i = 1$ ranks x_i as the input with the highest s_i, whereas $r_i = k$ corresponds to the input with the lowest s_i. Given that we know *a priori* the true ranking of the k inputs, thus which of them belong to the *influential set*, we define our measure of error Z as follows:

$$Z = \frac{1}{k_{\text{high}}} \sum_{i=1}^{k_{\text{high}}} 1(r_i > k_{\text{high}}), \tag{7}$$

where $1(\cdot)$ is the count function. Recall that in all the experiments here, the set of influential variables are the first $\{1, 2, \ldots, k_{\text{high}}\}$ indices. The sum $\sum_{i=1}^{k_{\text{high}}} 1(r_i > k_{\text{high}})$ therefore counts the number of *false-negative* inputs, i.e. the number of influential variables incorrectly identified as being in the un-influential set. Equivalently, since this is a binary classification problem, this is also equal to the number of un-influential variables incorrectly classified in the influential set, i.e. the number of false positives. By dividing by k_{high}, the number of influential variables, the final measure Z is the *fraction of false positives*. In summary, it is a measure of inaccuracy of the screening method in individualizing the influential set, such that $Z = 0$ indicates that all influential inputs have been correctly identified, and $Z = 1$ indicates that none of the influential variables have been identified (based on the ranking).

The measure Z can also be interpreted in other ways: for example, $1-Z$ measures the fraction of *true positives*: inputs that are correctly estimated as belonging to the influential set. In pattern recognition and binary classification, this is known as the *recall*. Similarly, $Zk_{\text{high}}/k_{\text{low}}$ is the fraction of false negatives, and $1 - Zk_{\text{high}}/k_{\text{low}}$ is the fraction of true negatives.

This methodology is purely a measure of sorting the inputs into high- and low-importance groups, and gives no regard to precise rankings or possible cut-off values that might be used to select high-importance from low-importance inputs, since in our experience, what is a ' high-importance' input is usually problem-dependent and is decided by the analyst. Instead, the Z measure relies on the basic logic that whatever the cut-off values that might be used, the first criterion for effective screening is that the most influential inputs are ranked before the less influential ones. A practical example of computation of the Z measure is offered in Section 3.4

3.3. Sampling and replications

All the measures of sensitivity discussed here require a random sample in $\mathcal{H}^k$ as a basis.[1] In practice however, many practitioners of sensitivity analysis use *quasi-random* numbers in place of pseudo-random numbers, to increase the rate of convergence of the estimators (see, for example, [24]). In the last decades, many techniques have been studied. Among these, latin hypercube sampling (LHS) [25,26] and latin supercube sampling (LSS) [27] were developed for the design of real experiments, whereas low-discrepancy sequences such as the Sobol' sequence [28] were conceived to address the problem of numerical integration. Recent studies have compared these experimental designs with regard to the convergence rates of sensitivity indices; Sobol' sequences have proved to be more efficient than both LHS [29], LSS [14] and simple Monte Carlo sampling at low sample size. Moreover, Sobol' sequences can be enhanced by adding new sampling points in the experimental design while keeping the sampling in $\mathcal{H}^k$ as uniform as possible – as such they have probably become the sequence of choice for the majority of practitioners of global SA. In this work, we therefore test both the Sobol' sequence as well as (pseudo-) random numbers as a basis for comparison.

Given that the test functions are explored in high dimensions and at low sample sizes, the location of sample points used to estimate sensitivity measures will have a significant impact on the results. In order to average the performance over possible sampling designs,

for each test function investigated, 50 replications are made. In the case of random sampling, this is simply done by drawing 50 independent samples of random numbers in $\mathcal{H}^k$ which are used for the basis of estimation of the sensitivity measures.

For the Sobol' sequence of quasi-random numbers, which is deterministic, the sample is randomised by applying a random shift in each dimension, following the so-called Cranley–Patterson rotation approach of Cranley and Patterson [30]. For each replication, we generate a random k-length vector, $\mathbf{v}$, such that $\mathbf{v} \in \mathcal{H}^k$. This vector is then used to shift all points from the Sobol' sequence such that for any point $\boldsymbol{x}_j$, we generate a corresponding randomly shifted point $\tilde{\boldsymbol{x}}_j$, where

$$\tilde{\boldsymbol{x}}_j = \boldsymbol{x}_j + \mathbf{v} - \lfloor \boldsymbol{x}_j + \mathbf{v} \rfloor, \tag{8}$$

where $\lfloor u \rfloor$ denotes the greatest integer less than or equal to u. This is equivalent to shifting the whole sample in a random direction, preserving points that move outside the unit hypercube. In this way, the structure of the sample is preserved at each replication, but a random element is introduced, which allows for an assessment of the properties of the sample, averaged over the position of the sample in the sample space.

Note that this randomisation approach was originally proposed for standard lattice rules, and other approaches such as the scrambling method of Owen can in theory better preserve the properties of the Sobol' sequence [31]. However, the computational cost of this method is substantial, particularly in high dimensions. Moreover, research has shown that in practical cases, the Cranley–Patterson rotation can be equally effective [32,33]. For these reasons, and for reasons of simplicity, this was taken as the method of choice in this study.

Additionally, in order to observe the effects of sample size, each function is tested at values of $N = 1, 2, \ldots, 10$, which result in total sample sizes N_T ranging from $N_T = 31$ (for $N = 1$ and $k = 30$), to $N_T = 1010$; (when $N = 10$ and $k = 100$). The dimensionality of the test functions k, is set to $k = 30{,}50{,}75{,}100$, reflecting typical numbers of inputs encountered in screening.

3.4. Example

In summary, we investigate the performance of each sensitivity measure in a large number of settings, using different test functions, at different dimensionalities, sample sizes, and parameter values. For each combination of these characteristics, we calculate Z averaged over 50 sample replications.

For the sake of clarity an example follows. Say that we have a test function comprising $k = 5$ independent random inputs : x_1, x_2, x_3, x_4 and x_5. The parameters of the function are set so that the influential set comprises $k_{\text{high}} = 2$ inputs, e.g. x_1 and x_2. Conversely, the non-influential set comprises $k_{\text{low}} = 3$ inputs, i.e. x_3, x_4 and x_5. At a given sample size, say $N_T = 6$, the sensitivity measures are estimated using one of the aforementioned screening techniques, leading to the following ranking: $\hat{s}_1 > \hat{s}_3 > \hat{s}_2 > \hat{s}_5 > \hat{s}_4$. Here, the measure of error is $Z = 0.5$, because one of the two influential inputs is falsely estimated as belonging to the negative set ($r_2 = 3 > k_{\text{high}}$); as a consequence, one of the un-influential inputs is misclassified in the positive set, resulting in a false positive.

A replication, applying the very same settings but with a different sample of $N_T = 6$ experiments, provides say $\hat{s}_2 > \hat{s}_1 > \hat{s}_4 > \hat{s}_3 > \hat{s}_5$. Here, the measure of error is $Z = 0$; none

of the two high-sensitivity variables are falsely estimated as belonging to the negative set. Another replication provides say $\hat{s}_5 > \hat{s}_2 > \hat{s}_4 > \hat{s}_1 > \hat{s}_3$. Again, the measure of error is $Z = 0.5$. Then, an average Z is computed considering the three replicas : $\bar{Z} = 0.33$. Hence, at a sample size $N_T = 6$, the average inaccuracy of the screening technique for individualizing the influential set is 0.33 ; i.e. the screening technique is 33% inaccurate on average in identifying the influential inputs from the non-influential ones. Clearly, in this example, there are only three replications instead of 50, but it is intended to illustrate the procedure.

4. Test functions

In this section, the results displayed are from Sobol's quasi-random sampling unless otherwise stated. This is because the relative performance (that is, how well each measure performs compared to the others) when using the Sobol' sequence as opposed to random numbers is almost identical. Additionally, Sobol' sampling results in lower Z values (screening error) than random sampling, at given sample sizes. In Section 4.4 we do however briefly summarise the relative performance with random sampling.

4.1. Polynomial additive function

The first function used to compare the four measures was a simple polynomial additive function, of the form,

$$y = f(\boldsymbol{x}) = \sum_{i=1}^{k} a_i x_i^p, \tag{9}$$

where p is the order of the polynomial, and the a_i are the parameters whose values we appropriately choose in order to set a priori the influential and the non-influential inputs. A plot of this function with $k = 2$ is found in Figure 3. To give an idea of sensitivities, at $k = 30$, $a_{\text{high}} = 2$, $a_{\text{low}} = 1$, $p = 2$ and $\gamma = 0.2$ (γ is the share of influential inputs), the first six inputs have a sensitivity of $S_T = 0.085$, while the remainder have $S_T = 0.021$, i.e. around a quarter of the sensitivity of the influential inputs.

Table 1 shows the results of four configurations of the polynomial function, at different dimensionalities. The numbers in each sensitivity measure column refer to ranks of which measure performed best on average over all sample sizes, at the given parameter values. Recall here that tests were performed at values of $N = 1, 2, \ldots, 10$, which resulted in total sample sizes N_T ranging from $N_T = 31$ for $N = 1$ and $k = 30$, to $N_T = 1010$; when $N = 10$ and $k = 100$ (given that $N_T = N(k + 1)$ for radial sampling, as explained in Section 2). A rank value of 1 is assigned to the measure which has the lowest Z values at the majority of values of N_T. A rank value of 2 is given to the measure which has the lowest Z values at the majority of values of N_T, if the first-ranked measure were not there, and so on. This is simply a way to illustrate the results of a large number of experiments in a condensed fashion.

The coefficients are set at $a_{\text{high}} = 2$ and $a_{\text{low}} = 1$ in all cases, but the order of the polynomial is increased from 2 to 3, and the fraction of significant inputs from 0.2 to 0.5. In these experiments, the best measure is now unequivocally μ^*, with ν being the second best, followed by ξ, and finally S_T in the last place. The results do not seem to depend on the dimensionality k or on other combinations of parameters a_i that have been investigated. To

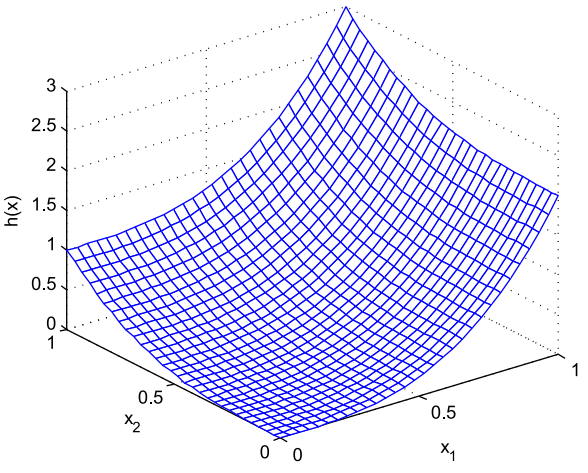

Figure 3. Plot of polynomial function for $k = 2$, $a_{\text{high}} = 2$, $a_{\text{low}} = 1$, $p = 3$, $\gamma = 0.5$.

Table 1. Configuration and performance rankings of experiments with polynomial additive function.

k	a_{high}	a_{low}	p	γ	S_T	μ^*	ν	ξ
30	2	1	2	0.2	4	1	2	3
50	2	1	2	0.2	4	1	2	3
75	2	1	2	0.2	4	1	2	3
100	2	1	2	0.2	4	1	2	3
30	2	1	2	0.5	4	1	2	3
50	2	1	2	0.5	4	1	2	3
75	2	1	2	0.5	4	1	2	3
100	2	1	2	0.5	4	1	2	3
30	2	1	3	0.2	4	1	2	3
50	2	1	3	0.2	4	1	2	3
75	2	1	3	0.2	4	1	2	3
100	2	1	3	0.2	4	1	2	3
30	2	1	3	0.5	4	1	2	3
50	2	1	3	0.5	4	1	2	3
75	2	1	3	0.5	4	1	2	3
100	2	1	3	0.5	4	1	2	3

see in a little more detail, Figure 4 shows two selected plots, the first showing the results of the quadratic function, and the second of the cubic function, with $\gamma = 0.2$ in both cases. These are representative of the relative performance of the other cases in Table 1. To briefly explain these plots, in Figure 4(a), for example, the upper line shows that at a total cost of around $N_T = 100$ function evaluations, S_T is able to rank the inputs so that about a little under half of the significant variables are identified (in the first k_{high} ranking positions), since $Z \approx 0.58$. The DGSM measures, at the same sample size, can identify more than half of the significant inputs ($Z \approx 0.45$). In all cases these values represent average Z values over 50 sample replications.

It is evident first of all that there is not a huge difference between any of the measures, but the μ^* measure is best by a small margin, particularly at the lowest sample sizes. In the quadratic case, the performance is overall better for all four measures compared to the cubic function, such that we see $Z = 0$ from $N_T = 400$ in the former case for all measures except S_T, whereas in the latter case $N_T = 800$ sample points are required.

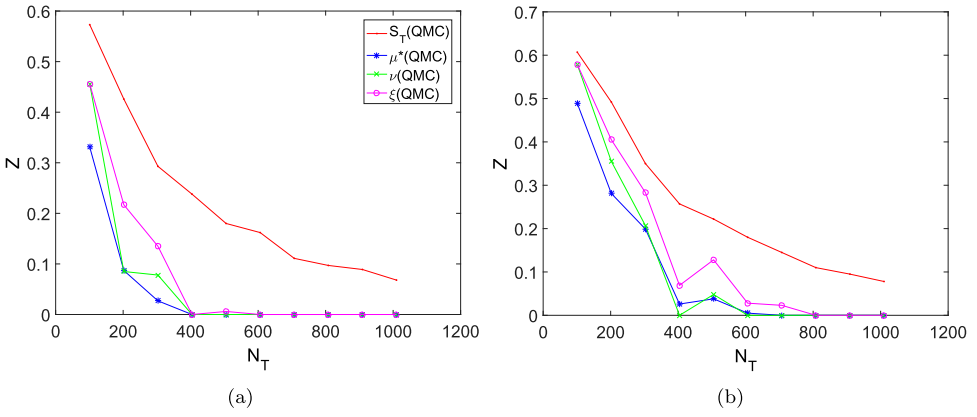

Figure 4. Convergence plots for polynomial function: (a) $p = 2$; (b) $p = 3$; In all cases $k = 100$, $a_{\text{high}} = 2$, $a_{\text{low}} = 1$ and $\gamma = 0.2$.

4.2. G* function

The G^* test function is a more complicated nonlinear and non-additive function, as used in [16] among many other sensitivity analysis articles. It has the following form:

$$y = G^* = G(x_1, x_2, \ldots, x_k, a_1, a_2, \ldots, a_k, \alpha_1, \alpha_2, \ldots, \alpha_k) = \prod_{i=1}^{k} g_i^*$$

$$g_i^* = \frac{(1+\alpha_i)|2(x_i - \mathbf{I}[x_i]) - 1|^{\alpha_i} + a_i}{1 + a_i}, \tag{10}$$

where $a_i \geq 0$ and $\alpha_i \geq 0$ are parameters which can be chosen to obtain different behaviours of the function, and $\mathbf{I}[x_i]$ is the integer part of x_i. The relative importance of the inputs $(x_1, x_2, \ldots, x_k)$ in the G^* function is controlled by the magnitude of a_i (for this test function, the smaller a_i the more important is x_i) and the nonlinearity by α_i.

To illustrate the influence of the parameters, Figure 5 shows plots of $g^*(a, \alpha)$ for values of a and α. In the left figure, a value of $a = 3$ is used, and plots of $\alpha = \{1, 2, 3\}$ are shown, which give linear, quadratic and cubic behaviour, respectively, centred at $x = 0.5$. On the right side, the same coefficients are used, except $a = 10$. Here, it is evident that the effect on y is of a lesser magnitude, which illustrates that the higher the value of a, the lower the amplitude of g^*, and the lower the importance of the corresponding input. The coefficient α acts on the curvature of the bottom part of the g^* function, thus maximizing the slope towards the high values.

As with the other functions, the dimensionality was set to $k = 30,50,75,100$ to represent dimensionalities that might be encountered in a screening analysis. The parameter α was also set to $\alpha = 1, 2, 3$ to test at different values of nonlinearity. Finally, the a_i parameters were set in two scenarios: the first a low-interaction with $a_{\text{high}} = 3$ and $a_{\text{low}} = 10$. The second scenario set $a_{\text{high}} = 1$ and $a_{\text{low}} = 2$, with strong interactions between the inputs. Figure 6 shows a plot of the G^* function for $k = 2$, using the coefficients from the latter case.

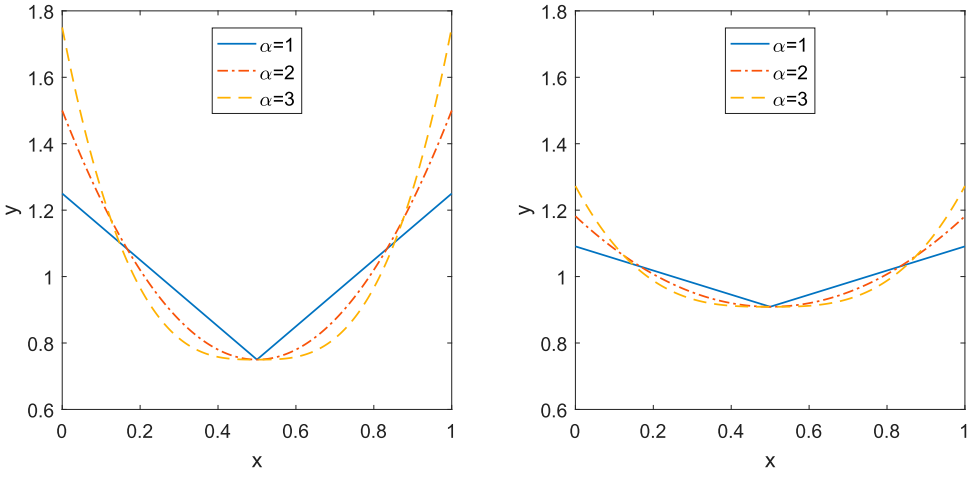

Figure 5. Plots of G^* function at $k = 1$ and $\alpha = \{1, 2, 3\}$, with: $a = 3$ (left), $a = 10$ (right).

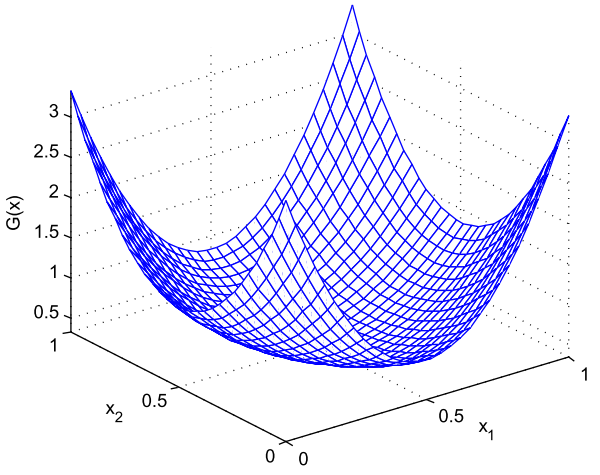

Figure 6. Plot of G^* function for $k = 2$, $a_{\text{high}} = 1$, $a_{\text{low}} = 2$, $\alpha = 2$, $\gamma = 0.5$.

Table 2 shows the relative performance of the various sensitivity measures on different configurations of the G^* function. The table reveals a number of features: first of all, for a given set of parameters, the dimensionality of the function does not generally impact the relative performance of the sensitivity measures. One can also see that, in contrast to the polynomial function, the DGSM measures ν and ξ are almost always the best, with ξ having the best performance in 85% of cases. Then for all cases the μ^* measure is the third best, followed by S_T, which is the last.

More detail can be obtained by examining plots of the error measure Z from Equation (7) against N_T, showing the decrease of the error of each measure with total sample size. Figure 7(a) shows the performance at $k = 100$, $a_{\text{high}} = 1$, $a_{\text{low}} = 2$, $\alpha = 2$ and $\gamma = 0.2$, in which it can be seen that the two derivative-based measures perform much better than S_T and μ^*, with a very small difference between ξ and ν. Keeping these parameter values

Table 2. Configuration and performance rankings of experiments with G^* function.

k	a_{high}	a_{low}	α	γ	S_T	μ^*	ν	ξ
30	1	2	2	0.2	4	3	2	1
50	1	2	2	0.2	4	3	2	1
75	1	2	2	0.2	4	3	1	2
100	1	2	2	0.2	4	3	1	2
30	1	2	2	0.5	4	3	2	1
50	1	2	2	0.5	4	3	2	1
75	1	2	2	0.5	4	3	2	1
100	1	2	2	0.5	4	3	2	1
30	1	2	3	0.2	3	4	1	2
50	1	2	3	0.2	4	3	1	2
75	1	2	3	0.2	4	3	1	2
100	1	2	3	0.2	4	3	1	2
30	1	2	3	0.5	4	3	2	1
50	1	2	3	0.5	4	3	2	1
75	1	2	3	0.5	4	3	2	1
100	1	2	3	0.5	4	3	2	1
30	3	10	1	0.2	4	3	1	1
50	3	10	1	0.2	4	3	1	1
75	3	10	1	0.2	4	3	1	1
100	3	10	1	0.2	4	3	1	1
30	3	10	1	0.5	4	3	1	1
50	3	10	1	0.5	4	3	1	1
75	3	10	1	0.5	4	3	1	1
100	3	10	1	0.5	4	3	1	1
30	3	10	2	0.2	4	3	2	1
50	3	10	2	0.2	4	3	2	1
75	3	10	2	0.2	4	3	2	1
100	3	10	2	0.2	4	3	2	1
30	3	10	2	0.5	4	3	2	1
50	3	10	2	0.5	4	3	2	1
75	3	10	2	0.5	4	3	2	1
100	3	10	2	0.5	4	3	2	1
30	3	10	3	0.2	4	3	2	1
50	3	10	3	0.2	4	3	2	1
75	3	10	3	0.2	4	3	2	1
100	3	10	3	0.2	4	3	2	1
30	3	10	3	0.5	4	3	2	1
50	3	10	3	0.5	4	3	2	1
75	3	10	3	0.5	4	2	3	1
100	3	10	3	0.5	4	2	3	1

but setting $\alpha = 3$ (which increases the nonlinearity), there is little difference in relative performance (Figure 7(b)), although the overall Z-values for all four measures increase somewhat. When the fraction of influential variables is increased to $\gamma = 0.5$, in Figure 7(c), the ordering changes slightly, with ξ now performing the best, followed closely by ν. The difference in ranking errors between the DGSM measures and μ^* and S_T is also markedly less. The same is true in Figure 7(d), in which the coefficients are changed to the low-interaction setting. In this case, ξ is the best performer, followed by the other measures in the same order.

One general feature of these results is that, at least in the G^* function, DGSM measures seem to be more efficient at ranking functions with strong interactions, given the wide gap between ξ and ν and the other two measures. Between the two DGSM measures themselves, there is little discernable difference, although ξ may have a slight advantage. Most likely the reason that DGSM measures perform much better than the other measures on the

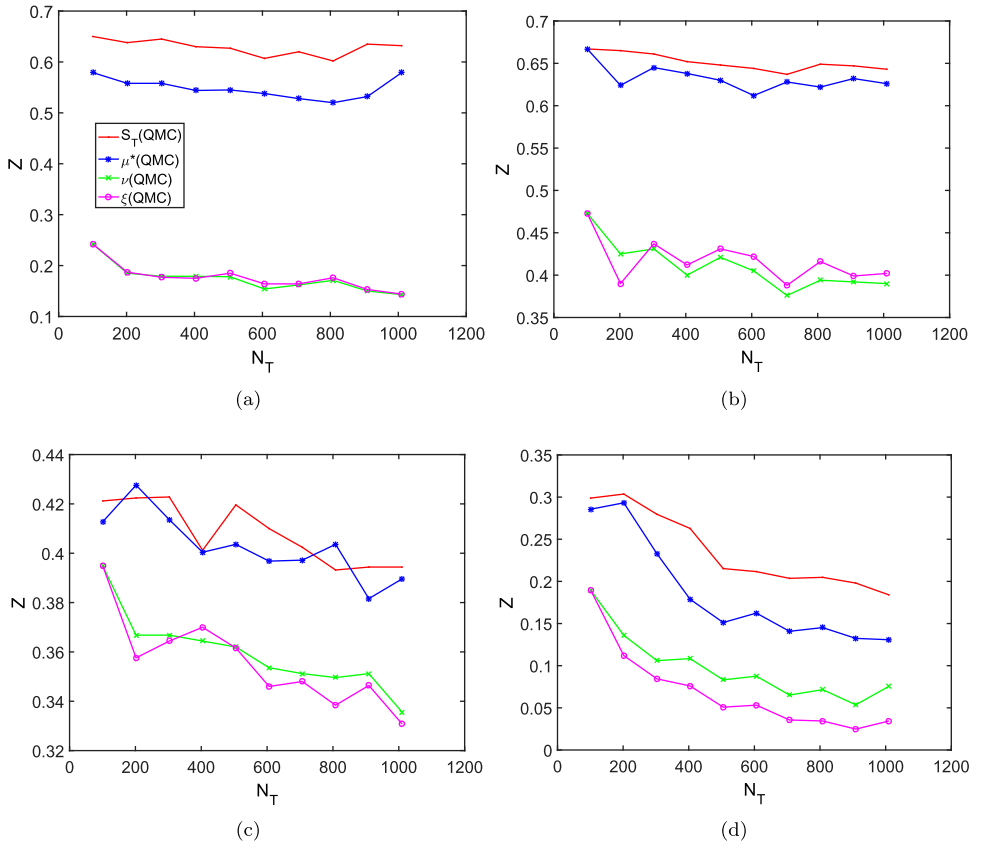

Figure 7. Convergence plots for G^* function: (a) $a_{\text{high}} = 1$, $a_{\text{low}} = 2$, $\alpha = 2$, $\gamma = 0.2$; (b) $a_{\text{high}} = 1$, $a_{\text{low}} = 2$, $\alpha = 3$, $\gamma = 0.2$; (c) $a_{\text{high}} = 1$, $a_{\text{low}} = 2$, $\alpha = 3$, $\gamma = 0.5$; (d) $a_{\text{high}} = 3$, $a_{\text{low}} = 10$, $\alpha = 2$, $\gamma = 0.5$; In all cases $k = 100$.

G^* function is that they are based on a sampling strategy that uses small steps. Given that the G^* function is non-monotonic, it is possible for the large-step samples of μ^* and S_T to 'miss' the high gradient of the function in a given dimension by sampling either side of the peak, which might output similar values. With the DGSM sample, this is much less likely to happen because the steps are very small, so the measure is better able to estimate the gradient of the function at a given point. A wider conclusion that can be drawn here then is that DGSM measures should be much more efficient when screening non-monotonic functions.

A related point here is that in the (piecewise) linear case, when $\alpha = 1$, the DGSM measures can perfectly capture the importance of the inputs, giving error values of $Z = 0$ for all values of N. This is because first, when a function is linear, the gradient at any point can be used to measure its global sensitivity, since the gradient does not change over the input space. Second, although the μ^* measure is also related to the gradient, because it measures across large steps it suffers from the problems related to non-monotonicity, as mentioned previously.

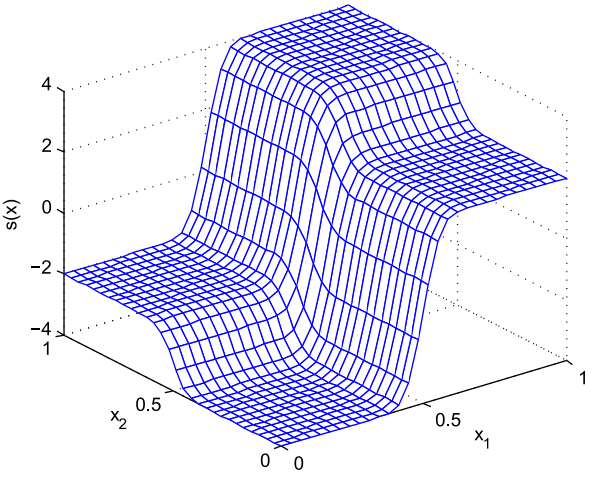

Figure 8. Plot of step function for $k = 2$, $a_{\text{high}} = 3$, $a_{\text{low}} = 1$, $\gamma = 0.5$.

Table 3. Configuration and performance rankings of experiments with step function.

k	a_{high}	a_{low}	γ	S_T	μ^*	ν	ξ
30	3	1	0.2	1	2	3	4
50	3	1	0.2	1	2	3	4
75	3	1	0.2	1	2	3	4
100	3	1	0.2	1	2	3	4
30	3	1	0.5	1	2	3	4
50	3	1	0.5	1	2	3	4
75	3	1	0.5	2	1	3	4
100	3	1	0.5	2	1	3	4

4.3. Step function

The final test function is a simple function with a near-discontinuity, of the form,

$$y = f(\boldsymbol{x}) = \sum_{i=1}^{k} a_i \text{erf}(15(x_i - 0.5)), \tag{11}$$

where erf is the error function. This function has a gradient of zero in most places, except around $x_i = 0.5$, at which point the gradient is very steep. This behaviour can be seen in Figure 8, where the function is plotted with $k = 2$ for illustration. For the numerical experiments, $a_{\text{high}} = 3$ and $a_{\text{low}} = 1$, with a fraction γ from 0.2 to 0.5 as shown in Table 3.

The step function was in fact chosen as a counter-example to show the limitations of the DGSM measures. Looking at Table 3, the results clearly reflect this. Contrary to the previous two functions, S_T performs the best at all the configurations tested except two, with the μ^* measure performing second-best, followed by ν and ξ.

More detail can be seen in Figure 9: S_T performs better in both plots (averaged over sample size), although only by a small margin compared to μ^*. The two DGSM measures perform very similarly to each other, with poor performance at lower sample sizes, but a rapid reduction in Z as sample size increases, such that by $N_T = 800$ they both give $Z = 0$. The reason for this ranking is the same as why the ranking is the opposite for the

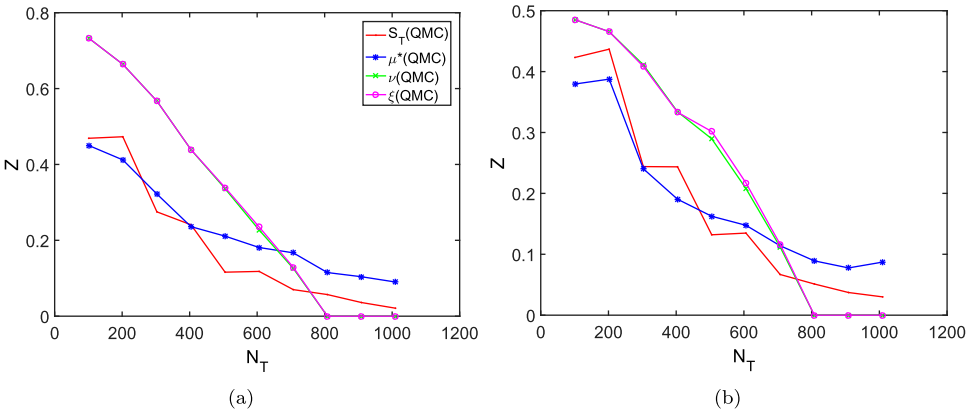

Figure 9. Convergence plots for step function: (a) $a_{\text{high}} = 3, a_{\text{low}} = 1, \gamma = 0.2$; (b) $a_{\text{high}} = 3, a_{\text{low}} = 1, \gamma = 0.5$; In all cases $k = 100$.

G^* function – the DGSM measures use small steps in each x_i direction, whereas the other two measures use large steps. The small steps are a disadvantage for a function such as the step function, because the sample points need to be very close to the 'discontinuity' to see a non-zero gradient (see again Figure 8). Indeed, if there were a true discontinuity, the DGSM measures would return sensitivity measures of zero unless the $\boldsymbol{x}_j$ and $\boldsymbol{x}_j^{i''}$ happened to fall either side of the discontinuity. The measures S_T and μ^*, on the other hand, are much more likely to see a change in the function output as a result of their larger steps.

4.4. Results with random sampling

The previous results have only shown the performance of the sensitivity measures using the Sobol' quasi-random sequence, since the relative performance of each is nearly identical to that when using random numbers. However it is worth briefly examining the differences between using random sampling and the Sobol' sequence. Figure 10 shows three selected plots. Clearly, in almost all cases, the use of the Sobol' sequence improves the performance of the sensitivity measures at a given sample size, relative to simple random sampling.

For the polynomial function, the difference is quite stark, with a practically error-free performance at sample sizes of around 800 upwards (except S_{Ti}, for 100 variables, for the Sobol' sequence, and significantly worse performance with random numbers. A lesser, but still significant difference is obtained with the G^* function, but only for the DGSM measures. Finally, the case of the step function is quite interesting. First, the rate of convergence is considerably improved for the DGSM measures, such that at higher sample sizes they actually perform the best. Strangely however, the use of the Sobol' sequence does not actually improve the performance of the S_T measure: on the contrary, it actually makes it worse.

Still, the overriding conclusion is that the Sobol' sequence performs better in almost all cases, and can lead to significant computational savings.

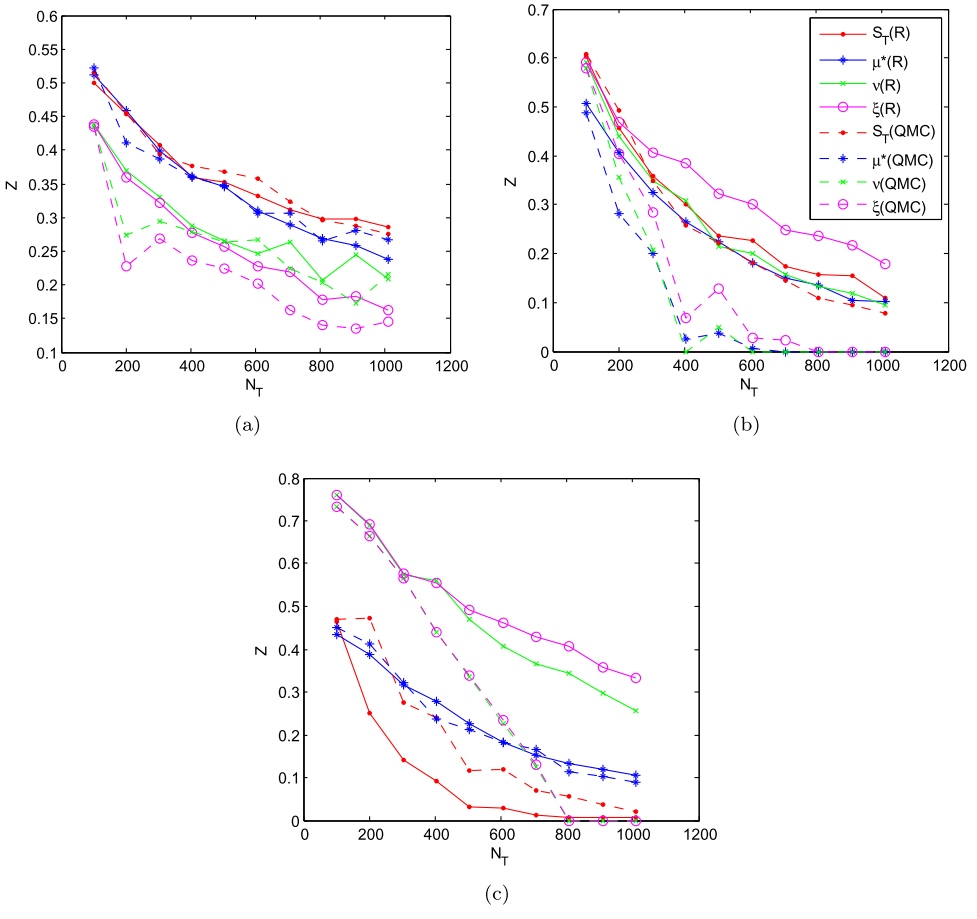

Figure 10. Convergence plots for three selected cases, all with $k = 100$ and $\gamma = 0.2$: (a) G^* function at $a_{\text{high}} = 3$, $a_{\text{low}} = 10$ and $\alpha = 3$; (b) polynomial function at $a_{\text{high}} = 2$, $a_{\text{low}} = 1$ and $p = 3$; (c) Step function at $a_{\text{high}} = 3$, $a_{\text{low}} = 1$.

5. Discussion and conclusions

By far the most evident conclusion of this work is that there is no 'one size fits all' solution to sensitivity analysis, and that the best-performing sensitivity measure in the screening context is very dependent on the type of function or model that is being analysed. This should come as no surprise to those who have experience in sensitivity analysis, or indeed in data analysis as a wider discipline, yet some studies still draw conclusions on the results of a single test function. It is clear from this work that a range of test functions with differing linearity, monotonicity, continuity and interactions should be investigated. Of course, our tests could be extended yet further, but on the basis of this work some useful conclusions can be drawn.

The first thing to note is that DGSM measures perform surprisingly well at low sample sizes, a feature which we do not believe has been specifically investigated to date. On functions such as the G^* function, they actually have a clear advantage, whereas on smoother functions such as the polynomial function, they exhibit comparable performance with the

μ^* measure. When the function is linear, or piecewise linear, they also provide a much faster convergence than the other measures, and also perform well in the presence of strong model interactions. Although the step function is a clear counter-example where DGSM measures have more difficulty, it is probably safe to say that this kind of response surface is not too common in physical models. In any case, the tests show that as sample size is moderately increased (around 800 model runs in 100 dimensions), DGSM measures actually perform better than other measures. Therefore, it is only in the very low sample sizes that they are worse than other measures in this particular example.

The total Sobol' index is not generally efficient at low sample size, and is outperformed by the DGSM measures and the μ^* measure in most cases, with the exception of the step function. It is not therefore recommended to be used as a screening tool.

The modified DGSM measure proposed in this work had some reasonable success – giving at least comparable performance to the standard DGSM measure, and possibly surpassing it depending on the function. However, the performance of the two was generally quite close, therefore, it is not possible to say that one is necessarily better than the other without a considerable amount of further testing.

Overall, DGSM measures would seem to be a very useful tool in a screening analysis, as long as their caveats are kept in mind. A safe strategy would be to try to estimate both DGSM measures *and*S_T, although this would require a larger sample which may be impractical.

A further general observation is that the rate of convergence for all the measures here is quite good – even with 100 variables, they can be mostly sorted into high- and low-influence groups with some few hundreds of runs. Additionally, the test functions here are designed to be taxing – in practice, many physical models do not exhibit strongly nonlinear behaviour. Substantial further computational savings can be obtained by using the Sobol' sequence in place of random numbers, and perhaps further by grouping variables – see, for example, Morris [9] and citing literature, and [19,34] in relation to derivative-based sensitivity measures.

In summary, this investigation indicates that sampling-based measures provide a solution to screening in high dimensions, at very low sample sizes; i.e. in cases where emulators would not be applicable. Derivative-based measures offer a particularly useful tool in this respect.

Note

1. Strictly speaking, these are *pseudo-random* numbers because computer algorithms cannot output truly random numbers, but they exhibit most of the properties of random numbers.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Sobol' IM. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math Comput Simul. 2001;55(1–3):271–280.
[2] Oakley J, O'Hagan A. Probabilistic sensitivity analysis of complex models: a Bayesian approach. J R Stat Soc B. 2004;66:751–769.

[3] Becker W, Oakley J, Surace C, et al. Bayesian sensitivity analysis of a nonlinear finite element model. Mech Syst Signal Process. 2012;32:18–31.
[4] Sudret B. Global sensitivity analysis using polynomial chaos expansions. Reliab Eng Syst Saf. 2008;93(7):964–979.
[5] Ratto M, Pagano A, Young P. State dependent parameter metamodelling and sensitivity analysis. Comput Phys Commun. 2007;177(11):863–876.
[6] Blatman G, Sudret B. Adaptive sparse polynomial chaos expansion based on least angle regression. J Comput Phys. 2011;230(6):2345–2367. Available from: http://www.sciencedirect.com/science/article/pii/S0021999110006856.
[7] Becker W. Applications of dynamic trees to sensitivity analysis. 2015 Jul; this collection contains the proceedings of ICASP12, the 12th International Conference on Applications of Statistics and Probability in Civil Engineering held in Vancouver, Canada on July 12–15, 2015. Abstracts were peer-reviewed and authors of accepted abstracts were invited to submit full papers. Also full papers were peer reviewed. The editor for this collection is Professor Terje Haukaas, Department of Civil Engineering, UBC Vancouver. Available from: https://open.library.ubc.ca/cIRcle/collections/53032/items/1.0076191, 2015.
[8] Sobol I, Tarantola S, Gatelli D, et al. Estimating the approximation error when fixing unessential factors in global sensitivity analysis. Reliab Eng Syst Saf. 2007;92(7):957–960. Available from: http://www.sciencedirect.com/science/article/pii/S0951832006001499.
[9] Morris MD. Factorial sampling plans for preliminary computational experiments. Technometrics. 1991;33(2):161–174.
[10] Campolongo F, Cariboni J, Saltelli A. An effective screening design for sensitivity analysis of large models. Environ Model Softw. 2007;22(10):1509–1518.
[11] Homma T, Saltelli A. Importance measures in global sensitivity analysis of nonlinear models. Reliab Eng Syst Saf. 1996;52(1):1–17.
[12] Kucherenko S, Rodriguez-Fernandez M, Pantelides C, et al. Monte carlo evaluation of derivative-based global sensitivity measures. Reliab Eng Syst Saf. 2009;94(7):1135–1148.
[13] Sobol' IM, Kucherenko S. Derivative based global sensitivity measures and their link with global sensitivity indices. Math Comput Simul. 2009;79(10):3009–3017.
[14] Tarantola S, Becker W, Zeitz D. A comparison of two sampling methods for global sensitivity analysis. Comput Phys Commun. 2012;183(5):1061–1072.
[15] Jansen MJW. Analysis of variance designs for model output. Comput Phys Commun. 1999;117(1–2):35–43.
[16] Saltelli A, Annoni P, Azzini I, et al. Variance based sensitivity analysis of model output, design and estimator for the total sensitivity index. Comput Phys Commun. 2010;181(2):259–270.
[17] Campolongo F, Saltelli A, Cariboni J. From screening to quantitative sensitivity analysis. A unified approach. Comput Phys Commun. 2011;182(4):978–988.
[18] Kiparissides A, Rodriguez-Fernandez M, Kucherenko S. Application of global sensitivity analysis to biological models. In: Computer aided chemical engineering. Bertrand Braunschweig and Xavier Joulia (Editors) Vol. 25. 18th European Symposium on Computer Aided Process Engineering – ESCAPE 18, Lyon, France. Elsevier; 2008. p. 689–694.
[19] Kiparissides A, Kucherenko S, Mantalaris A, et al. Global sensitivity analysis challenges in biological systems modeling. Ind Eng Chem Res. 2009;48(15):7168–7180.
[20] Molkenthin C, Scherbaum F, Griewank A, et al. Derivative-based global sensitivity analysis: Upper bounding of sensitivities in seismic-hazard assessment using automatic differentiation. Bull Seismol Soc Am. 2017;107(2):984–1004.
[21] Becker W, Rowson J, Oakley J, et al. Bayesian sensitivity analysis of a model of the aortic valve. J Biomech. 2011;44(8):1499–1506.
[22] Pujol G. Simplex-based screening designs for estimating metamodels. Reliab Eng Syst Saf. 2009;94(7):1156–1160.
[23] Pareto V. Manuale di economia politica. Vol. 13. Milan, Italy: Societa Editrice; 1906.
[24] Caflisch RE. Monte Carlo and quasi-Monte Carlo methods. Acta Numer. 1998;7:1–49.

[25] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 1979;21(2):239–245.
[26] Iman RL, Helton JC, Campbell JE. An approach to sensitivity analysis of computer models, part 1. Introduction, input variable selection an preliminary variable assessment. J Qual Technol. 1981;13(3):174–183.
[27] Owen AB. Latin supercube sampling for very high-dimensional simulations. ACM Trans Model Comput Simul.. 1998;8(1):71–102.
[28] Sobol IM. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys. 1967;7(4):86–112.
[29] Kucherenko S, Feil B, Shah N, et al. The identification of model effective dimensions using global sensitivity analysis. Reliab Eng Syst Saf. 2011;96(4):440–449.
[30] Cranley R, Patterson TNL. Randomization of number theoretic methods for multiple integration. SIAM J Numer Anal. 1976;13(6):904–914.
[31] Owen AB. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In: Monte carlo and quasi-monte carlo methods in scientific computing. Proceedings of a conference at the University of Nevada, Las Vegas, Nevada, USA, June 23–25, 1994 Editors: Niederreiter, Harald, Shiue, Peter J. (Eds.). Springer; 1995. p. 299–317.
[32] Morohosi H, Fushimi M. A practical approach to the error estimation of quasi-Monte Carlo integration. In: Monte Carlo Quasi-Monte Carlo Methods. 1998. Niederreiter, Harald; Spanier, Jerome (editors); Springer-Verlag New York, Inc. p. 377–390.
[33] Tuffin B. Technical report: On the use of low discrepancy sequences in monte carlo methods. I.R.I.S.A., Rennes, France; 1996 (Report No.: 1060).
[34] Sobol' IM, Kucherenko S. A new derivative based importance criterion for groups of variables and its link with the global sensitivity indices. Comput Phys Commun. 2010;181(7):1212–1217.