Design for sensitivity analysis, in Chapman and Hall "Handbook of Design of Experiments"

William Becker and Andrea Saltelli
Joint Research Centre,
Institute for the Protection and Security of the Citizen, Ispra

August 14, 2014

Contents

Chapter 18

Design for Sensitivity Analysis

18.1 Introduction

Sensitivity analysis is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input (Saltelli et al., 2004). Sometimes the term is also used to indicate simply the quantification of the uncertainty in the model's prediction, although strictly speaking this is the closely-related discipline of *uncertainty analysis*. In general, sensitivity analysis is used to test the robustness of model-based inference, i.e. how much the results of the model depend on the assumptions made in its construction, and in particular on the specification of model input values. In engineering and risk analysis, sensitivity analysis mostly involves an exploration of the multidimensional space of the input variables.

Sensitivity analysis may also take slightly different meanings dependent on the field: in econometrics, sensitivity analysis has been advocated first in the form of "extreme bounds analysis", measuring the sensitivity of regressor coefficients to the omission or inclusion of other regressors in a regression model (Leamer, 1985, 2010). A form of derivative-based sensitivity analysis is also used to check the sensitivity of regression models to misspecification (Magnus, 2007). In engineering, *design sensitivity analysis* uses the gradient of

the error function between a model output and experimental measurements to estimate unknown model parameters, such as the stiffness parameters in a structural model (Tortorelli and Michaleris, 1994). A succinct review of sensitivity analysis methods for use in impact assessment, i.e. in relation to models used for policy, is in Saltelli and D'Hombres (2010). In this chapter however, the focus will be on sensitivity analysis in the context of uncertainty in the inputs and outputs of a model.

Very often, in chemistry, physics, biology and so on, one sees sensitivity analysis performed by changing one input at a time, the so-called OAT design. This practice is not recommended because it only examines the sensitivity of model inputs at nominal (average) values, and does not allow the possibility of exploring the model response at other values of input variables, which could produce very different results if the model is nonlinear (Saltelli and Annoni, 2010). Instead, current best practice involves designs based on a multidimensional exploration of the space of the input variables, as in classic experimental design.

An important point to note from the start is that sensitivity analysis does not typically examine "structural uncertainty" , which is the uncertainty due to the model's approximation to reality. As such, the results of a sensitivity analysis, which relate to uncertainty in the input variables, are conditional on the model. Structural uncertainty can often represent the largest source of uncertainty, so it is important to recognise that a sensitivity analysis is only half the story in a thorough analysis of uncertainty. Techniques for managing structural uncertainty will not be addressed here, but two approaches are noted with some references as a starting point: first, "model ensemble averaging" (Tebaldi and Knutti, 2007; Rougier et al., 2013) which uses the results of a number of different but plausible models to approximate a "distribution over models" — an application to climate modelling can be found in Murphy et al. (2009). Second, an approach which considers the "discrepancy" between a single calibrated model and the true observed value (Kennedy and O'Hagan, 2001; Strong et al., 2012). A discussion of model uncertainty in a wider context can be found in Saltelli and Funtowicz (2013).

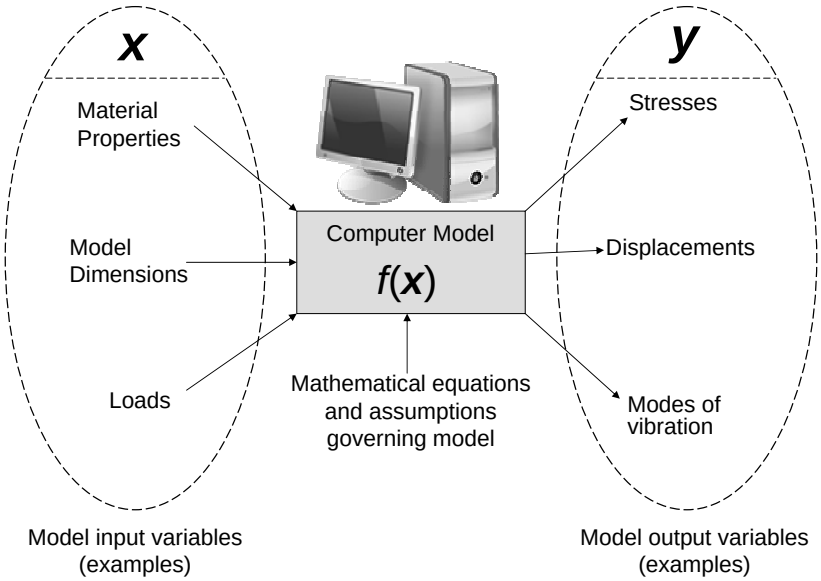


Figure 18.1: Black box view of an engineering computer model, with typical model inputs and outputs.

18.1.1 The Black Box Perspective

In this chapter, the term "model" refers to a computer program which represents a mathematical construct built to simulate some physical, economic or other "real-world" process — examples could be models to predict climate change, engineering models to analyse the response of a component under loading, or economic models forecasting the behaviour of markets.

Since any numerical model has quantifiable inputs and outputs, it is helpful in sensitivity analysis to consider it from the "black-box" perspective (see Figure 18.1). This views the model as a function $f(\boldsymbol{x})$ of k inputs, where $\boldsymbol{x} = \{x_i\}_{i=1}^{k}$. The model will typically return a large number of output quantities, but in this chapter, for simplicity it shall be assumed that the output is a scalar y, such that $y = f(\boldsymbol{x})$. Note that although the $\boldsymbol{x}$ and y will often appear as random variables, they will always be expressed in lower case. Importantly, the models in this chapter will be assumed to be deterministic, such that f is fixed, and the *structure* of the model does not contain random components. The uncertainty in the output

therefore is due uniquely to randomness in $\boldsymbol{x}$.

Although the function (model) f is known in the sense that it represents a computer program based on mathematical equations, it will generally be complex enough as to be only accessible via simulation (i.e. not analytically tractable). Therefore in practice, all sensitivity analysis approaches involve sampling the inputs a number of times according to an experimental design, evaluating the model for each selected input vector, and estimating useful properties from the resulting outputs/data. With this in mind, it is useful to think of a particular set of model input values $\boldsymbol{x}$ as a "point" in a k-dimensional hyperspace, which is bounded by the maximum and minimum values of each input variable.

18.1.2 Types of Problem

There is no "one size fits all" solution in sensitivity analysis, due to the fact that each problem has its own unique characteristics and challenges, such as a large number of model inputs, model nonlinearities, correlations or other relationships between model inputs and limitations in CPU time to name but a few. As such, there exist a great number of approaches that are designed for use under particular circumstances. Here a (non-exhaustive) taxonomy is attempted of some of the most common settings encountered in sensitivity analysis, and the tools that are available for each. One of the first defining features is whether or not the model is actually accessible to the analyst (the person performing the sensitivity analysis) — two main cases arise in this respect:

Case 1 The analyst can 'run' the model. In other words, the model is available to the analyst, such that it can be evaluated at chosen input values and the corresponding results recorded. A design can be specified in this case where, for example, n model input points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ are selected in the k-dimensional input space, to obtain corresponding model outputs $\{y_1, y_2, ..., y_n\}$. In this case the sample of the input space is customarily generated without correlation among the input variables, although designs for correlated inputs are also available (Xu and Gertner, 2008; Li et al., 2010; Kucherenko et al., 2012; Jacques et al., 2006). The output y could represent, for exam-

ple, some modeled property of an engineering design such as an aeroplane wing or of a natural system such as groundwater flow through a geologic region.

Case 2 The sample points are given and the analyst can neither control their positioning nor generate additional points. Such data might come either from measurements or experiments, or from a design that is not specifically intended for sensitivity analysis. The form of the model could be unknown, and the input variables could be correlated with one another in the sample. To give a simple example, y could be the Human Development Index computed over k countries and the x_i could be the indicators used in the construction of the index (Paruolo et al., 2013). In this case one cannot generate additional points/countries.

In Case 1 (when the design points can be specified) the best approach to performing a sensitivity analysis is determined by the cost of the model runs required to perform the analysis. In this context, "cost" refers to the total computational time required to evaluate the model at all the sample points, which is the product of the total number of model runs and the time required for each run. Since complex models can take minutes, hours or longer to evaluate for a single input point, it is not always feasible to sample a large number of input points, see e.g. Becker et al. (2011); Batterbee et al. (2011). The strategies available for case I are as follows:

Case 1A For "cheap" models (for which a single model evaluation will take a matter of seconds or less), a fully-fledged quantitative sensitivity analysis can be performed using Monte Carlo estimators, estimating all k "first order indices" and all k "total order indices" directly from model output values (see Section 18.3). This approach requires a large number of sample points (typically hundreds or thousands per input variable), but is preferred where possible since all sensitivity indices can be estimated with an accuracy related to the number of sample points. Furthermore, no assumptions are required about the functional form of the model (apart from that the model $f(\boldsymbol{x})$ is square-integrable, though this should not be a limitation in the vast majority of cases).

Case 1B For expensive models a design based on Fourier analysis can be used to compute all first order indices at a cost which is weakly dependent on the number of input vari-

ables (see Section 18.4). The cost is of the order of some hundreds of model simulations. Alternatively, a space-filling design can be used in conjunction with an emulator (see Section 18.6 and Chapter 17). Although computationally cheaper, both of these approaches introduce a data-modelling problem which involves making assumptions about the functional form of the model, such as smoothness and continuity. Additionally, the use of certain types of emulators becomes rapidly infeasible as the number of sample points and the dimensionality of the problem increase.

Case 1C In the case where the model is computationally expensive *and* one has many input variables, a set of methods known as *screening* can be applied to sort variables into influential and non-influential groups: this is known as *factor fixing* in some literature (Saltelli et al., 2008). A more detailed sensitivity analysis based on Monte Carlo or the use of emulators can then be applied on the set of influential variables, while non-influential variables are kept fixed (see Section 18.5).

A summary of Cases 1A-1C is shown in Figure 18.2: Monte Carlo methods scale well with dimensionality but need many runs per variable, so are not appropriate for expensive models. Emulators can deal with expensive models, but only for a limited number of input variables. In the case where the model has many input variables and is expensive to run, screening methods can help to reduce dimensionality to bring the problem in reach of emulators. Another possibility is to re-examine the model itself to see whether its run-time can be reduced, therefore allowing a Monte Carlo analysis. Although this is not a method of sensitivity analysis, it is an important consideration, since most models are not built with sensitivity analysis in mind, and therefore are not necessarily optimised for speed. However this requires access to the model and a deep understanding of it, which is not always possible.

For Case 2, when data are given, two approaches are considered in this chapter:

- Use an *emulator* either to generate additional points and then perform a Monte Carlo sensitivity analysis, or to directly estimate sensitivity indices from the available data (depending on the method – see Section 18.6).
- Estimate directly the k first order indices by univariate regression on the sorted model

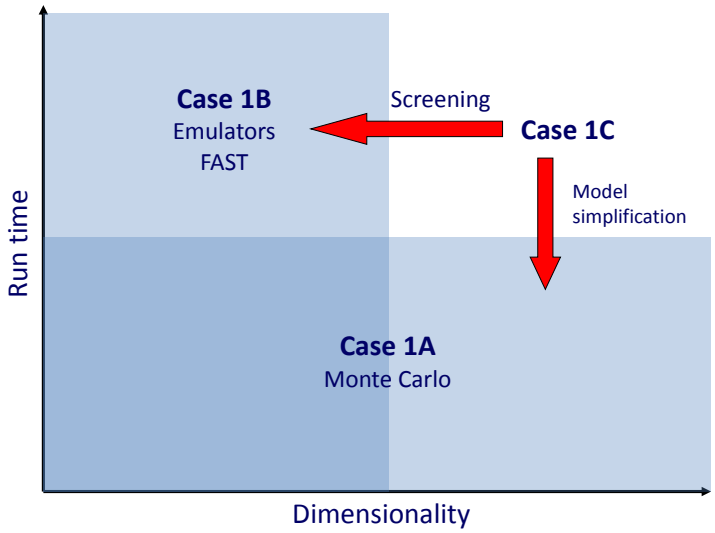


Figure 18.2: Case 1 sensitivity analysis problems based on dimensionality of model and run time.

evaluations $y_1, y_2, ..., y_n$ (see Section 18.7). In effect, this involves making one-dimensional scatter plots of y against each x_i, then fitting (nonlinear) trend curves. In simple problems, even a visual inspection of scatter plots may be useful for qualitative analysis.

The various approaches discussed here and the context in which they can be applied are summarised in Figure 18.3. Note that in the present chapter only three measures of sensitivity are proposed:

1. First order sensitivity index (see Section 18.2)
2. Total order sensitivity index (see Section 18.2)
3. Elementary effects (see Section 18.5)

The following section gives a brief description of variance-based sensitivity analysis which underpins measures 1 and 2 above, after which in Section 18.3 Monte Carlo numerical procedures will be described for estimating them (this is the domain of Case 1A). In Section 18.4 an alternative way of estimating the same measures is described, using an approach based on the Fourier series. Section 18.5 deals with screening approaches (see Case 1C), while Section

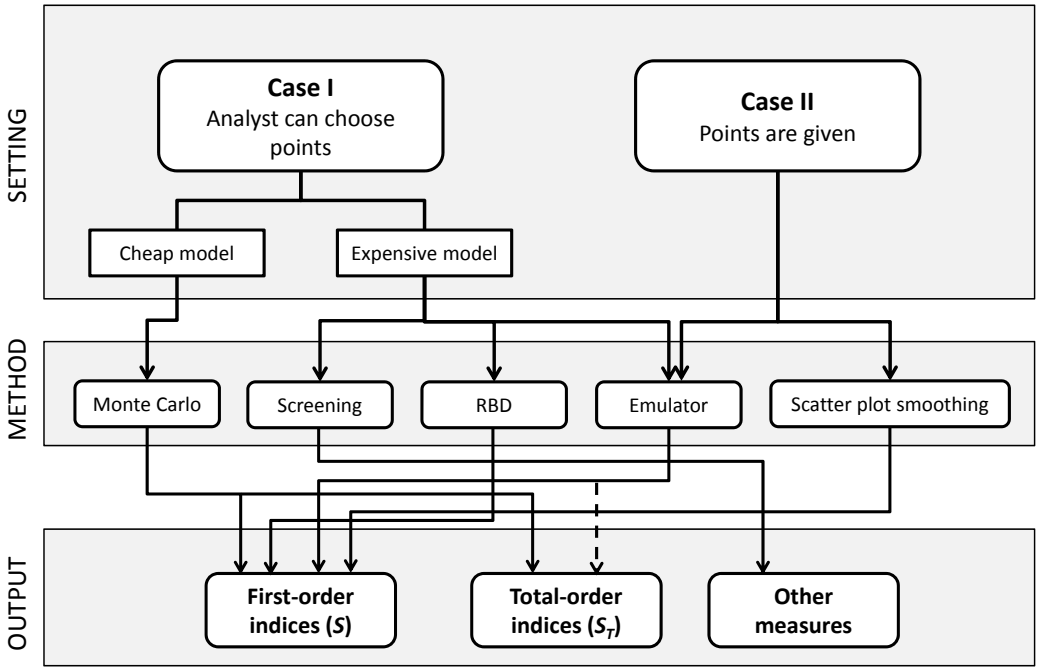


Figure 18.3: Various approaches to sensitivity analysis: when they can be used, and what they produce. The dotted line here indicates that S_{Ti} can only be estimated with certain types of emulators, and the acronym "RBD" stands for "Random Balance Designs".

18.6 explains the concepts of emulation and Section 18.7 scatterplot smoothing to deal with computationally expensive models (Case 1B), or the case where points are given (Case 2). Finally, some concluding remarks are given in Section 18.8.

18.2 Variance-Based Sensitivity Indices

Many measures of sensitivity have been proposed in the literature. For example, a well-known measure is to regress the data against each input variable x_i, and take the coefficients of determination R_i^2 as measures of sensitivity. An obvious drawback of this is that linear regression (linear in $\boldsymbol{x}$) can only meaningfully interpret a linear model response ("model response" refers to the effect on the model output of changing its inputs). While this approach can be extended by more sophisticated forms of regression, it is preferable not to rely on any modelling of a functional relationship between y and $\boldsymbol{x}$, since unwanted assumptions would thus be introduced. Variance-based approaches to measuring sensitivity, which consider the uncertainty in the model inputs and output from a probabilistic perspective, have become very popular in recent years, since they allow for highly nonlinear model responses, and account for variations in the output over the full input space.

In variance-based sensitivity analysis, it is required that the uncertainty in each model input $x_1, x_2, ..., x_k$ is characterised by known probability distributions $p_1(x_1), p_2(x_2), ..., p_k(x_k)$, and furthermore that these distributions are independent of one another, such that $\prod_{i=1}^{k} p_i(x_i) = p(\boldsymbol{x})$. In fact this first step may often pose major challenges to the analyst because it is rarely the case that enough information exists to characterise the input distributions to a great degree of confidence, and correlations are not uncommon in many problems. This chapter will not however address these difficulties since the focus is on the statistical aspects of sensitivity analysis as commonly practiced, although some brief discussion of the practical aspects of sensitivity analysis is given at the end in Section 18.8.

Given known and independent input distributions then, a useful sensitivity measure for a given input variable x_i is:

$$V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y \mid x_i)] \tag{18.1}$$

The meaning of the inner E operator is the expected value of the model output y taken over all possible values of variables other than x_i (i.e. over $\boldsymbol{x}_{\sim i}$), while keeping x_i fixed (the conditional mean). The outer V is the variance taken over all possible values of x_i.

The associated normalised sensitivity measure, known as a *first order sensitivity coefficient* is defined as:

$$S_i = \frac{V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y \mid x_i)]}{V(y)}. \tag{18.2}$$

The measure S_i gives the fraction of model output variance which is caused by the input x_i alone, averaged over variations in all other input variables. Formula (18.2) has a long history, the foundations having been laid by Karl Pearson in the form of the "product-moment correlation coefficient" in 1895 (Pearson, 1895) (a further discussion of this is given in Section 18.7). However, the use of partial variances (i.e. (18.2)) as measures of sensitivity in computer models appears to have been first proposed in 1973 in the context of the Fourier amplitude sensitivity test (Cukier et al., 1973), which uses a Fourier series representation of the model (see Section 18.4). The idea was further developed by Ilya Sobol' in 1990 who also introduced an approach to estimate S_i by the Monte Carlo method, thereby bypassing the need to use a Fourier series approximation (Sobol', 1993) (see Section 18.3).

The numerator of S_i is in fact the first term in a variance decomposition whereby the unconditional model output variance $V(y)$ is decomposed as the sum of a set of conditional variances of first, second, $\cdots$, up to the k^{th} order (Sobol', 1993). Such a decomposition holds only if the input variables x_i are independent, in which case,

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{i,j} + ... + V_{1,2,...,k}, \tag{18.3}$$

where:

$$V_i = V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)]$$

$$V_{i,j} = V_{x_i,x_j}[E_{\boldsymbol{x}_{\sim i,j}}(y|x_i,x_j)] - V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)] - V_{x_j}[E_{\boldsymbol{x}_{\sim j}}(y|x_j)]$$

and so on for the higher order terms. The terms in (18.3) derive from an analogous functional

decomposition of $f(\boldsymbol{x})$ into orthogonal functions of increasing dimensionality:

$$f(\boldsymbol{x}) = f_0 + \sum_i f_i(x_i) + \sum_i \sum_{j>i} f_{i,j}(x_i, x_j) + \ldots + f_{1,2,\ldots,k}(x_1, x_2, \ldots, x_k) \tag{18.4}$$

where $f_0 = E(y)$, $f_i = E_{\boldsymbol{x}_{\sim i}}(y|x_i) - f_0$, $f_{i,j} = E_{\boldsymbol{x}_{\sim i,j}}(y|x_{ij}) - f_i - f_j - f_0$, and so on. Taking the variance of (18.4) gives the variance decomposition in (18.3), noting that for example $V_{x_i}(f_i) = V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i) - E(y)] = V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)]$. A discussion of the importance of the first order terms $f_i = E_{\boldsymbol{x}_{\sim i}}(y|x_i) - E(y)$ is returned to in Section 18.7.

Dividing all terms in (18.3) by $V(y)$ gives:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{i,j} + \ldots + S_{1,2,\ldots,k} = 1. \tag{18.5}$$

where the S_i are the first-order sensitivity coefficients defined in (18.2), and higher-order terms are generalisations of these to multiple inputs. For example, $S_{i,j}$ measures the variance due to the interaction between x_i and x_j, additional to the variance caused by each input alone. A knowledge of all the sensitivity indices in (18.5) gives a detailed picture of how each input contributes to the uncertainty of the model output and the interactions between inputs in the model. Note that the case where $\sum_{i=1}^{k} S_i = 1$ is known as an *additive* model, in which there are no interactions between model inputs. This is however rarely the case in complex models, for which reason the calculation of the first order indices alone is not usually sufficient.

In the ideal case then, one would like to know all sensitivity indices of all orders in (18.5). Due to computational limitations however, estimating all terms in (18.3) is often impractical for larger k given that they number $2^k - 1$ in total. For this reason, a measure known as the the *total order sensitivity index*, S_T, may be estimated, which measures the total effect of an input, including its first order effect and interactions of any order (Homma and Saltelli, 1996):

$$S_{Ti} = 1 - \frac{V_{\boldsymbol{x}_{\sim i}}[E_{x_i}(y \mid \boldsymbol{x}_{\sim i})]}{V(y)} = \frac{E_{\boldsymbol{x}_{\sim i}}[V_{x_i}(y \mid \boldsymbol{x}_{\sim i})]}{V(y)} \tag{18.6}$$

where $\boldsymbol{x}_{\sim i}$ denotes the vector of all variables but x_i. In $E_{\boldsymbol{x}_{\sim i}}[V_{x_i}(y \mid \boldsymbol{x}_{\sim i})]$ the inner variance V of y, the scalar output of interest, is taken over all possible values of x_i while keeping $\boldsymbol{x}_{\sim i}$ fixed, while the output expectation E is taken over all possible values $\boldsymbol{x}_{\sim i}$ (Homma and Saltelli, 1996).

One can see that $E_{x_i}(y \mid \boldsymbol{x}_{\sim i})$ is the main effect of $\boldsymbol{x}_{\sim i}$, and therefore $V_{\boldsymbol{x}_{\sim i}}[E_{x_i}(y \mid \boldsymbol{x}_{\sim i})]$ is the variance caused by the main effects and interactions of all the variables and sets of variables *not* involving x_i. The remaining variance, $V(y) - V_{\boldsymbol{x}_{\sim i}}[E_{x_i}(y \mid \boldsymbol{x}_{\sim i})]$, is the variance due to all terms in the decomposition (18.3) including x_i, i.e. the variance of its main effect and all interactions of any order involving x_i, giving:

$$S_{Ti} = S_i + \sum_{j>i} S_{i,j} + \sum_{l>j>i} S_{i,j,l} + \cdots + S_{1,2,\ldots,k}. \tag{18.7}$$

As an example, consider a function of three input variables, $f(x_1, x_2, x_3)$. The standardised variance decomposition in (18.5) would in this case consist of:

$$S_1 + S_2 + S_3 + S_{1,2} + S_{1,3} + S_{2,3} + S_{1,2,3} = 1. \tag{18.8}$$

In this case, the S_{Ti} can be expressed as the sum of any indices involving the index i:

$$\begin{aligned} S_{T1} &= S_1 + S_{1,2} + S_{1,3} + S_{1,2,3} \\ S_{T2} &= S_2 + S_{1,2} + S_{2,3} + S_{1,2,3} \\ S_{T3} &= S_3 + S_{1,3} + S_{2,3} + S_{1,2,3} \end{aligned} \tag{18.9}$$

from which one can note that in general $\sum_{i=1}^{k} S_{Ti} \neq 1$, unless all the interaction terms are zero. Observe also that the S_{Ti} could in fact be calculated from evaluating and summing all component sensitivity indices as in (18.9), but in practice this is rarely done, since it involves a much higher computational effort – for this reason the expression in (18.6) is usually used as the basis for estimation unless one is particularly interested in the precise nature of the interactions.

In the next section the design and estimation procedures for the the cases detailed in

Section 18.1 are described.

18.3 Monte Carlo Estimation of Sensitivity Indices

Monte Carlo estimation of sensitivity indices is generally considered as the preferred approach to sensitivity analysis where possible, since it makes no assumptions about the functional form of the model (unlike emulators and FAST – see Sections 18.6 and 18.4). It is however only possible under the circumstances of Case 1, i.e. when the analyst has full control over the placement of input points and possibly thousands of model runs can be executed without difficulty. Monte Carlo estimation involves sampling the model at a large number of points in the input space using random or *quasi-random* numbers as a basis. In this section the use of quasi-random numbers is described, specifically the LP_τ sequences of Sobol' (Sobol', 1967, 1976) (also known simply as Sobol' sequences) coupled with a Monte Carlo design described in Section 18.3.2 (Saltelli, 2002; Saltelli et al., 2010). In the following, therefore, the focus will be on the Sobol' sequence. However, the approaches described are also valid with random numbers and other low-discrepancy sequences – see Niederreiter (1992) for a summary of many common approaches.

18.3.1 Input Distributions and Sampling

It is assumed here that all random variables $x_1, x_2, \ldots, x_k$ are sampled uniformly in the k-dimensional unit hypercube $\mathcal{X}$;

$$\boldsymbol{x} \in \mathcal{X} : \ \mathcal{X} = [0,1]^k. \tag{18.10}$$

Different distributions can easily be generated by mapping the points in (18.10) onto the desired probability density function (uniform, normal, log-normal, etc). This involves the use of the inverse cumulative distribution function of the variable of interest (also known as the *quantile function*), which allows uniformly-distributed points in $[0,1]$ to be transformed into points distributed as required (Saltelli et al., 2008). An example is shown in Figure

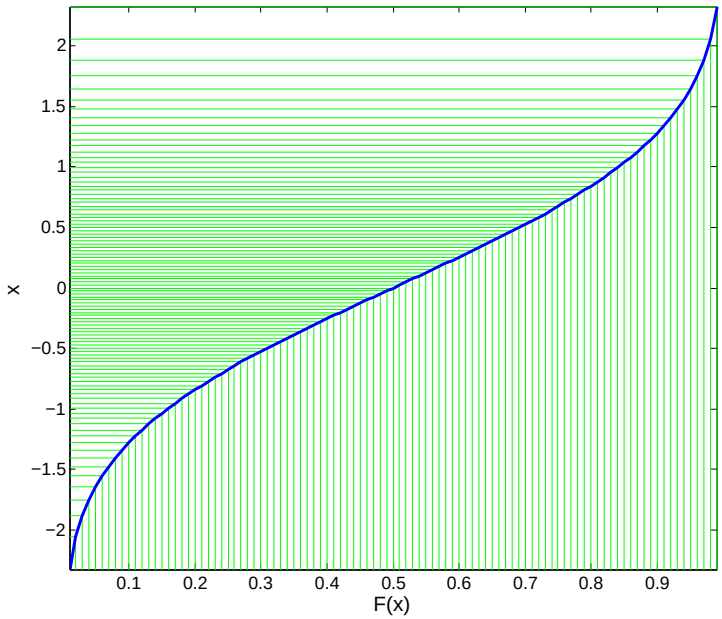
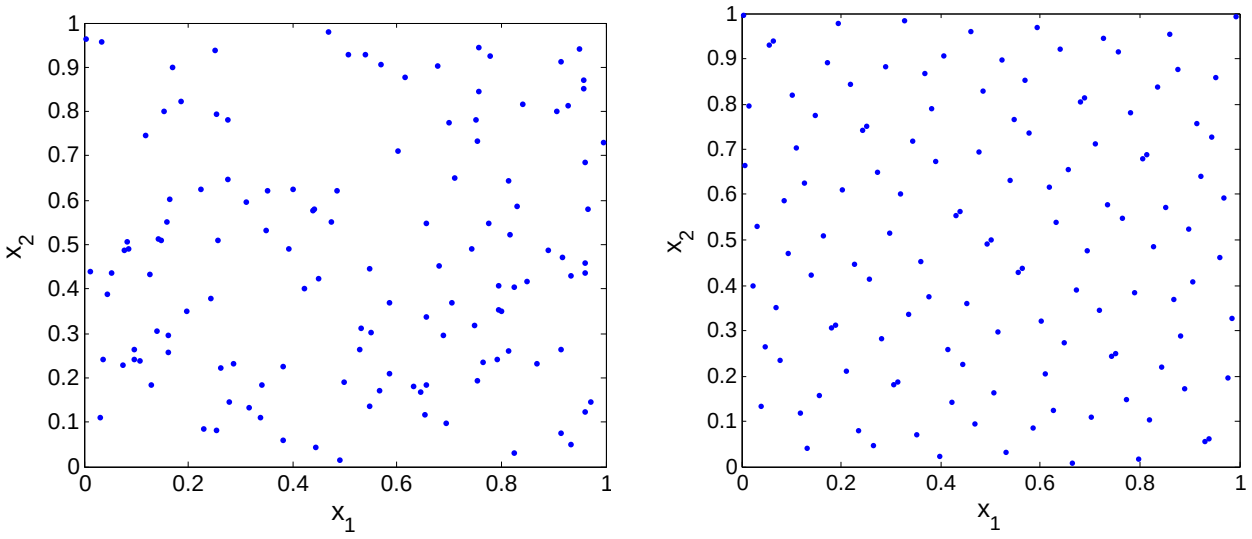


Figure 18.4: Generating normally-distributed points using a equally-spaced points and the inverse cumulative distribution of the normal distribution. Vertical and horizontal lines illustrate the mapping of sample points from one distribution to another.

18.4 in which a set of equally-spaced points are transformed into normally-distributed points (equally-spaced points are used here rather than random sampling to more clearly illustrate the transformation).

The Monte Carlo estimators presented in the following section rely on the use of random or quasi-random numbers — in particular, the approach recommended in this chapter is to use the Sobol' sequence. The Sobol' sequence and other quasi-random number sequences (also known as "low-discrepancy sequences") are fixed sequences of numbers which are designed to fill hypercubes as uniformly as possible – in the context of sensitivity analysis they can be used as a list of model input values that explore the model response with a high efficiency. Figure 18.5 shows a comparison of the Sobol' sequence against random (strictly speaking *pseudo-random*) numbers – observe the clusters and large "holes" in the random design compared to the relatively well-spaced points in the Sobol' design. The use of quasi-random sequences is motivated by their good space filling properties; these sequences outperform both pseudo-random Monte Carlo sampling as well as Latin Hypercube Sampling (LHS) in the estimation of multi-dimensional integrals (Sobol' and Kucherenko, 2005) – for details on LHS, see Chapter 19. Recent extensive testing with a large variety of functions spanning



(a) (b)

Figure 18.5: 128 points in 2D space: (a) random numbers; (b) Sobol' sequence

different degrees of dimensionality, linearity and additivity has demonstrated their suitability for sensitivity analysis (Kucherenko et al., 2011). An additional desirable property of Sobol' sequences when compared to LHS is that with the former, additional points can be added sequentially to the analysis until a desired target accuracy is achieved (note the points follow increasingly fine divisions of the input space – see the first four rows of Figure 18.6 for an example of the first four points in a 6-dimensional Sobol' sequence). With LHS, the sample size cannot be extended once the analysis is performed, without starting again from the beginning, because the positioning of all points is dependent on the sample size. Sobol' sequences can be generated using freely available software both in FORTRAN, and Matlab –see European Commission (2012).

18.3.2 Steps for Estimating Sensitivity Indices

The steps needed to estimate a full set of first order and total order sensitivity indices via the Monte Carlo method are as follows (see Figure 18.6 for an illustration of the construction of the matrices):

1. Generate n points of a $2k$-dimensional Sobol' sequence as in Figure 18.6, such that it is arranged in a $n \times 2k$ matrix with each row giving the coordinates of each point in the sequence. Call the first k-column submatrix $\mathbf{A}$ (i.e. the first k columns), and the remaining k-column submatrix $\mathbf{B}$. The generic coordinates of $\mathbf{A}$ and $\mathbf{B}$ can be indicated respectively as $x_{ji}^{(a)}$ and $x_{ji}^{(b)}$, where the index i runs from one to k (the number of input variables), and the index j runs from one to n, the number of rows[1].

2. Generate an additional k matrices $\mathbf{A}_B^i$, $i = 1, 2, ..., k$, such that the i^{th} matrix is entirely composed of coordinates from $\mathbf{A}$ except for its i^{th} column, which is the i^{th} column of $\mathbf{B}$. A total of $k + 2$ sets of coordinates (matrices) have thus been generated.

3. Evaluate the computer model for each of the $n(k + 2)$ input vectors generated as the rows of each of the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A}_B^i$, $i = 1, 2, ..., k$.

4. Compute the sample mean $\hat{f}_0$ of output associated with rows from both matrices of quasi-random points $\mathbf{A}$ and $\mathbf{B}$ combined, i.e. using $\hat{f}_0 = \frac{1}{2n}\sum_{j=1}^{n}\left(f(\mathbf{A})_j + f(\mathbf{B})_j\right)$ where, for example, $f\left(\mathbf{A}\right)_j$ indicates values of y computed from running the model f using the input values given by row j of matrix $\mathbf{A}$. The unconditional sample variance is also calculated using the unbiased estimator $\hat{V}(y) = \frac{1}{2n-1}\sum_{j=1}^{n}[(f(\mathbf{A})_j - \hat{f}_0)^2 + (f(\mathbf{B})_j - \hat{f}_0)^2]$.

5. To estimate S_i (see (18.2)) one needs first to estimate $V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y \mid x_i)]$. Denoting $V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y \mid x_i)] = V_i$, model outputs associated with coordinates from $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A}_B^i$ are used in the following estimator (Saltelli, 2002; Sobol' et al., 2007; Saltelli et al., 2010):

$$\hat{V}_i = \frac{1}{n}\sum_{j=1}^{n} f\left(\mathbf{B}\right)_j \left(f\left(\mathbf{A}_B^i\right)_j - f\left(\mathbf{A}\right)_j\right), \tag{18.11}$$

   A rationale for estimator (18.11) is given below. S_i is estimated by dividing (18.11) by the sample variance $\hat{V}(y)$.

6. For S_{Ti} one needs first to estimate $E_{\boldsymbol{x}_{\sim i}}[V_{x_i}\left(y \mid \boldsymbol{x}_{\sim i}\right)]$ (see (18.6)). Letting $E_{\boldsymbol{x}_{\sim i}}[V_{x_i}\left(y \mid \boldsymbol{x}_{\sim i}\right)] = V_{Ti}$, this can be estimated using model evaluations from the couple $\mathbf{A}$ and $\mathbf{A}_B^i$ (Jansen,

[1]Note that in this chapter, although j is in general an index over samples, it is also sometimes used to index over input variables, for example in the variance decomposition in (18.3). The meaning should however be clear given the context.

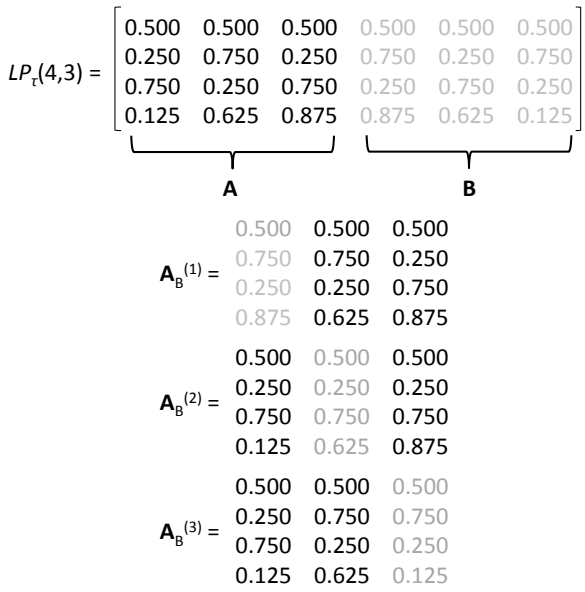


Figure 18.6: Construction of the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A}_B^i$ matrices, using the Sobol' LP$_\tau$ sequence with $k = 3$ and $N = 4$. Grey columns correspond to those taken from the matrix $\mathbf{B}$.

1999):

$$\hat{V}_{Ti} = \frac{1}{2n} \sum_{j=1}^{n} \left(f\left(\mathbf{A}\right)_j - f(\mathbf{A}_B^i)_j \right)^2, \tag{18.12}$$

with a similar meaning of symbols as above. Again, S_{Ti} is estimated by dividing (18.12) by the sample variance $\hat{V}(y)$.

Note that each matrix $\mathbf{A}_B^i$ is used twice for estimating sensitivity indices associated with x_i, once to compute $\hat{S}_i$ and once to compute $\hat{S}_{Ti}$. A derivation of estimators (18.11) and (18.12) can be found in Saltelli et al. (2008, 2010) – these designs are also called "substituted column sampling" (Morris et al., 2008). One can notice that the estimators make use of sums of products of model output values, and that in each product the two function values being multiplied by one another have some symmetry. In the case of $\hat{S}_i$ the two function values $f\left(\mathbf{B}\right)_j$ and $f\left(\mathbf{A}_B^i\right)_j$ have identical values for coordinate x_i, whereas in the case of $\hat{S}_{Ti}$ the two function values $f\left(\mathbf{A}\right)_j$ and $f\left(\mathbf{A}_B^i\right)_j$ have identical values for all coordinates except x_i. Take the case of $\hat{S}_i$ for illustration: if x_i is influential, then the two function values being multiplied, $f\left(\mathbf{B}\right)_j$ and $f\left(\mathbf{A}_B^i\right)_j$, will be correlated, such that high values will tend be multiplied by high values and low values by low values. The resulting sum of these products

will tend to be greater than the sum of the products of $f(\mathbf{B})_j$ and $f(\mathbf{A})_j$ (the two terms of which are uncorrelated), giving a value of $\hat{S}_i$ greater than zero. In contrast, if x_i is non-influential, high and low values of $f(\mathbf{B})_j$ and $f(\mathbf{A}_B^i)_j$ will be randomly coupled, resulting in an estimation of S_i which will tend to zero.

To see where the estimators (18.11) and (18.12) come from, refer back to (18.2) and (18.6). In the case of S_i, the numerator $V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)]$ can be expressed as,

$$V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)] = \int E^2_{\boldsymbol{x}_{\sim i}}(y|x_i)dx_i - \left(\int E_{\boldsymbol{x}_{\sim i}}(y|x_i)dx_i\right)^2 \tag{18.13}$$

using the variance identity $V(y) = E(y^2) - E^2(y)$. The second term in (18.13) reduces to $E^2(y)$ (since $E[E(y|x_i)] = E(Y)$) which is denoted as f_0^2 (refer back to (18.4)). The first term can be written as the following,

$$\begin{aligned}\int E^2_{\boldsymbol{x}_{\sim i}}(y|x_i)dx_i &= \int E_{\boldsymbol{x}_{\sim i}}(y|x_i)E_{\boldsymbol{x}_{\sim i}}(y|x_i)dx_i \\ &= \int \left(\int\int f(\boldsymbol{x}_{\sim i}, x_i)f(\boldsymbol{x}'_{\sim i}, x_i)d\boldsymbol{x}_{\sim i}d\boldsymbol{x}'_{\sim i}\right)dx_i \\ &= \int\int f(\boldsymbol{x}_{\sim i}, x_i)f(\boldsymbol{x}'_{\sim i}, x_i)d\boldsymbol{x}d\boldsymbol{x}'_{\sim i}\end{aligned} \tag{18.14}$$

Now one can see that the integral in (18.14) can be estimated by Monte Carlo integration using the first product $f(\mathbf{B})_j f(\mathbf{A}_B^i)_j$ of (18.11).

Finally it is worth briefly examining the experimental design generated in Steps 1 and 2 of this section. In the description given, matrices have been used to facilitate the programming of this procedure. However one should note that any row of the sample matrix $\mathbf{A}_j$ is simply a (quasi) random point in the input space, and the corresponding row $(\mathbf{A}_B^i)_j$ describes a point which has the same coordinates, except for the ith input variable, which takes its value $x_{ji}^{(b)}$ from the jth row and ith column of $\mathbf{B}$. Considering the set of points $\{\mathbf{A}_j, (\mathbf{A}_B^1)_j, (\mathbf{A}_B^2)_j, ..., (\mathbf{A}_B^k)_j\}$, one can see that this subset of the design forms a "star" in the input space, with a centre point $\mathbf{A}_j$ and each subsidiary point $(\mathbf{A}_B^i)_j$ a step away in the x_i direction. The example design given in Figure 18.6 is plotted for illustration in Figure 18.7. One can see that the design is nothing more than a number of OAT designs replicated

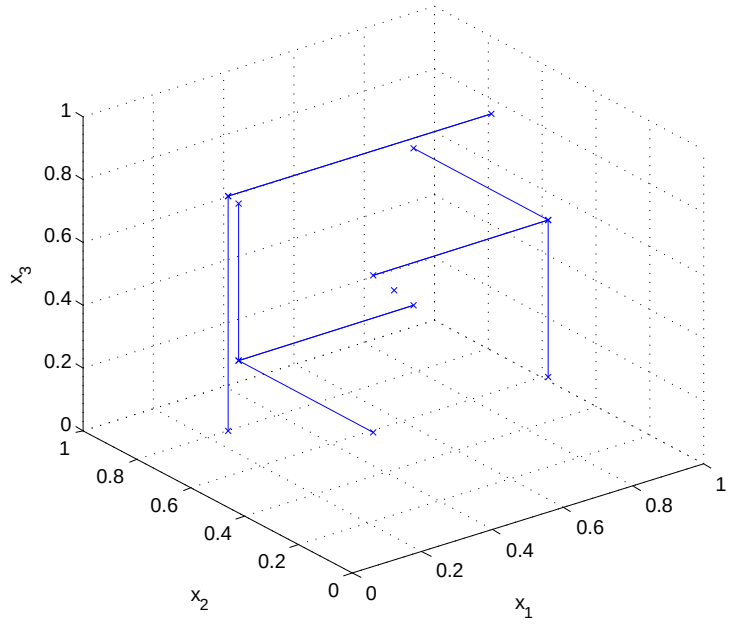


Figure 18.7: Sample design used for estimating sensitivity indices - example points as in Figure 18.6.

at various locations in the input space. However, by performing multiple OAT experiments one can begin to understand the "global" behaviour of the model — that is to say, the sensitivity of the model averaged over the full input space. Note that screening methods also use replicated OAT designs – see Section 18.5. A shortcoming of the use of the Sobol' sequence is in fact evident in Figure 18.7 — one can see that the top left "star" is missing a step in the x_2 direction, and another has no steps at all. Going back to the design in Figure 18.6 the reason can be understood: the coordinate values in the Sobol' sequence tend to repeat, which results in some instances where a coordinate value is substituted with the same number, resulting in $\mathbf{A}_j = \mathbf{A}^i_{B,j}$. These duplicates can however be accounted for, for example, by excluding them from the design when running the model (to avoid unnecessary runs), then adding the $f\left(\mathbf{A}^i_B\right)_j = f\left(\mathbf{A}\right)_j$ values where necessary. A further discussion on this point in the context of screening is given in Section 18.5.3.

18.3.3 Example

To show how the Monte Carlo estimators described above perform at different values of n, consider a simple polynomial example,

$$y = 3x_1^2 + 2x_1x_2 - 2x_3; \tag{18.15}$$

where the coefficients have been chosen quite arbitrarily. To illustrate the behaviour of the function with respect to its inputs, Figure 18.8 shows the scatter plots of y against x_1, x_2 and x_3, using random uniform sampling over $[0, 1]^3$. It is evident that x_1 has quite a strong, slightly nonlinear effect on y. Variable x_2 has apparently quite a weak effect (there is little discernable trend), whereas x_3 has a slight negative effect. These trends are clearly reflected in the coefficients of (18.15) – of course, normally one would not have the coefficients of an analytical equation to examine. The analytical values of S_i and S_{Ti} are given in columns 3 and 5 of Table 18.1.

To estimate the sensitivities of the variables, a Sobol' design is created in 3 dimensions, assuming uniform distributions for x_1, x_2, x_3 for simplicity, and estimators (18.11) and (18.12) are used. The only choice is what value of n, the number of sample points, to use. Given that the Sobol' sequence allows sequential addition of new points, one can start with a small number of points, then gradually increase until numerical convergence is observed. Figure 18.9 shows the convergence of these measures with n ranging from 8-1024. It is evident that the estimators converge quite quickly to an accurate estimate of the sensitivity indices; even at the lowest n, the variables are already correctly sorted, and at $n \geq 128$ the indices have converged to two decimal places. For most applications of sensitivity analysis, this would be sufficient accuracy. Table 18.1 shows the results at $n = 128$ compared to analytical values. Note that since there is a weak interaction between x_1 and x_2, the S_{Ti} of these variables is slightly higher than their respective S_i values, due to the fact that both S_{T1} and S_{T2} additionally include the interaction effect $S_{1,2}$ (refer back to (18.8) and (18.9) to see why). The value of $S_{1,2}$ is not estimated here, though it can be deduced from the table, noticing that x_3 does not interact with any variables since $S_3 = S_{T3}$ (and hence $S_{1,3} = S_{2,3} = S_{1,2,3} = 0$), therefore $S_{1,2} = S_{T2} - S_2 = S_{T3} - S_3$.

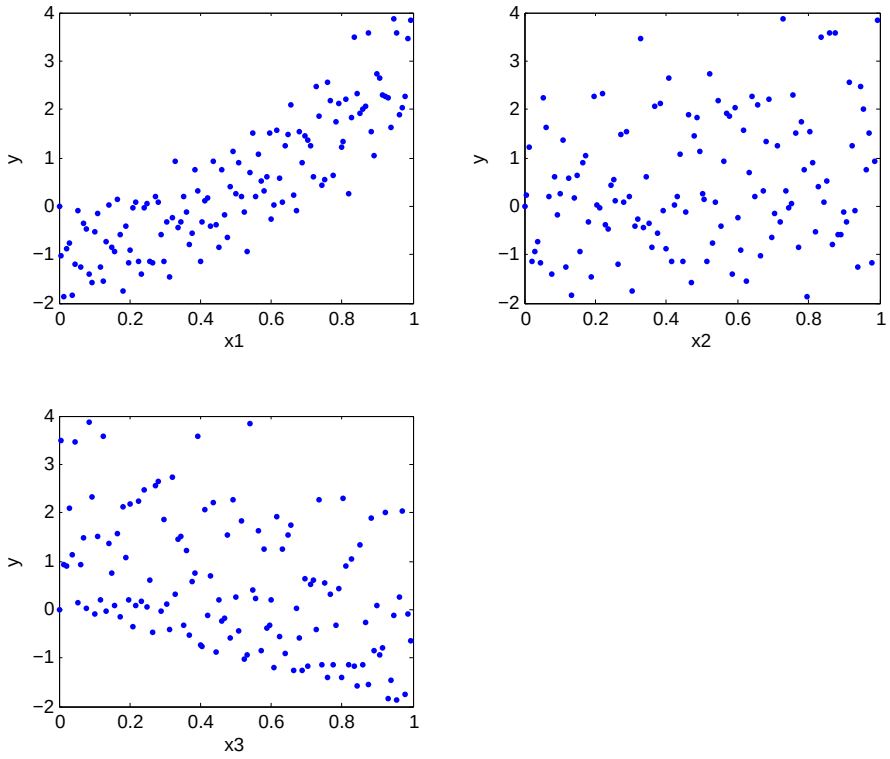


Figure 18.8: Scatter plots of the variables in the test equation (18.15).

Variable	$\hat{S}_i$ (MC)	S_i (analytic)	$\hat{S}_{Ti}$ (MC)	S_{Ti} (analytic)
x_1	0.7517	0.7568	0.7781	0.7720
x_2	0.0503	0.0456	0.0604	0.0608
x_3	0.1870	0.1824	0.1829	0.1824

Table 18.1: Monte Carlo estimates and analytical values of S_i and S_{Ti} of polynomial function with $n = 128$.

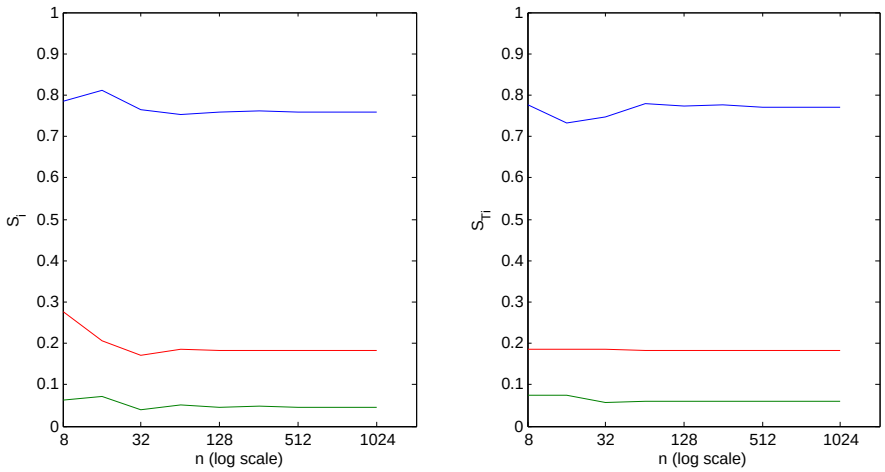
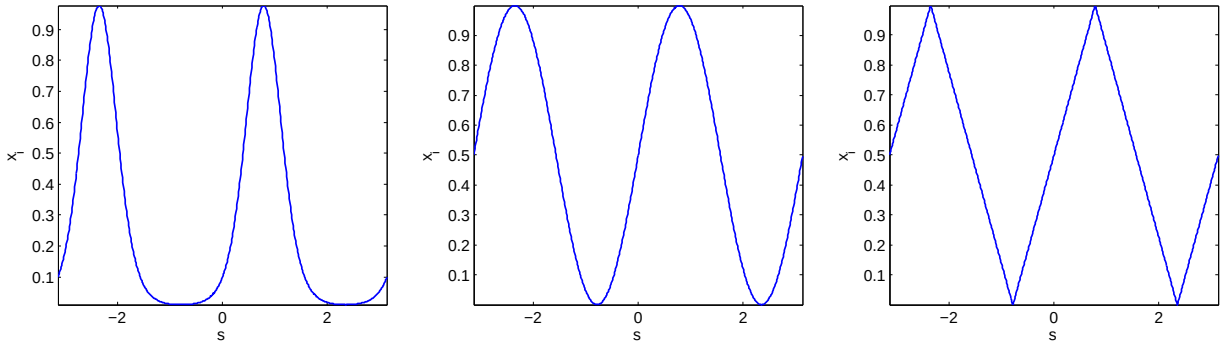


Figure 18.9: Convergence of the S_i and S_{Ti} of the polynomial equation (18.15) with increasing n. Lines represent, from top to bottom, x_1, x_3 and x_2 respectively.

Despite the flexibility of Monte Carlo estimators, one should remember that the cost is $n(k+2)$ model runs (see again Figure 18.6) – i.e. in the previous example the total number of model runs required was $128 \times 5 = 640$. While this is fine for fast models, for large models which are slower to run it may be impractical. In the following sections some alternative approaches are discussed that have lower computational requirements.

18.4 FAST and the Random Balance Design

The Fourier Amplitude Sensitivity Test (FAST) (Cukier et al., 1973, 1978), which was actually proposed around 20 years before the Monte Carlo estimators described in the previous section, uses a transformation of the input variables of the function (model) to represent a multivariate function as a periodic function of a single "frequency variable", s, which allows the function to be analysed using the Fourier series. The transformation of each input variable into s uses a unique characteristic sampling frequency which allows the contribution of each input to be assessed using the tools of Fourier analysis, which give analytical expressions for variance-based sensitivity indices (based on the Fourier series approximation). The advantage, compared to the Monte Carlo method, is that the integrals required to calculate the sensitivity indices, mean and variance (which are k-dimensional, see for example



(a) (b) (c)

Figure 18.10: Examples of transformations from s to x: (a) Equation (18.17), (b) Equation (18.18), (c) Equation (18.19).

(18.14)) can be expressed as univariate integrals with respect to s. Thus, a full set of S_i can be estimated from a single FAST sample, which means that the computational cost can be lower. However, the FAST approach relies on using the Fourier series to approximate the model output, which requires assumptions of smoothness, and furthermore uses truncated series to estimate sensitivity indices, which introduces estimation bias. A hybrid approach combines the concept of FAST with *random balance designs*, a form of experimental design first proposed by Satterthwaite (1959) – this is described in Section 18.4.4. In the following, a description of the transformation functions is given in Section 18.4.1, followed by the estimation of sensitivity indices in Section 18.4.2. A recent overview of FAST literature can be found in Xu and Gertner (2011).

18.4.1 Sampling Designs and Transformation of variables

In order to apply the tools of Fourier analysis to the model, each input variable x_i is transformed into a *periodic function* of a single variable, s, in the following way:

$$x_i = G\left(\sin(\omega_i s), \right) \quad i = 1, 2, ..., k, \tag{18.16}$$

where s is a variable in $[-\pi, \pi]$, G is a specified transformation function, and ω_i is an integer. The effect of this transformation is that uniformly sampling s within its range (i.e. taking

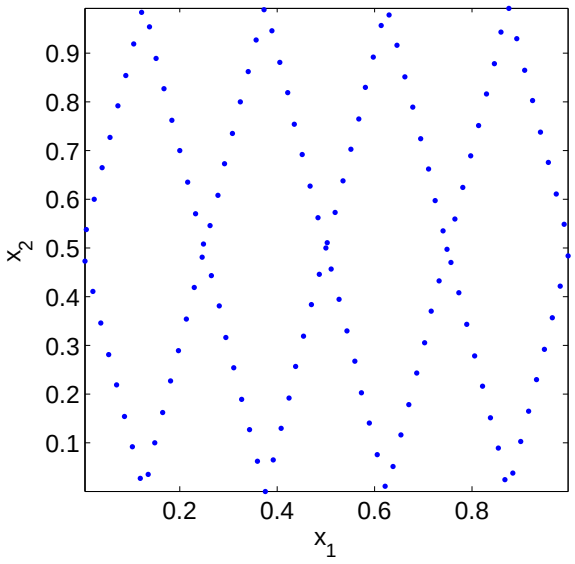


Figure 18.11: A 2D example of a FAST search curve using a triangular basis function (18.19) and with $\omega_1 = 1$ and $\omega_2 = 4$.

equally spaced values) results in oscillations in the corresponding sampled values of each x_i over its respective range. Figure 18.10 shows this effect on the following three transformation functions:

$$x_i = a_i \exp(b_i \sin(\omega_i s)) \tag{18.17}$$

$$x_i = a_i \left(1 + b_i \sin(\omega_i s)\right) \tag{18.18}$$

$$x_i = \frac{1}{2} + \frac{1}{\pi} \arcsin\left(\sin(\omega_i s)\right) \tag{18.19}$$

proposed respectively by Cukier et al. (1973), Koda et al. (1979) and Saltelli et al. (1999) as functions that are intended to approximate uniform sampling over the sample space $\mathcal{X}$. Note however, that functions (18.17) and (18.18) give higher densities of points at the edges of the sample space, therefore for truly uniform sampling, (18.19) is preferred. For non-uniform input distributions, other transformation functions would be necessary. The parameters a and b can be altered to give different ranges over x_i, for example in Figure 18.10 they have been adjusted to give $x_i \in [0, 1]$. Note also that by varying ω_i one can control the number of oscillations over the range of x_i, therefore ω_i represents the "search frequency" of each variable.

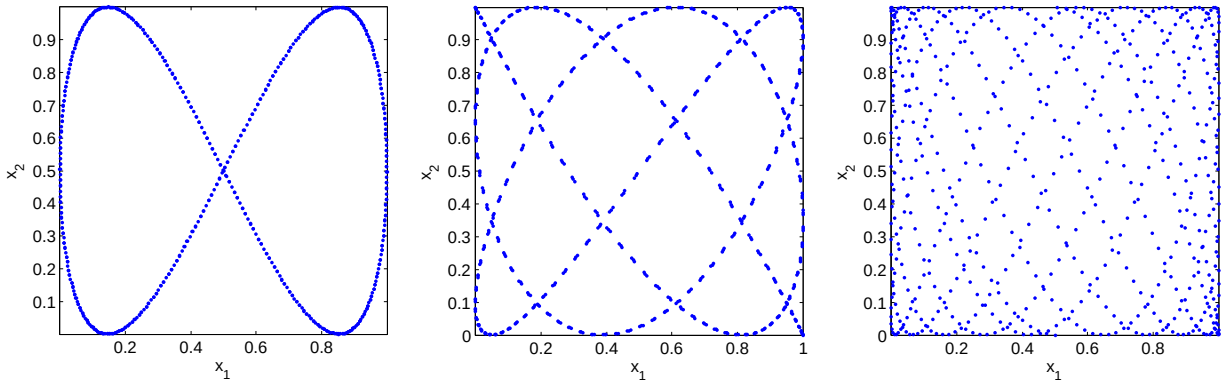


Figure 18.12: Search curves in two-dimensional space using transformation function (18.18): (a) $\omega_1 = 10$, $\omega_2 = 20$; (b) $\omega_1 = 15$, $\omega_2 = 21$; (c) $\omega_1 = 10$, $\omega_2 = 21$.

When all input variables are determined as functions of s, sampling uniformly over s produces samples along a "search curve" over the input space of the x_i. Figure 18.11 shows as an example the search curve produced for two input variables, using the triangular transformation given in (18.19), with $\omega_1 = 1$ and ω_2=4. Notice that taking evenly-spaced values of s over $[-\pi, \pi]$ results in values of x_1, x_2 that oscillate once over the range of x_1 and four times over the range of x_2.

Clearly, a desirable property of the search curve is that it should be able to explore the input space as efficiently as possible; in other words to generate a space-filling design. Given a choice of transformation function, the extent to which the sampled points along the search curve fill the input space is dependent on the choices of the ω_i, and n, the number of sample points. To illustrate this, Figure 18.12 shows the points generated by transforming s into two input variables with three different sets of search frequencies. Although in all three cases n is the same, the space-filling properties of the three curves are very different. In particular, when the two frequencies share common factors (such as in Figure 18.12(a), where $\omega_1 = 10$ and $\omega_2 = 20 = 2\omega_1$), the input space is explored very poorly because the curve repeats itself with a period of $\pi/5$. In contrast, in Figure 18.12(c), where $\omega_1 = 10$ and $\omega_2 = 21$, the only common factor is 1, which results in a better search curve because the points are unique over the whole range of s. The problem of frequency selection is not as simple as simply choosing, for example, large prime number values of ω_i, because more sample points are then required

to adequately represent the search curve due to the Nyquist criterion – this is explained in a little more detail in the following section. In fact, the choice of the ω_i requires a balance between higher frequency and lower sample size. Sets of ω_i that optimise the space-filling properties of the search curve, for given dimensionality and sample size, can be found in Schaibly and Shuler (1973).

18.4.2 Calculation of sensitivity index estimates

Given the transformation of variables, $f(\boldsymbol{x})$ is now expressed as a function $f(s)$ which is periodic over 2π. As such it can be expanded as a Fourier series:

$$f(s) = \frac{A_0}{2} + \sum_{r=1}^{\infty} \left(A_r \sin(rs) + B_r \cos(rs)\right) \tag{18.20}$$

with coefficients given as,

$$\begin{aligned} A_r &= \tfrac{1}{2\pi} \textstyle\int_{-\pi}^{\pi} f(s) \cos(rs) ds \\ B_r &= \tfrac{1}{2\pi} \textstyle\int_{-\pi}^{\pi} f(s) \sin(rs) ds, \end{aligned} \tag{18.21}$$

which can also be viewed as the "amplitudes" of $f(s)$ at a given frequency r, or in other words, the contribution of frequency r to the function. This provides the basis for FAST — the coefficients of the Fourier terms at $r = \omega_i$ can be interpreted as a measure of sensitivity, because if the function output has a strong component at frequency ω_i, this implies that it is strongly affected by input x_i. The corresponding estimators for the coefficients in (18.21) are:

$$\begin{aligned} \hat{A}_r &= \tfrac{1}{n} \textstyle\sum_{j=1}^{n} f(s_j) \cos(rs_j) \\ \hat{B}_r &= \tfrac{1}{n} \textstyle\sum_{j=1}^{n} f(s_j) \sin(rs_j) \end{aligned} \tag{18.22}$$

The use of the Fourier series allows expressions for the variance and partial variances of the output y. The expression for $V(y)$ can be given in terms of the coefficients of the Fourier series as follows (Koda et al., 1979),

$$\begin{aligned} V(y) &= \tfrac{1}{2\pi} \textstyle\int_{-\pi}^{\pi} \left(f^2(s) ds - E(y)^2\right) ds \\ &= 2 \sum_{r=1}^{\infty} (A_r^2 + B_r^2). \end{aligned} \tag{18.23}$$

In practice this expression is truncated to a maximum frequency R and is therefore an approximation to the variance. Similarly, the partial variance of y, $V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)]$ (as discussed in Section 18.2), can be expressed in terms of the coefficients that correspond to the frequency ω_i and its multiples (harmonics) $p\omega_i$, where p is a positive integer (Koda et al., 1979):

$$V_{\omega_i} = 2\sum_{p=1}^{\infty}\left(A^2_{p\omega_i} + B^2_{p\omega_i}\right), \tag{18.24}$$

which will again be truncated to a maximum order $p = M$, i.e. the Mth harmonic (multiple) of ω_i. The calculation of first-order sensitivity indices proceeds by noting that $S_i = V_{\omega_i}/V(y)$. A problem with (18.24) is that of interference: for higher values of p there will inevitably exist some pth harmonic of ω_i that is the same frequency as a qth harmonic of ω_j (the sampling frequency of another input variable x_j), such that $p\omega_i = q\omega_j$. This means that the coefficients of this frequency would be counted in both S_i and S_j, resulting in estimation bias. However, the amplitudes of higher harmonics in (18.24) generally decrease, so that if the ω_i are carefully chosen, any interferences up to the truncation order will be minimal.

As with Monte Carlo methods, the precision of the FAST estimates increases as n, the number of sample points, increases. For FAST, n should be at least $2R\omega_{\max} + 1$, where $\omega_{\max}$ is the highest frequency considered in the estimators (18.23) and (18.24), a limit which is imposed by the Nyquist criterion (Nyquist, 1928). The required n can therefore become quite high as k increases, since higher frequencies are required to avoid interference. FAST additionally suffers from a number of sources of bias – first, that the Fourier series used to approximate the model must be truncated, and second that for any given set of ω_i there will be points in the input space that can never be sampled no matter how many sample points are used. Although the Fourier series is known to converge to any periodic function that is square-integrable (Carleson, 1966) (the same property as the Monte Carlo estimators), the existence of bias suggests that the Monte Carlo method should generally be preferred when possible. Some further discussion on bias in FAST can be found in Xu and Gertner (2011); Tissot and Prieur (2012).

18.4.3 Extended FAST

An extension to FAST was proposed in Saltelli et al. (1999), which additionally allows the estimation of the total effect indices discussed in Section 18.2. It proceeds by the observation that the set of frequencies *not* in the set of search frequencies and their harmonics (i.e. $\omega \notin \{p\omega_i\};\ p = 1, 2, 3, ...;\ i = 1, 2, ..., k$) contains information on the residual variance that is unaccounted for by first order effects, which means that it contains information on the variance due to interactions of any order between variables (see again (18.3)). In the configuration described above, however, there is no obvious way to attribute this residual variance to interactions of particular inputs.

The proposal (known as "Extended FAST") is therefore to use a high frequency ω_i for the ith variable, then to assign a set of low frequencies to all the remaining variables, such that their frequencies and harmonics $\{p\omega_j\}_{p=1}^{M},\ j \neq i$, will be lower than $\omega_i/2$. The result is that the information about the output variance explained by the $\boldsymbol{x}_{-i}$ variables, including all interactions between them, is isolated in the frequencies below $\omega/2$. By summing the variances from all these frequencies, an estimator for $V_{\boldsymbol{x}_{\sim i}}[E_{x_i}(y \mid \boldsymbol{x}_{\sim i})]$ is obtained, which can be directly used to estimate S_{Ti} – see (18.6).

The obvious drawback to this method is that, whereas the estimation of all the S_i can be performed with one search curve, to estimate all the S_{Ti} requires k search curves. However, the approach has still been shown to be of at least comparable efficiency to the Monte Carlo approach given in Section 18.3 (Saltelli et al., 1999), although the sources of bias discussed in the previous section mean that the Monte Carlo method may still be preferable in some cases.

18.4.4 Random Balance Designs

Since different frequencies must be used to investigate each variable, the computational cost of FAST quickly rises with k (albeit less than with the Monte Carlo approach), because higher frequencies are required to avoid interferences between harmonics. Due to the Nyquist

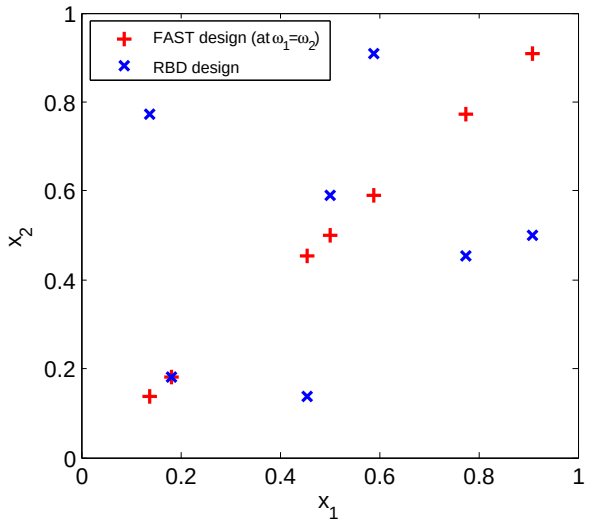


Figure 18.13: A 2D example of a RBD sample before and after scrambling of coordinates, with $\omega_1 = \omega_2 = 1$.

criterion, this requires more sample points. The Random Balance Design (RBD) approach to FAST (Tarantola et al., 2006) circumvents this to some extent by using a single frequency ω for all inputs. With no further modifications here, the points would be very poorly distributed in the sample space - in fact they would be limited to a straight line (see Figure 18.13). The key to RBD, which was first proposed as an approach to experimental design back in 1959 Satterthwaite (1959); Budne (1959), is to then take random permutations of the coordinates of these points. As an example, seven points for two input variables are generated using transformation (18.18). Scrambling involves independently randomly permuting columns, for example as follows,

$$\begin{array}{cc} x_1 & x_2 \\ 0.50 & 0.50 \\ 0.18 & 0.18 \\ 0.14 & 0.14 \\ 0.45 & 0.45 \\ 0.77 & 0.77 \\ 0.91 & 0.91 \\ 0.59 & 0.59 \end{array} \quad \Longrightarrow \quad \begin{array}{cc} x_1 & x_2 \\ 0.50 & 0.59 \\ 0.18 & 0.18 \\ 0.14 & 0.77 \\ 0.45 & 0.14 \\ 0.77 & 0.45 \\ 0.91 & 0.50 \\ 0.59 & 0.91 \end{array}$$

where the first two columns represent the sample before scrambling, and the last two show the sample after scrambling. These points are also illustrated in Figure 18.13 – notice that

the scrambled points fill the input space quite well because the design is very similar to a Latin hypercube (see Chapter 19).

In order to calculate sensitivity indices, the model is run at the points given by the random balance design, and then for a given input i the points are sorted into increasing order with respect to x_i, which recreates a periodic function of x_i. Then the first-order sensitivity indices can be calculated using (18.22)-(18.24) in the same way as the standard FAST method. The advantage of RBD is that, since the same frequency can be used for all inputs (which can be low, e.g. $\omega = 1$), the number of sample points required is less than conventional FAST. However, the drawback is that when estimating the effect of x_i, the random scrambling of the input variables generates random noise in the signal of x_i, resulting in a biased estimation of S_i. A bias-corrected estimator for sensitivity indices of any order has in fact been proposed by Tissot and Prieur (2012), where a more detailed discussion on bias in RBD can also be found. Readers might also want to refer back to the original extensive discussion on bias and other issues surrounding RBD (in the general context) after Satterthwaite's original paper (Youden et al., 1959). RBD has also been extended to compute total effect indices – readers are referred to Mara (2009) for more details.

18.5 Screening Designs

In the case where the user has access to the model, but it is expensive to run and there are a large number of input variables (Case 1C, see Section 18.1.2), screening methods offer a computationally-efficient way of identifying influential and non-influential variables. In this setting, it is not possible to use emulators (see Section 18.6) because they become increasingly expensive to fit as the number of model inputs increases, so are not suitable for high-dimensionality problems. Typically, screening methods are used in the "factor fixing" setting, i.e. to eliminate uninfluential variables before applying a more informative analysis to the remaining set (thus reducing the dimensionality – see again Figure 18.2), although they can also be used as a sensitivity analysis in their own right. The reason a user might prefer a variance-based measure however, is because screening measures do not have a clear

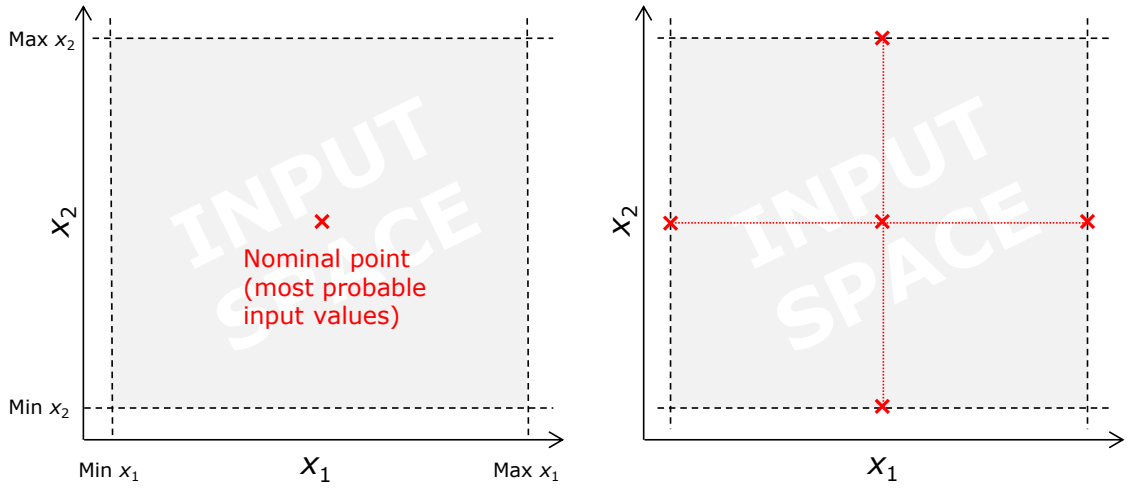


Figure 18.14: Basic OAT design in two dimensions; red crosses indicate design points

interpretation in terms of contribution the output variance, in contrast to S_i and S_{Ti}.

The most common screening approach, suggested by Morris (1991), is an extension of a basic method of sensitivity analysis, the one-at-a-time (OAT) design. Basic OAT designs simply involve varying one input whilst keeping all other inputs fixed at nominal (mean) values (see Figure 18.14). While this gives some information about the behaviour of the model, it is a very limited analysis because the sensitivity of input variables is only seen at the nominal value of the remaining inputs, and the extremities of the input space (i.e. the corners in Figure 18.14) are not explored. Morris's approach, which will be called the "elementary effects method" overcomes this problem by performing a number of OAT designs at random locations in the input space, rather than being restricted to nominal values. In this way nonlinearities and interactions can be accounted for by observing how the sensitivity of an input variable varies when moving about the input space.

18.5.1 Winding stairs design

The elementary effects design begins by dividing the range of each input into M equally-spaced intervals (M being chosen by the analyst), such that the input space is divided into a grid of points. The design then proceeds by selecting a random point from this grid

as a starting point, then moving in steps of Δ in each coordinate direction, where Δ is a predetermined multiple of $1/(M-1)$. This design was stated in general terms by Morris (1991), but one implementation, known as the "winding stairs design" in some literature (Campolongo et al., 2011) is given here as an example. The design proceeds as follows:

1. A random point on the grid is selected as the first design point.

2. The first input, x_1, is varied by an amount Δ_j, which is a randomly chosen multiple of the intervals in the grid, while keeping all other inputs fixed. This is the second design point.

3. Using the previous design point as a starting point, the next input, x_2 is varied by the same amount Δ_j, keeping all other inputs fixed.

4. Step 3 is repeated for all k inputs.

5. Steps 1-4 are repeated for $j = 1, 2, ..., n$ (n is selected by the user).

This design results in n trajectories, each consisting of $k+1$ points (see Figure 18.15 for an example in three dimensions with $n = 5$). Because each trajectory is built by moving just one variable at a time, it is essentially a form of an OAT design. However, in the elementary effects design, there are n OAT designs in each x_i direction, at different points in the input space, which is much more informative than a single OAT design. The efficiency of the winding stairs design can be improved slightly by using the last point of the jth trajectory as the starting point of the $(j+1)$th trajectory. This would then form a single continuous trajectory with cost $nk+1$ instead of $n(k+1)$ in the case of the design stated here in Steps 1-5, although this may come at the expense of exploring the input space less thoroughly. Other variations of these OAT designs are discussed by Morris (1991), including "clustered" designs which can use fewer sample points to calculate the same number of elementary effects — readers are referred to the original article for more details.

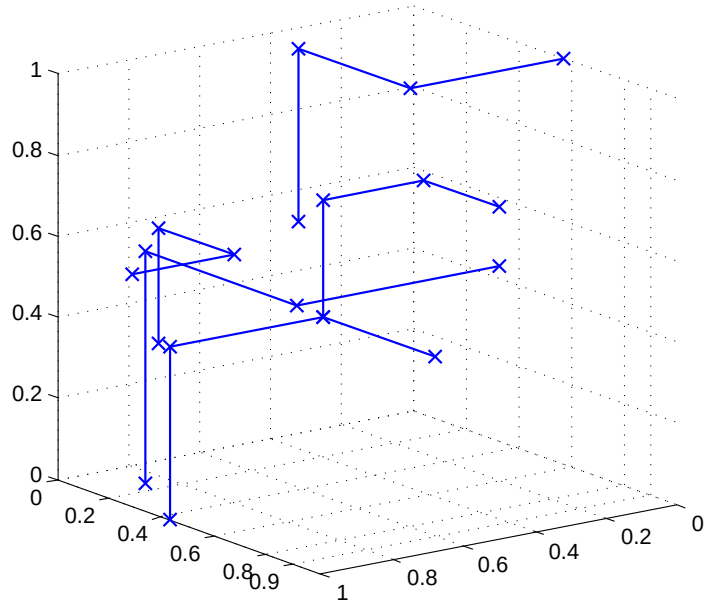


Figure 18.15: A trajectory screening design in three dimensions, with $n = 5$.

18.5.2 Measures

Let $\boldsymbol{x}_j^{(i)}$ 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. The point $\boldsymbol{x}_j^{(i)}$ will therefore be the preceding point to $\boldsymbol{x}_j^{(i')}$ in the five steps just described. Sensitivity is then estimated for the ith input using the n "elementary effects" $\{\xi_{ji}\}_{j=1}^{n}$, where

$$\xi_{ji} = \frac{\boldsymbol{x}_j^{(i')} - \boldsymbol{x}_j^{(i)}}{\Delta_j} = \frac{\boldsymbol{x}_j^{(i')} - \boldsymbol{x}_j^{(i)}}{|x_{ji}^{(i')} - x_{ji}^{(i)}|} \tag{18.25}$$

where j is the index over trajectories. The first measure of sensitivity for the ith input is thus estimated as the mean of the ξ_{ji},

$$\hat{\mu}_i = \frac{1}{n}\sum_{j}^{n} \xi_{ji}. \tag{18.26}$$

A further useful measure of nonlinearity and interaction is given by the variance σ_i^2 of the elementary effects, which is estimated as follows:

$$\hat{\sigma}_i^2 = \frac{1}{n-1}\sum_{j}^{n}(\xi_{ji} - \hat{\mu}_i)^2. \tag{18.27}$$

The logic here is that if the response of the output to a given input were perfectly linear, the elementary effects would be identical anywhere in the input space (and hence $\hat{\sigma}_i^2$ would be zero); in the nonlinear case the opposite would be true.

A drawback with the sensitivity measure given in (18.26) is that if the main effect of an input is non-monotonic, the average of the elementary effects may be close to zero even though, individually, they may be significant positive or negative values. The result is that the measure $\hat{\mu}_i$ could potentially miss influential variables (although one would observe a high value of $\hat{\sigma}_i^2$). A modified measure μ^*, proposed in Campolongo et al. (2007) suggests the use of the mean of the absolute values of the elementary effects, i.e.

$$\hat{\mu}_i^* = \frac{1}{n}\sum_{j}^{n}|\xi_{ji}|. \tag{18.28}$$

By using $\hat{\mu}_i$, $\hat{\mu}_i^*$ and $\hat{\sigma}_i^2$ in conjunction, one can assemble a picture of the strength and nature of the effect of each input at a low computational cost.

18.5.3 Radial design

A drawback of the winding stairs design is that there is no guarantee that the trajectories are well-spaced, and that the input space has been well-explored given the number of runs. A glance at the design in Figure 18.15 shows that points can sometimes be close to one another, therefore inefficiently exploring the input space. An alternative implementation of this design uses a so-called "radial" configuration based on Sobol's LP_τ sequence to achieve a screening design with better-spaced trajectories (Campolongo et al., 2011). This design is in fact almost exactly the same as that used in variance-based sensitivity analysis, but will be repeated here for clarity.

To construct the radial design, an LP_τ sequence of n points in $2k$ dimensions is generated and written as an $n \times 2k$ array (k being the number of model inputs). Let the first k columns be called the "baseline points", i.e. n points in k dimensions which will be denoted as a matrix $\mathbf{A}$, with rows (individual points) $\{\mathbf{A}_j\}_{j=1}^n$. The remaining k columns are called the n "auxiliary" points in a matrix $\mathbf{B}$ of the same size, with rows $\{\mathbf{B}_j\}_{j=1}^n$. For a given baseline point $\mathbf{A}_j$ and auxiliary point $\mathbf{B}_j$, a radial configuration of $k+1$ points is constructed as the following,

$$
\begin{gathered}
A_{j,1}, A_{j,2}, A_{j,3}, ..., A_{j,k} \\
B_{j,1}, A_{j,2}, A_{j,3}, ..., A_{j,k} \\
A_{j,1}, B_{j,2}, A_{j,3}, ..., A_{j,k} \\
A_{j,1}, A_{j,2}, B_{j,3}, ..., A_{j,k} \\
\vdots \\
A_{j,1}, A_{j,2}, A_{j,2}, ..., B_{j,k}
\end{gathered}
$$

where e.g. $A_{j,1}$ is the first coordinate of $\mathbf{A}_j$, the jth row of $\mathbf{A}$. This configuration is repeated for $j = 1, 2, ..., n$, resulting in n sets of $k+1$ points: a total of $n(k+1)$ model runs altogether. Notice that each set of $k+1$ points (as shown above) defines a "star" with $\mathbf{A}_j$ as its centre and the other points at steps in each coordinate direction, defined by the coordinates of $\mathbf{B}_j$. Figure 18.17 shows an example of the radial design in three dimensions, with $n = 5$, and Figure 18.16 shows the construction using the $\mathbf{A}$ and $\mathbf{B}$ matrices. Comparing Figures 18.15 an 18.17 one can see the strengths of each design: the spacing between trajectories/stars is better in the radial design because the centre points are guaranteed to be well-spaced, being drawn from the Sobol' sequence. On the other hand, the spacing of points within each trajectory/star is arguably better with the winding stairs design, since points are allowed to differ in more than one variable. Which of these properties is more important, and whether there is perhaps an advantage in combining the two designs, is left as an open question.

Since in Sobol' sequences the values of coordinates tend to repeat (see the example of the sequence in Figure 18.6), it is recommended that the baseline point $\mathbf{A}_j$ is not paired with $\mathbf{B}_j$, but rather with a different row, otherwise there could be no perturbation in certain dimensions (this problem is also mentioned in Section 18.3). It has been suggested that pairing $\mathbf{A}_j$ and $\mathbf{B}_{j+\delta}$, where $\delta = 4$ gives good results (Campolongo et al., 2011), although

A

Sob(7,6) =

0.500	0.500	0.500	0.500	0.500	0.500
0.250	0.750	0.250	0.750	0.250	0.750
0.750	0.250	0.750	0.250	0.750	0.250
0.125	0.625	0.875	0.875	0.625	0.125
0.625	0.125	0.375	0.375	0.125	0.625
0.375	0.375	0.625	0.125	0.875	0.875
0.875	0.875	0.125	0.625	0.375	0.375

B

Radial design =

0.500	0.500	0.500
0.375	0.500	0.500
0.500	0.125	0.500
0.500	0.500	0.625
0.250	0.750	0.250
0.125	0.750	0.250
0.250	0.875	0.250
0.250	0.750	0.875
0.750	0.250	0.750
0.625	0.250	0.750
0.750	0.375	0.750
0.750	0.250	0.375

Figure 18.16: Construction of the first three points of the radial design based on the Sobol' sequence (design shown in Figure 18.17). Values from the **B** matrix are shown in grey.

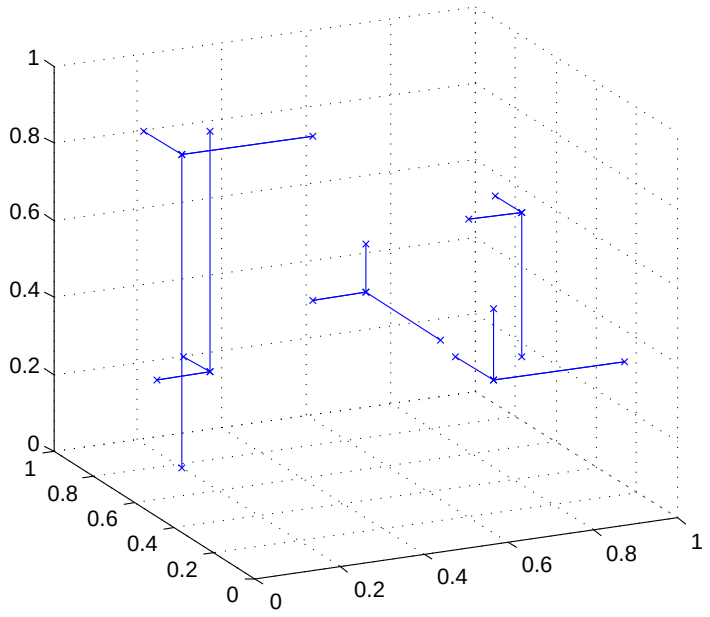


Figure 18.17: A radial screening design in three dimensions, with $n = 5$.

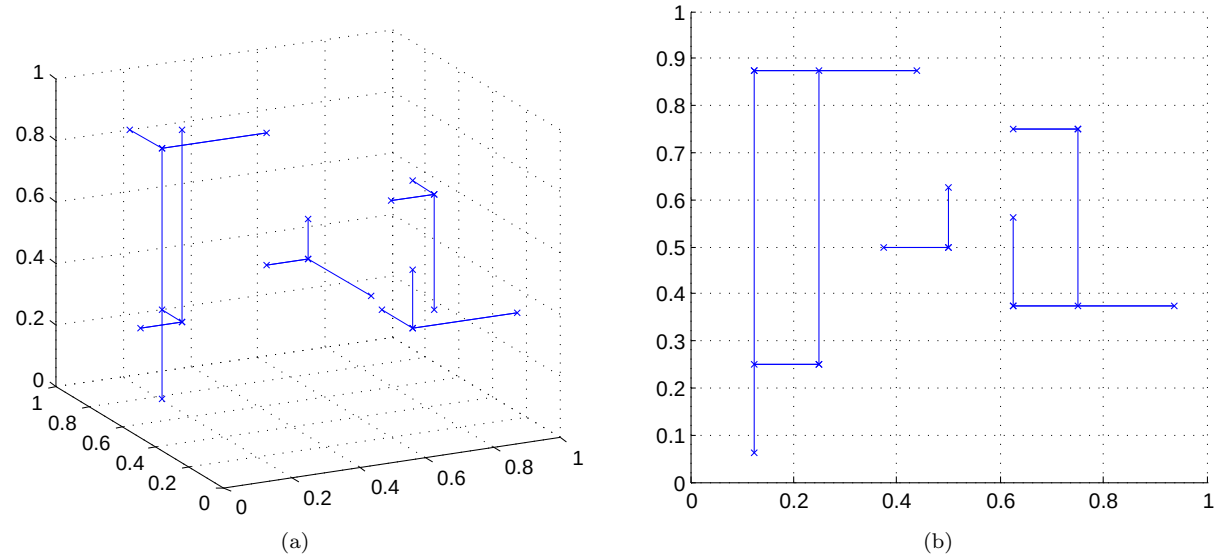


Figure 18.18: Collapse of radial design when projected onto a subspace: (a) a radial design in three dimensions; (b) the same design projected onto the subspace of x_1 and x_3.

there is no reason not to consider higher values of δ or to program the algorithm to skip any replicated values. In any case, this "row shift" means that for a design of n radial configurations, one needs a Sobol' sequence of $n+\delta$ points in $2k$ dimensions. This still has a computational cost of $n(k+1)$ runs because the first δ rows of $\mathbf{B}$ and the last δ rows of $\mathbf{A}$ are discarded. Indeed, this "row-shift" strategy could also be applied to the estimators for the variance-based sensitivity indices discussed in Section 18.3, but is not generally deemed to be an issue since when estimating S_{Ti} one is in the domain of Case 1A (a cheap computer model that can be run thousands of times) where an extra few model runs make little difference, whereas in the screening setting (Case 1C) one has to conserve model runs as much as possible.

The elementary effects method is often used in the factor fixing setting (identifying input variables that have little or no effect on the model output), with the possible intention of estimating sensitivity indices on the remaining set of "important" variables via Monte Carlo methods. An advantage therefore of using the radial design, noted in Campolongo et al. (2011), is that since it is effectively the same as the Monte Carlo design, the points can be re-used as the basis of a Monte Carlo design for estimating sensitivity indices. Alternatively (and perhaps more realistically), one might want to fit an emulator to the reduced set of input

variables, since the computational cost of Monte Carlo methods is likely still prohibitive even after screening. In that case, the OAT-type designs discussed in this section (both winding stairs and radial) are wasteful since if they are projected onto the reduced-dimensionality subspace of the set of screened inputs, for every input that is discarded, n model runs are lost. To see why this is, consider the example in Figure 18.18. A radial design is built in three dimensions, the elementary effects method is applied, and x_2 is found to be unimportant. It is desired to examine the remaining inputs (x_1 and x_3) in more detail using an emulator. When the design is projected into the subspace of x_1 and x_3, the points that differ only by a step in the x_2 direction, a total of n points, one for each "star", are now duplicate points. A recent approach that overcomes this problem uses a set of simplex designs to estimate screening measures (Pujol, 2009), which have the property that all design points are retained after screening out unimportant variables.

18.5.4 Example

To show a simple example, the design shown in Figure 18.17 is used to estimate the μ_i, μ_i^* and σ_i^2 measures on a test function which is defined as,

$$y = 3x_1^2 + 2x_1x_2 - 4\pi \sin(x_3); \tag{18.29}$$

Figure 18.19 shows scatter plots of the function values against each input, using random uniform sampling from $[0, 1]^3$ to give a visual idea of the behaviour of the function. The screening design is run with $n = 5$ and $k = 3$ (exactly as in Figure 18.17); a cost of 20 model runs. Figure 18.20 shows the results of the screening analysis. Input x_2 is clearly the least influential by any measure, and is relatively linear, having a small value of $\hat{\sigma}_i^2$. Similarly, x_1 is quite linear (having also a small value of $\hat{\sigma}_i^2$), but is judged to be more important by its $\hat{\mu}_i$ and $\hat{\mu}_i^*$ measures. In both these cases $\hat{\mu}_i = \hat{\mu}_i^*$, which indicates monotonicity. For x_3 one can observe a much higher $\hat{\sigma}_i^2$ value, indicating a strong nonlinearity. Finally, the fact that $\hat{\mu}_3 \neq \hat{\mu}_3^*$ indicates that the model is non-monotonic with respect to x_3. These results are reflected by the scatter plots in Figure 18.19.

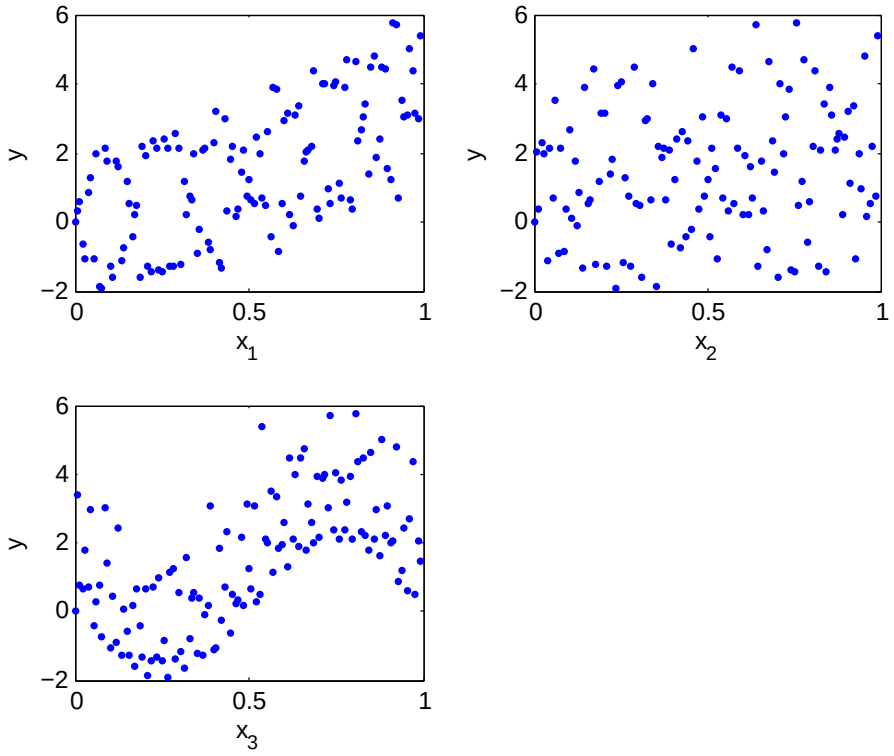


Figure 18.19: Scatter plots for test function (18.29). Note that these are generated using random uniform sampling from $[0,1]^3$ — these are not the same design points used in the screening design.

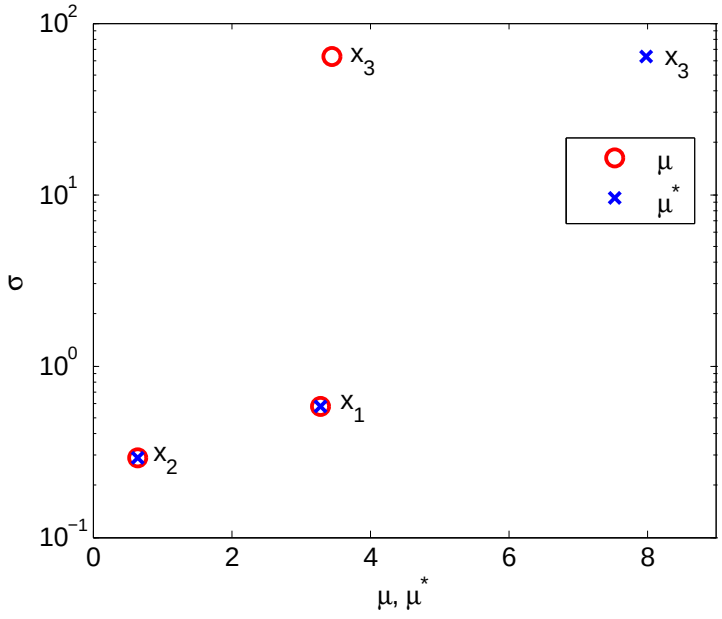


Figure 18.20: Estimates of screening measures μ_i, μ_i^* and σ_i^2 plotted for test function (18.29).

Note that the screening measures do not have an interpretation with regard to the variance of the output (as compared to the S and S_T measures from Section 18.2), but they allow the user to sort between influential and uninfluential variables. Consider also that the example given is trivial, since screening is generally for use with high-dimensional problems, but even with 20 runs the order and to some extent the magnitude of importance of each variable can be distinguished with these relatively simple measures, as well as information regarding the linearity and monotonicity.

18.5.5 Discussion

It is interesting to consider briefly the similarity of the estimator of μ^* in (18.28) and the estimator for S_{Ti} given in (18.12). Let $\boldsymbol{x}_j$ be the jth point of a random or quasi-random sequence in the input space, and let $\boldsymbol{x}_j^{(i')}$ be a point that differs from $\boldsymbol{x}_j$ only by the ith variable. Let $E_{\boldsymbol{x}_{\sim i}}[V_{x_i}(y \mid \boldsymbol{x}_{\sim i})]$, the numerator of S_{Ti}, also be denoted as V_{Ti} as before. Then the estimators for μ_i^* and V_{Ti} can be stated in similar terms as follows,

$$\begin{aligned} \hat{\mu}_i^* &= \frac{1}{n}\sum_{j=1}^{n} \frac{\left| f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j) \right|}{|x_{ji}^{(i')} - x_{ji}|} \\ \hat{V}_{Ti} &= \frac{1}{2n}\sum_{j=1}^{n} \left| f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j) \right|^2 . \end{aligned} \tag{18.30}$$

where the estimator for μ_i^* has been written slightly differently compared to (18.25) and (18.28) since in the radial design all k elementary effects in each "star" use the same centre point, $\boldsymbol{x}_j$. Notice that both measures rely on averages of model outputs from multiple OAT designs conducted at different points in the input space. The only differences (up to a proportional constant) are that $\hat{\mu}_i^*$ uses the absolute value of $(f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j))$, whereas $\hat{S}_{Ti}$ uses the square, and $\hat{\mu}_i^*$ also incorporates the information about the distance between $\boldsymbol{x}_j^{(i')}$ and $\boldsymbol{x}_j$, which is instead discarded by $\hat{S}_{Ti}$.

Another very similar related measure of sensitivity that has been recently the subject of interest, under the heading of "derivative-based global sensitivity measures" (DGSM) is the

integral of squared derivatives, i.e. a measure $\nu_i = \int_{\mathcal{X}} (\partial f / \partial x_i)^2 d\boldsymbol{x}$ (Kucherenko et al., 2009). Using the notation just defined, its estimator is stated as

$$\hat{\nu}_i = \frac{1}{n} \sum_{j=1}^{n} \frac{\left| f(\boldsymbol{x}_j^{(i')}) - f(\boldsymbol{x}_j) \right|^2}{|x_{ji}^{(i')} - x_{ji}|}, \tag{18.31}$$

with the difference that the $\boldsymbol{x}_j^{(i')}$ and $\boldsymbol{x}_j$ points are defined to be very close to each other to provide as close an approximation as possible to the partial derivative at $\boldsymbol{x}_j$. Some further observations on the similarities of these three measures, and further information on DGSM are found in Sobol and Kucherenko (2009).

Connections can also be found between the radial design (Figure 18.17) and the Monte Carlo design for estimating S_i and S_{Ti} (Figure 18.7), and their respective design matrices (Figures 18.16 and 18.6). One can see that the two designs are conceptually identical, with the only difference being that the screening design here uses a shift of the $\mathbf{B}$ matrix to avoid replacing a coordinate with an identical value. Recalling the discussion on the similarity of the estimators of μ_i^* and S_{Ti} in the previous section, it is evident that the elementary effects method and the Monte Carlo estimation of sensitivity indices have much in common.

Finally, in some cases the number of input variables may be so large, or the cost of running the model so high that even the designs here may be too expensive. In that case, one option available to the analyst is to use grouped designs, where inputs are grouped and designs are built with respect to each group, rather than to each input. In that case one would have a measure of sensitivity for each group, but be ignorant of the relative sensitivity of inputs inside each group. This concept is discussed more in Watson (1961); Morris (1991); Campolongo et al. (2007); Moon et al. (2012).

18.6 Emulators

In Case 1B (the model is expensive to run but the dimensionality is not too high) and Case 2 (points are "given" and the analyst cannot specify new points) – refer back to Section

18.1.2, a general method is presented here which adopts a data modelling approach. The concept is to fit an *emulator* (a relatively simple mathematical function, also known as a *metamodel*) to the data, which behaves in the same way as the model itself. The emulator can then be used to estimate the model output at any point in the input space, allowing analytical or Monte Carlo estimation of the S_i and S_{Ti} (see Section 18.3) at a considerably lower computation cost. An alternative approach is to project the data onto a single axis x_i (i.e. create k one-dimensional scatter plots) and attempt to infer the main effect $E(y|x_i)$ using a smoothing regression approach, for example kernel regression. This latter approach is discussed in Section 18.7.

The central idea of emulation is to find some relatively simple function η (the emulator) such that it closely approximates the output of the model f at any point in the input space, i.e. $\eta(\boldsymbol{x}) \approx f(\boldsymbol{x}),\ \forall\, \boldsymbol{x} \in \mathcal{X}$. If η is considerably cheaper to evaluate at a given $\boldsymbol{x}$ than the original model, but produces sufficiently similar results for any point in the input space, then it can be used to generate a very large number of estimated model output values, for example at the points specified by a Monte Carlo design for estimating S_i and S_{Ti} (see Section 18.3). Even better, if η is analytically tractable it can be used to calculate sensitivity indices analytically, because (18.2) and (18.6) can be expressed as integrals which can be solved if $\eta(\boldsymbol{x})$ is sufficiently simple. This then bypasses Monte Carlo methods altogether and the associated approximation error of numerical integration.

The four steps associated with building an emulator are,

1. Select a type of emulator, η, that is appropriate for emulating the model $f(\boldsymbol{x})$. Options could be as simple as a linear regression, to more complex methods such as Gaussian processes, neural networks or smoothing splines. A brief description of some emulators follows in this section, but for a more detailed treatment one should refer to one of the many books on the subject, such as Bishop (2006).

2. Sample (run) the model f at appropriate points in the input space to provide "training data" for the emulator. This will be a set of n points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ and n corresponding model outputs $\{y_1, y_2, \ldots, y_n\}$ where the location of the inputs is usually chosen to be a space-filling design or "optimal design" (see Section 18.6.3). Emulators can how-

ever be fit to any sample of data (thus can be used for Case 2 problems) although in areas of the input space with very few training data they will be less accurate.

3. Use the training data to estimate any parameters or hyperparameters associated with the model (this is called "training" the emulator).

4. Check the accuracy of the estimator using methods such as cross-validation (an approach which involves training the emulator on a subset of the training data, and then comparing the predictions of the emulator with the known model runs in the remaining set of data).

A large number of different emulators are available – see for example Bishop (2006)); a comparison of some of these methods in the context of sensitivity analysis can be found in Storlie and Helton (2008). Once an appropriate emulation method is selected, the accurate estimation of parameters can be achieved provided that a sufficiently-large sample of training data (the given points) is available. How large this sample needs to be is dependent on the type of emulator, the complexity of the model f, and increases dramatically with the number of input variables (the so-called curse of dimensionality). For this reason, emulator-based approaches are best suited to situations with few input variables — perhaps fewer than thirty, depending on the emulator — which demand fewer model evaluations for training data. Higher-dimensionality problems can sometimes be brought into the reach of emulators by a precursory screening analysis (see Section 18.5) to reduce the number of variables (refer back to Figure 18.2). While there are many emulators available in the literature, the following sections give some brief information on only a small subset to give a flavour of the field.

18.6.1 High-dimensional Model Representation

A significant proportion of emulation methods rely on a technique known as High-Dimensional Model Representation (HDMR), which seeks to approximate the model by performing a functional decomposition of f into orthogonal terms, then truncating the series (Rabitz and Aliş, 1999; Li, Wang, Rabitz, Wang and Jaffé, 2002). This has already been given in (18.4), and

is restated as:

$$\begin{aligned} f(\boldsymbol{x}) &= f_0 + \sum_i f_i(x_i) + \sum_i \sum_{j>i} f_{i,j}(x_i, x_j) + \ldots + f_{1,2,\ldots,k}(x_1, x_2, \ldots, x_k) \\ &\approx f_0 + \sum_i f_i(x_i) + \sum_i \sum_{j>i} f_{i,j}(x_i, x_j) \end{aligned} \tag{18.32}$$

where the series is truncated after the second-order terms, based on the (empirical) observation that the effect of third-order and higher interactions on the model output is negligible in many models. The task then remains to find suitable orthogonal functions to approximate the $f_i(x_i)$ and $f_{i,j}(\boldsymbol{x}_{i,j})$ (some approaches are discussed in Li, Wang and Rabitz (2002)). Orthogonality is satisfied for any two components $f_u(\boldsymbol{x}_u)$, $f_v(\boldsymbol{x}_v)$, of the functional decomposition in (18.32) when $\int_{\mathcal{X}} f_u(\boldsymbol{x}_u) f_v(\boldsymbol{x}_v) d\boldsymbol{x} = 0$, where $u, v \subseteq \{1, 2, \ldots, k\}$ and e.g. $\boldsymbol{x}_u = \{x_i\}_{i \in u}$. The advantage of HDMR is that it alleviates to some extent the problem of dimensionality, since a model with many inputs can be approximated by a sum of one and two-dimensional terms which are relatively easy to fit.

One method for approximating the terms in the functional decomposition (18.32) which is relatively simple is to use a series of orthogonal polynomial functions (Li, Wang, Rabitz, Wang and Jaffé, 2002; Sudret, 2008; Draper and Smith, 1981). These take the form,

$$\begin{aligned} f_i(x_i) &= \sum_{r=1}^{\infty} \alpha_r^{(i)} \phi_r(x_i) \\ f_{i,j}(x_i, x_j) &= \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \beta_{p,q}^{(i,j)} \phi_{p,q}(x_i, x_j) \end{aligned} \tag{18.33}$$

where the ϕ are terms from a suitable series of orthogonal polynomials, and the α and β are their corresponding coefficients. Of the many series of orthogonal polynomials, it is typical to use either the Legendre or the Hermite types (Bayin, 2006). Clearly, it is necessary to truncate the infinite series of each of the equalities in (18.33) to a certain order M, which is usually done by discarding terms after the third order based on a heuristic assumption that higher orders are negligible (Li, Wang, Rabitz, Wang and Jaffé, 2002), or in some cases by sequentially adding terms and re-estimating coefficients until estimated sensitivity indices

appear to converge (Zuniga et al., 2013). This gives the approximated functions as follows,

$$\begin{aligned} f_i(x_i) &\approx \sum_{r=1}^{M} \hat{\alpha}_r^{(i)} \phi_r(x_i) \\ f_{i,j}(x_i, x_j) &\approx \sum_{p=1}^{M} \sum_{q=1}^{M} \hat{\beta}_{p,q}^{(i,j)} \phi_{p,q}(x_i, x_j) \end{aligned} \tag{18.34}$$

where the estimates of the coefficients α and β are obtained by minimising the squared difference between the terms in (18.34) and their counterparts in the HDMR decomposition one at a time. This minimisation problem can be expressed as a series of integrals which can be solved by Monte Carlo integration (Li, Wang and Rabitz, 2002). The HDMR emulator can then be built by assembling the terms from (18.34) into the truncated representation in (18.32). However, due to the simplicity of the orthogonal polynomials, it is possible to derive analytical expressions for S_i and $S_{i,j}$, which are as follows,

$$\hat{S}_i = \frac{\sum_{r=1}^{\infty} (\hat{\alpha}_r^{(i)})^2}{\hat{V}(y)} \tag{18.35}$$

$$\hat{S}_{i,j} = \frac{\sum_{p=1}^{\infty} \sum_{q=1}^{\infty} (\hat{\beta}_{pq}^{(ij)})^2}{\hat{V}(y)} \tag{18.36}$$

where the summations are again truncated to the same orders used in (18.34). S_{Ti} may be approximated from the sum of S_i and its second order interactions, based on the same assumption used in (18.33) that interactions above the second order do not contribute significantly to the model output (and therefore by extension to $V(y)$). The variance term $\hat{V}(y)$ can be calculated either from an analytical expression similar to those for the sensitivity indices, or from the original sample data. The drawbacks to the use of orthogonal polynomials are that the HDMR representation must be truncated, then a further truncation is necessary of the polynomial series. Higher-order polynomials can be used, but this requires the estimation of more coefficients, each of which has an associated Monte Carlo error, and tends to increase the error rather than diminish it (Li, Wang and Rabitz, 2002).

A somewhat related method to HDMR with orthogonal polynomials is called *polynomial chaos expansion*, which is an approach which involves representing the random variable of

the model output, y, as a function of other random variables (the model inputs $\boldsymbol{x}$) via a polynomial expansion in orthogonal polynomials. The details of this method will not be explored here, but the reader is referred to Sudret (2008) for details on the methodology in the context of sensitivity analysis.

Another more complex approach for approximating functions that has been used with considerable success is the use of smoothing splines (Ratto and Pagano, 2010). Smoothing splines are most commonly used on univariate data but can be extended to the multivariate case by the use of HDMR decompositions. In the following short introduction the univariate case is therefore described for simplicity. A smoothing spline model assumes that the underlying function to be emulated is continuous, has a continuous first derivative, and further that the second derivative is square-integrable. The smoothing spline estimate arises from considering the function g that minimises the following,

$$\frac{1}{n}\sum_{j=1}^{n}\left\{y_j - g(x_j)\right\}^2 + \lambda \int_0^1 \left\{g''(x)\right\}^2 dx \tag{18.37}$$

The first term in this "tradeoff" expression is simply the sum of squared errors between the training data and the emulator g — if the function were to pass through every data point (exact interpolation), this term would be zero. The second term expresses the integral of the square of the second derivative of g, which is a global measure of roughness, and λ is a "tuning parameter" that controls the weighting between the two terms. Overall therefore, the expression summarizes the tradeoff between interpolation and model simplicity. The solution to this minimisation problem can be shown to be a natural cubic spline, with knots (joins between the local cubic functions) at each of the data points. Natural cubic splines are simply local cubic polynomial functions between each data point and the next, with the constraints that the global function is continuous and the first derivative is continuous at knots. Since multivariate spline emulators rely on HDMR decompositions however, estimates of total effect indices S_{Ti} are difficult since the HDMR series is truncated (usually neglecting third-order and higher interactions), so higher-order interactions that may contribute to S_{Ti} are not accounted for. A common approximation is therefore to assume that $S_{Ti} \approx S_i + \sum_{j=1}^{k} S_{i,j}$.

The basis functions of splines as described above are not mutually orthogonal and there-

fore cannot be used in their standard form as the components of an HDMR emulator (Li, Wang and Rabitz, 2002). However, a class of spline methods known as "smoothing spline ANOVA models" allows the construction of mutually orthogonal spline basis functions via the reproducing kernel Hilbert space approach, which is discussed extensively in Gu (2002).

An extension of multivariate splines, known as Adaptive Component Selection and Shrinkage Operator (ACOSSO), uses a modified version of (18.37) that uses norms rather than square-norms, and also includes the integral of the first derivative of g (Lin and Zhang, 2006). The result is that terms in the HDMR decomposition that contribute very little to the function are eliminated, resulting in a simpler and more computationally-efficient emulator. This approach has also been combined with state-dependent parameter regression, as described in Ratto et al. (2007). Matlab scripts for performing sensitivity analysis with this approach can be found in European Commission (2012).

18.6.2 Gaussian Processes

Another emulator that is widely used in sensitivity and uncertainty analysis is a *Gaussian process* (GP), otherwise known as *kriging*. GPs are widely used in the machine learning community as a sophisticated form of nonlinear regression and classification (Rasmussen and Williams, 2006). In short, a GP is a *distribution over functions*, i.e. the random variable of the distribution is a function rather than a single number or fixed-length vector. Instead of returning only a point estimate $\hat{y}$ for any given input point $\boldsymbol{x}$ (as in a standard regression), the GP returns a specification for a Gaussian probability distribution (a mean and a variance). Figure 18.21 shows an example of a simple one-dimensional GP fitted to a few points from a sine wave. Notice that at any value of x the output y is estimated by the GP as a predicted mean value (which forms a curve over the range of x) and a variance, here plotted as plus/minus two standard deviations of the predictive distribution.

GPs are fitted to training data following a Bayesian procedure. A prior distribution over functions is specified, which is then conditioned on the training points to give a posterior distribution over functions. Figure 18.21 shows an example of this process applied to data from

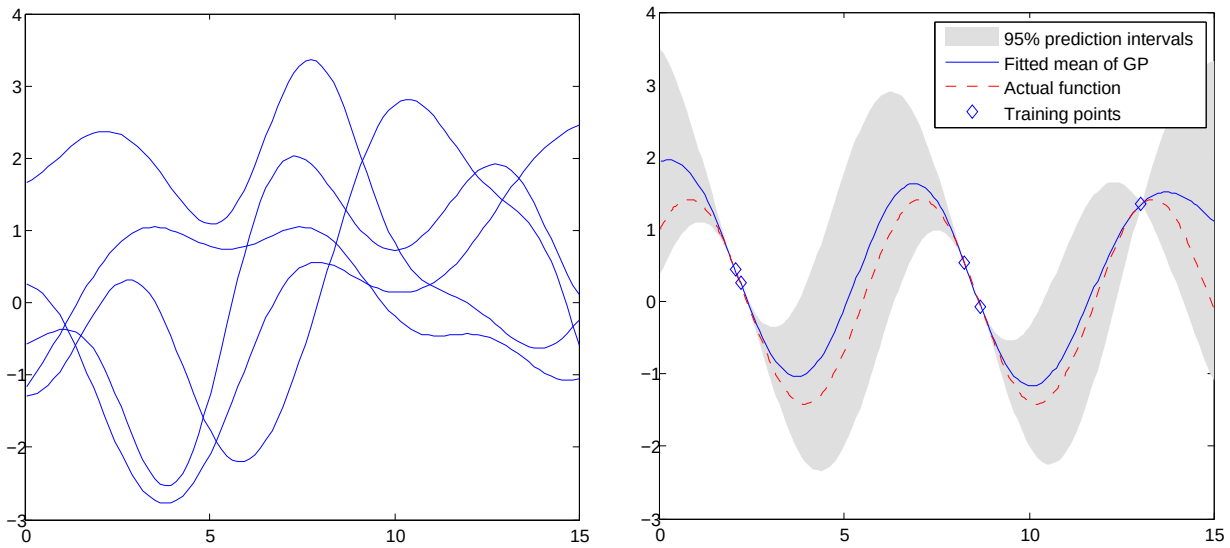


Figure 18.21: Gaussian process examples: (a) samples from the prior distribution over functions; (b) the GP conditioned on randomly-generated points from a sine wave. The dotted line is the underlying sine function, the solid line is the fitted mean of the GP and the grey region represents 95% prediction intervals.

a noisy sine wave — Figure 18.21(a) shows samples from the prior distribution over functions, and Figure 18.21(b) shows the posterior distribution over functions after conditioning on the training data.

The specification for the GP includes a number of hyperparameters. These may be estimated, for example, by plug-in estimators, e.g. by using maximum likelihood estimation – this effectively treats the hyperparameters as known. Alternatively, one can assign prior distributions to hyperparameters, then integrate them out (marginalise them) either analytically or using Markov Chain Monte Carlo methods – see Bishop (2006) and Chapter 16 for more information on Bayesian inference). In this "full Bayesian" framework the confidence intervals of the GP also include the uncertainty in the estimation of the hyperparameters.

GPs have the advantage that they do not invoke the HDMR assumption (of neglecting higher-order interactions, see (18.32)), and as result can be used to estimate sensitivity indices of all orders, including the S_{Ti}. A further useful property of the GP is that, given certain assumptions, estimated sensitivity indices can be calculated analytically (Oakley and

O'Hagan, 2004; Becker et al., 2012). Another extremely useful property is that, since the GP is a probabilistic emulator, sensitivity indices can be stated with confidence intervals which account for the uncertainty in the fit of the emulator (and also the uncertainty in the hyperparameter estimation, when Bayesian inference is used). However, perhaps the main weakness of GPs is that the cost of training scales poorly with n, the number of training data, since it involves the inversion of a $n \times n$ covariance matrix at a cost of the order of n^3. This limitation impacts on the dimensionality of the problems to which GPs can be applied, since more training data is required as the dimensionality increases. Added to the fact that the estimation of hyperparameters becomes increasingly expensive as the dimensionality increases, GPs tend to encounter problems emulating models with $k > 30$ inputs or so, depending on computational resources (this is however the case with all emulators, to some extent). A number of techniques are being developed to alleviate this problem – see Rasmussen and Williams (2006) for some examples. Some fairly recent additions in the field of GPs with respect to sensitivity analysis include a method of automatically screening out unimportant variables using the correlation parameter in the covariance function (see Linkletter et al. (2006), based on Welch et al. (1992)), a method based on screening variables in groups using GPs (Moon et al., 2012), and the use of multiple GPs divided by decision trees to allow for discontinuous responses (Gramacy and Taddy, 2010; Becker et al., 2013) (available as an R package). A very good general online resource on many aspects of GPs, emulation and sampling can be found at *Managing Uncertainties in Complex Models (MUCM) Toolbox* (2013).

As an example of how an emulator may reduce the number of model runs required for sensitivity analysis, consider again the simple polynomial from (18.15). Using 128 points of the Sobol' sequence over the unit cube, a Gaussian process was trained (i.e. the hyperparameters of the mean and covariance functions were estimated using the training data), and estimated sensitivity indices inferred analytically from the resulting posterior distribution. Table 18.2 shows the results. The GP is achieving accuracies of three or more decimal places on only 128 points – recall that the Monte Carlo estimator, for a similar level of accuracy, requires several thousands of runs per variable, therefore the GP is at least an order of magnitude more efficient. However, the GP and other emulators are only as good as their fit to the data: here the polynomial function is a smooth, "well-behaved" function, which is an

Variable	$\hat{S}_i$ (GP)	S_i (analytic)	$\hat{S}_{Ti}$ (GP)	S_{Ti} (analytic)
x_1	0.7566	0.7568	0.7715	0.7720
x_2	0.0456	0.0456	0.0605	0.0608
x_3	0.1829	0.1824	0.1830	0.1824

Table 18.2: Comparison of S_i and S_{Ti} estimates from a Gaussian process regression against analytical values

easy data modelling problem. For data that are heteroscedastic, discontinuous, or of varying smoothness, the emulators are likely to be much less reliable. Additionally, they scale poorly with dimensionality. However, in the cases where the model gives a relatively smooth output (which appears to be the majority of physical models) emulators can offer a powerful solution to the problem of analysing uncertainty and sensitivity for computationally-expensive models. For more detailed information on GPs please refer to Rasmussen and Williams (2006) and Chapter 17 of this handbook.

Overall, there is no "best" emulator available. The approach will depend on computational resources, sample size and model dimension, amongst other things. Both (Bishop, 2006) and (Storlie and Helton, 2008) are recommended as background reading. Furthermore, it is essential to test the fit of any emulator by methods such as cross validation.

18.6.3 Custom Sampling for Emulators

Although the points were considered as "given" in the discussion above, it can happen that the analyst has the possibility to design their own set of training data for fitting an emulator. This can be the case when, for example, the analyst only has a small number of input variables, but a very computationally-expensive model (Case 1B in Section 18.1.2). In this scenario it makes sense to go directly to an emulator approach, since pure Monte Carlo would be too expensive and screening too inaccurate.

Experimental designs for fitting emulators can be divided into two categories - space-filling designs, and model-based designs. In the former, the design is constructed to fill the sample space as evenly as possible, which is to say that points should be as far apart from each other as possible. The reasoning for this is that first, it is required to capture the behavior

of the model over the whole input space with as few points as possible. Second, assuming that the output of the model is deterministic and smooth with respect to its inputs, little information can be gained by having points close to each other (since the outputs will be very similar). For this reason, purely random sampling is not an efficient design.

For a general-purpose design for fitting an emulator, a space-filling design such as the Sobol' sequence discussed in Section 18.3 is a good choice. Sobol' designs have a low-discrepancy property that avoids "clumping" of points, and allow the sequential addition of new points. For other space-filling designs the reader is referred to Chapter 19 or Niederreiter (1992).

An even more sophisticated approach is to use "optimal design". In this approach, the design is constructed so as to optimise some emulator-dependent criterion of interest, thus tailoring the design to the requirements of the emulator. For example, a popular criterion, called "D-optimality", is to select design points which minimise the variance of the estimators of the emulator parameters. Another way to select design points is to minimise the maximum variance of the emulator prediction at any given input point (G-optimality). Note however that for emulators based on linear models or stationary Gaussian processes, these designs are *not* dependent on the output of the model, only on the form of the emulator and the values of its parameters. See Chapter 2, Section 2.3 for more information on D-optimality.

A further approach to building designs for emulators is to construct them sequentially. One starts with an initial design with a few input points (e.g. a space-filling or optimal design) and runs the model at these points. The emulator is fitted to this training data (i.e. its hyperparameters are estimated), then the next input point is selected as the point which optimises some criterion of interest. An example, for a GP, would be to choose the point which returns the highest predictive variance. This new point is then run through the model and used to re-estimate the hyperparameters. The procedure is repeated, choosing each time the point with the highest predictive variance, until a design with a sufficient number of points is created. The advantage of the sequential approach (known as adaptive design) is twofold – first, the output values will influence the optimum placement of new points, so knowledge of previous points will produce a more effective design than one that

is constructed in one go. Second, by proceeding in small steps, one can generate exactly the required number of points to reach some level of accuracy of interest, perhaps measured by cross validation. More information on adaptive sampling can be found in Gramacy and Lee (2009).

The theory of model-based designs is a large field of research that is beyond the remit of this chapter, therefore the reader is referred to Atkinson et al. (2007) for a good general resource. There is also a strong interest in Bayesian approaches to optimal design; a review can be found in Chaloner and Verdinelli (1995).

18.7 Scatter Plot Smoothing

A useful approach for estimating first order sensitivity indices with "given" data is based on one-dimensional nonlinear smoothing regression. From a computational point of view, this method is less vulnerable to the curse of dimensionality, although it cannot be used to calculate the total order sensitivity indices S_{Ti}.

A first visual indication of the effects of input variables can be gained by making k plots of x_i against y (see Figure 18.8). If the data shows any kind of trend (or shape) with respect to x_i, this indicates that x_i has some effect on the output. Indeed, the effect of x_i on the output is described by the curve $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$ — in other words, the expected value of the model output if we were to fix x_i at some value. Over the range of x_i, $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$ is equivalent to a moving weighted average of the points in the x_i against y plot. As long as the $\boldsymbol{x}_j, j = 1, 2, ..., n$ have been randomly drawn from their specified distribution $p(\boldsymbol{x})$, S_i can be estimated by taking the variance of the y values of this curve (since $S_i = V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y|x_i)]$).

To estimate such a moving average, it is a matter of using any of a number of smoothing regression approaches. Kernel regression has been already used for sensitivity analysis (Paruolo et al., 2013). As with any smoothing regression, the data are modelled as,

$$y = m(x_i) + \epsilon \tag{18.38}$$

where $m(x_i)$ is the smoothed curve (ideally equivalent to $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$), and ϵ is an independent error term with mean 0 and a variance that may be dependent on x_i. Although most nonlinear regression approaches assume a fixed variance over x_i, the smoothing curves that result when this assumption does not hold are still valid, albeit less efficient. However, nonlinear regression that accounts for heteroscedasticity is still a field of active research, therefore this chapter does not venture into this territory and readers are referred to a discussion in Ruppert et al. (2003) as a starting point for further information.

In the kernel regression setting, $m(x_i)$ is typically chosen to be either a *local mean* or *local linear* kernel. The local mean estimator first proposed by Nadaraya (1964); Watson (1964), for a single input variable x_i, is expressed as,

$$\hat{m}(x_i) = \frac{\sum_{j=1}^{n} w(x_{ji} - x_i; h) y_j}{\sum_{j=1}^{n} w(x_{ji} - x_i; h)} \tag{18.39}$$

where w is a weighting function and h is a tuning parameter. The weighting function typically gives the strongest weight to points close to x_i, which reflects the belief that the closer two points are to each other in x_i, the more likely they are to have similar values in y. A commonly-used function that fulfils this requirement is a Gaussian density function with standard deviation h. The local linear estimator is expressed in a similar fashion – see (Bowman and Azzalini, 1997) for details – and is generally regarded as preferable to the local mean, due to its improved properties near the edges of the data cloud. In all cases, the smoothing parameter h can be optimised by cross-validation. Following the simple polynomial example from (18.15), Figure 18.22 shows an illustration of local linear kernel regression applied to scatter plots of y against each x_i. The resulting estimates of sensitivity are given in Table 18.3.

In order to estimate sensitivity from the fitted kernel regression, the estimated variance of the conditional expectation can be calculated using a standard variance identity (the domain of the expected value is explicitly stated as a subscript here for clarity),

$$V_{x_i}\{E_{\boldsymbol{x}_{\sim i}}(y|x_i)\} = E_{x_i}\{E_{\boldsymbol{x}_{\sim i}}(y|x_i)^2\} - E_{x_i}\{E_{\boldsymbol{x}_{\sim i}}(y|x_i)\}^2. \tag{18.40}$$

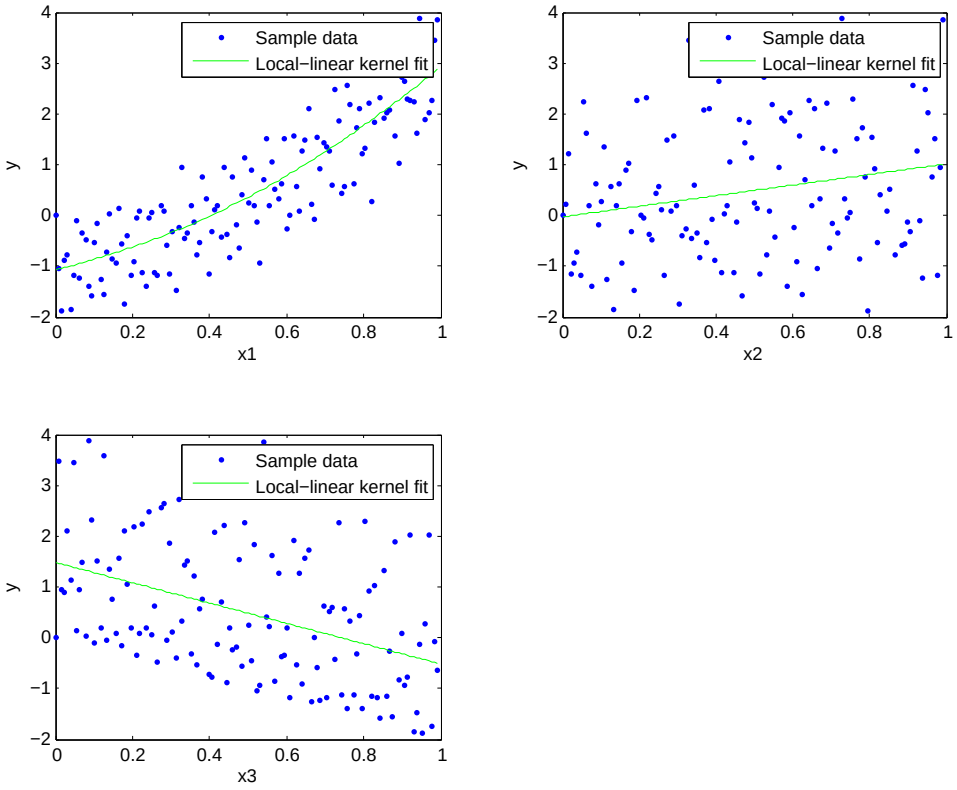


Figure 18.22: Local-linear kernel regression applied to the polynomial function.

Variable	$\hat{S}_i$ (kernel)	S_i (analytic)
x_1	0.7345	0.7568
x_2	0.0494	0.0456
x_3	0.1821	0.1824

Table 18.3: Comparison of S_i estimates from local-linear kernel regression of polynomial function against analytical values

Then denoting $V_i = V_{x_i}[E_{\boldsymbol{x}_{\sim i}}(y \mid x_i)]$ as before, V_i is estimated by the smoothed curve $\hat{m}(x_i)$ to give,

$$\hat{V}_i = E_{x_i}\{\hat{m}(x_i)^2\} - E_{x_i}\{\hat{m}(x_i)\}^2. \tag{18.41}$$

To evaluate (18.41) one simply calculates a large number of values of $\hat{m}(x_i)$ at different locations and uses the standard estimator for sample variance for each term.

Note here that since the expectation of a random variable A is defined as $\int A\, p(A) dA$, the expected values in (18.40) should be evaluated with respect to $p(x_i)$, the probability distribution of x_i. The simplest way of doing this is to make kernel predictions at the same x_i values as the training data. A more sophisticated approach would be to estimate $p(x_i)$ with kernel density estimation or a similar technique. In both cases, the practitioner should ensure that the given data has been sampled with respect to the correct underlying distributions – this may not necessarily be the case if the points have come from an optimization process, for example.

While the examples here have focused on kernel smoothing, this is by no means the only viable approach to estimating $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$. The problem is essentially an emulation/regression problem in one dimension, which can be tackled by any number of methods such as smoothing splines – see e.g. Ruppert et al. (2003) or earlier in Section 18.6, or Gaussian processes (Rasmussen and Williams, 2006). Even basic linear regression will provide a good estimate if the data is sufficiently linear. Good references on parametric and nonparametric regression can be found in Bishop (2006) and Ruppert et al. (2003).

The estimation of S_i using smoothing techniques is very closely related to the well-known "coefficient of determination", R^2. The "standard" R^2 is the square of the Pearson correlation coefficient, and measures linear correlation between two variables. The nonlinear R^2

measure, which measures correlation when nonlinear regression is used, can be stated in general terms as $R^2_{\text{nonlin}} = SS_{\text{reg}}/SS_{\text{tot}}$, where $SS_{\text{reg}} = \sum_j (\hat{m}(x_{i,j}) - \bar{y})^2$ and $SS_{\text{tot}} = \sum_j (y_j - \bar{y})^2$. It can be seen therefore as the ratio of the variance explained by the regression of y on x_i and the total variance of y. Therefore if the function $m(x_i)$ is equal to $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$, the nonlinear R^2 measure is exactly equivalent to S_i.

Finally, it should be pointed out that while the idea of reducing a multidimensional problem to a series of one-dimensional problems is very appealing from a computational point of view, it is not a "silver bullet" solution. The estimation is dependent on a good approximation of $E_{\boldsymbol{x}_{\sim i}}(y|x_i)$, which can be difficult to obtain depending on the data and smoothing method used. Moreover, as the dimensionality of the problem increases, trends in scatterplots can be increasingly biased and/or less precise due to variation in other dimensions.

18.8 Conclusions and Discussion

In this chapter, a number of "best practice" methods have been outlined that address many of the various situations that can confront an analyst. It is not claimed nor intended by the authors that an exhaustive review of all methods has been addressed here, but the reader should have found here enough material to apply or adapt to a practical case, with references to other material to direct the reader to more in-depth descriptions.

It is however useful to summarise some areas not covered by the chapter. What did this chapter leave out? An incomplete list is as follows:

- Gradient-based sensitivity measures (mentioned in Section 18.5.1) use a measure similar to μ_i^* given in (18.28), which is the integral of the squared partial derivative as a measure of sensitivity, which has been shown to have a relationship with S_{Ti} – see Sobol and Kucherenko (2009); Kucherenko et al. (2009).
- Polynomial Chaos Expansions, briefly mentioned in Section 18.6, constitute a significant branch of emulator-based sensitivity analysis, but were omitted from the chapter due to space limitations and with the intention of focusing more on the sampling aspect of

sensitivity analysis. More information can be found in Sudret (2008).

- Moment independent methods form a class of sensitivity analysis approaches that examine how fixing an input variable modifies the entire empirical probability distribution function of y – see Borgonovo (2007). The rationale behind these methods is that variance is but one of several possible moments that can indicate sensitivity. Such methods focus on measures of "distance" between the unconditional distribution of y and the conditional distribution of $y|x_i$.

- A sensitivity analysis approach based on plots of how the sample mean varies as successively greater quantiles of the distribution x_i are included is known as the "contribution to the sample mean" (Bolado-Lavin et al., 2009), with a similar method based on the contribution of the sample variance (Tarantola et al., 2012).

- Correlated input variables: variance-based sensitivity analysis (the variance decomposition given in (18.3)) is based on the assumption that the probability distributions of input variables are independent. Quite often, however, a modeller encounters the situation where this assumption is not valid. In such cases other methods must be considered, although this is a relatively immature field of research and (arguably) no one "definitive" approach exists. Some examples of methods in the literature include: (i) a method based on linear regression (Xu and Gertner, 2008); (ii) a method based on decomposition of the input covariance matrix (Li et al., 2010); (iii) a copula-based approach (Kucherenko et al., 2012), and (iv) an approach based on grouping inputs into sets which are independent of one another but variables may be correlated inside each group (Jacques et al., 2006).

The present chapter has in particular stressed the fact that a design should be explorative, in contrast to approaches that rely on a single one-at-a-time experiment, a practice which is so often seen in the literature but so poor at exploring nonlinear non-additive problems (Saltelli and Annoni, 2010). It should always be noted however, that the designs that have been described here only explore the uncertainty due to the input variables of the model. This ignores the extremely significant "structural" uncertainty that results in the model's simplification of reality, which is extremely difficult to quantify.

All the sensitivity analysis techniques here assume some knowledge of the probability distributions of the input variables, from simple ranges in the elementary effects method to full probability distributions in variance-based sensitivity analysis. As any practitioner of sensitivity and uncertainty analysis knows, finding reliable information on distributions of input variables can be the most challenging task in the whole process. Some literature gives details on eliciting distributions from experts (O'Hagan et al., 2006), although this in itself assumes that there are knowledgable experts available to consult. This issue is left as an open problem.

Lastly, every sensitivity analysis is a case apart (since every model has its idiosyncracies), and the choice of which technique to apply is highly dependent on factors such as the model run-time, the dimensionality, linearity and whether the analyst can access the model and run it at chosen input values or not. The outline that has been presented here should however be helpful in devising a suitable approach to sensitivity analysis for a wide variety of models.

References

Atkinson, A., Donev, A. and Tobias, R. (2007), *Optimum experimental designs, with SAS*, Vol. 34, Oxford University Press, USA.

Batterbee, D., Sims, N., Becker, W., Worden, K. and Rowson, J. (2011), 'Computational model of an infant brain subjected to periodic motion simplified modelling and bayesian sensitivity analysis', *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* **225**(11), 1036–1049.

Bayin, S. (2006), *Mathematical Methods in Science and Engineering*, Wiley.

Becker, W., Oakley, J., Surace, C., Gili, P., Rowson, J. and Worden, K. (2012), 'Bayesian sensitivity analysis of a nonlinear finite element model', *Mechanical Systems and Signal Processing* **32**, 18–31.

Becker, W., Rowson, J., Oakley, J., Yoxall, A., Manson, G. and Worden, K. (2011), 'Bayesian sensitivity analysis of a model of the aortic valve', *Journal of biomechanics* **44**(8), 1499–1506.

Becker, W., Worden, K. and Rowson, J. (2013), 'Bayesian sensitivity analysis of bifurcating nonlinear models', *Mechanical Systems and Signal Processing* **34**(1), 57–75.

Bishop, C. (2006), *Pattern Recognition and Machine Learning*, Springer, New York.

Bolado-Lavin, R., Castaings, W. and Tarantola, S. (2009), 'Contribution to the sample mean plot for graphical and numerical sensitivity analysis', *Reliability Engineering & System Safety* **94**(6), 1041–1049.

Borgonovo, E. (2007), 'A new uncertainty importance measure', *Reliability Engineering & System Safety* **92**(6), 771–784.

Bowman, A. and Azzalini, A. (1997), *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, Vol. 18, Oxford University Press, USA.

Budne, T. A. (1959), 'The application of random balance designs', *Technometrics* **1**(2), 139–155.

Campolongo, F., Cariboni, J. and Saltelli, A. (2007), 'An effective screening design for sensitivity analysis of large models', *Environmental Modelling & Software* **22**(10), 1509–1518.

Campolongo, F., Saltelli, A. and Campolongo, J. (2011), 'From screening to quantitative sensitivity analysis. a unified approach', *Computer Physics Communication* **182(4)**, 978–

988.

Carleson, L. (1966), 'On convergence and growth of partial sums of Fourier series', *Acta Mathematica* **116**(1), 135–157.

Chaloner, K. and Verdinelli, I. (1995), 'Bayesian experimental design: A review', *Statistical Science* pp. 273–304.

Cukier, R. I., Fortuin, C., Schuler, K. E., Petschek, A. G. and Schaibly, J. (1973), 'Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory', *The Journal of Chemical Physics* **59**, 3873–3878.

Cukier, R., Levine, H. and Shuler, K. (1978), 'Nonlinear sensitivity analysis of multiparameter model systems', *Journal of computational physics* **26**(1), 1–42.

Draper, N. R. and Smith, H. (1981), *Applied Regression Analysis*, Wiley, New York.

European Commission (2012), 'Sensitivity analysis software - Joint Research Centre', `http://sensitivity-analysis.jrc.ec.europa.eu/software/index.htm`.

Gramacy, R. B. and Lee, H. K. (2009), 'Adaptive design and analysis of supercomputer experiments', *Technometrics* **51**(2), 130–145.

Gramacy, R. B. and Taddy, M. A. (2010), 'Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models', *Journal of Statistical Software* **33**(6), 1–48.

Gu, C. (2002), *Smoothing spline ANOVA models*, Springer.

Homma, T. and Saltelli, A. (1996), 'Importance measures in global sensitivity analysis of nonlinear models', *Reliability Engineering & System Safety* **52**(1), 1–17.

Jacques, J., Lavergne, C. and Devictor, N. (2006), 'Sensitivity analysis in presence of model uncertainty and correlated inputs', *Reliability Engineering & System Safety* **91**(10), 1126–1134.

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

Kennedy, M. C. and O'Hagan, A. (2001), 'Bayesian calibration of computer models', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(3), 425–464.

Koda, M., Mcrae, G. J. and Seinfeld, J. H. (1979), 'Automatic sensitivity analysis of kinetic mechanisms', *International Journal of Chemical Kinetics* **11**(4), 427–444.

Kucherenko, S., Feil, B., Shah, N. and Mauntz, W. (2011), 'The identification of model

effective dimensions using global sensitivity analysis', *Reliability Engineering and System Safety* **96**(4), 440–449.

Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C. and Shah, N. (2009), 'Monte carlo evaluation of derivative-based global sensitivity measures', *Reliability Engineering & System Safety* **94**(7), 1135–1148.

Kucherenko, S., Tarantola, S. and Annoni, P. (2012), 'Estimation of global sensitivity indices for models with dependent variables', *Computer Physics Communications* **183**(4), 937–946.

Leamer, E. E. (1985), 'Sensitivity analyses would help', *The American Economic Review* **75**(3), 308–313.

Leamer, E. E. (2010), 'Tantalus on the road to asymptopia', *Journal of Economic Perspectives* **24**(2), 3146.

Li, G., Rabitz, H., Yelvington, P. E., Oluwole, O. O., Bacon, F., Kolb, C. E. and Schoendorf, J. (2010), 'Global sensitivity analysis for systems with independent and/or correlated inputs', *The Journal of Physical Chemistry A* **114**(19), 6022–6032.

Li, G., Wang, S.-W. and Rabitz, H. (2002), 'Practical approaches to construct RS-HDMR component functions', *The Journal of Physical Chemistry A* **106**(37), 8721–8733.

Li, G., Wang, S.-W., Rabitz, H., Wang, S. and Jaffé, P. (2002), 'Global uncertainty assessments by high dimensional model representations (HDMR)', *Chemical Engineering Science* **57**(21), 4445–4460.

Lin, Y. and Zhang, H. (2006), 'Component selection and smoothing in smoothing spline analysis of variance models', *Annals of Statistics* **34**(5), 2272–2297.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D. and Ye, K. (2006), 'Variable selection for Gaussian process models in computer experiments', *Technometrics* **48**(4), 478–490.

Magnus, J. R. (2007), 'Local sensitivity in econometrics', *Measurement in economics: A handbook* pp. 295–319.

Managing Uncertainties in Complex Models (MUCM) Toolbox (2013), `http://mucm.aston.ac.uk/MUCM/MUCMToolkit/`.

Mara, T. A. (2009), 'Extension of the RBD-FAST method to the computation of global sensitivity indices', *Reliability Engineering & System Safety* **94**(8), 1274–1281.

Moon, H., Dean, A. M. and Santner, T. J. (2012), 'Two-stage sensitivity-based group screening in computer experiments', *Technometrics* **54**(4), 376–387.

Morris, M. (1991), 'Factorial sampling plans for preliminary computational experiments', *Technometrics* **33**(2), 161–174.

Morris, M. D., Moore, L. M. and McKay, M. D. (2008), 'Using orthogonal arrays in the sensitivity analysis of computer models', *Technometrics* **50**(2).

Murphy, J. M., Sexton, D., Jenkins, G., Booth, B., Brown, C., Clark, R., Collins, M., Harris, G., Kendon, E., Betts, R. et al. (2009), 'UK climate projections science report: climate change projections'.

Nadaraya, E. (1964), 'On estimating regression', *Teoriya Veroyatnostei i ee Primeneniya* **9**(1), 157–159.

Niederreiter, H. (1992), *Quasi-Monte Carlo Methods*, Wiley Online Library.

Nyquist, H. (1928), 'Certain topics in telegraph transmission theory', *American Institute of Electrical Engineers, Transactions of the* **47**(2), 617–644.

Oakley, J. and O'Hagan, A. (2004), 'Probabilistic sensitivity analysis of complex models: a Bayesian approach', *Journal of the Royal Statistical Society B* **66**, 751–769.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., Oakley, J. E. and Rakow, T. (2006), *Uncertain judgements: eliciting experts' probabilities*, John Wiley & Sons.

Paruolo, P., Saisana, M. and Saltelli, A. (2013), 'Ratings and rankings: voodoo or science?', *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **176**(3), 609–634.

Pearson, K. (1895), 'Notes on regression and inheritance in the case of two parents', *Proceedings of the Royal Society of London* **58**, 240–242.

Pujol, G. (2009), 'Simplex-based screening designs for estimating metamodels', *Reliability Engineering & System Safety* **94**(7), 1156–1160.

Rabitz, H. and Aliş, Ö. F. (1999), 'General foundations of high-dimensional model representations', *Journal of Mathematical Chemistry* **25**(2-3), 197–233.

Rasmussen, C. and Williams, C. (2006), *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, Massachusetts.

Ratto, M. and Pagano, A. (2010), 'Using recursive algorithms for the efficient identification of smoothing spline anova models', *AStA Advances in Statistical Analysis* **94**(4), 367–388.

Ratto, M., Pagano, A. and Young, P. (2007), 'State dependent parameter metamodelling and sensitivity analysis', *Computer Physics Communications* **177**, 863–876.

Rougier, J., Goldstein, M. and House, L. (2013), 'Second-order exchangeability analysis for multimodel ensembles', *Journal of the American Statistical Association* **108**(503), 852–863.

Ruppert, D., Wand, M. and Carroll, R. (2003), *Semiparametric regression*, Vol. 12, Cambridge Univ Pr.

Saltelli, A. (2002), 'Making best use of model evaluations to compute sensitivity indices', *Computer Physics Communications* **145**(2), 280–297.

Saltelli, A. and Annoni, P. (2010), 'How to avoid a perfunctory sensitivity analysis', *Environmental Modelling & Software* **25**(12), 1508–1517.

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

Saltelli, A. and D'Hombres, B. (2010), 'Sensitivity analysis didn't help. A practitioner's critique of the Stern Review', *Global Environmental Change* **20**(2), 298–302.

Saltelli, A. and Funtowicz, S. (2013), 'When all models are wrong', *Issues in Science and Technology* pp. 30–2.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M. and Tarantola, S. (2008), *Global Sensitivity Analysis - The Primer*, John Wiley & Sons, Ltd.

Saltelli, A., Tarantola, S., Campolongo, F. and Ratto, M. (2004), *Sensitivity Analysis in Practice. A Guide to Assessing Scientific Models*, John Wiley and Sons publishers.

Saltelli, A., Tarantola, S. and Chan, K. (1999), 'Quantitative model-independent method for global sensitivity analysis of model output', *Technometrics* **41(1)**, 3956.

Satterthwaite, F. (1959), 'Random balance experimentation', *Technometrics* **1**(2), 111–137.

Schaibly, J. H. and Shuler, K. E. (1973), 'Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. II Applications', *The Journal of Chemical Physics* **59**, 3879.

Sobol', I. M. (1967), 'On the distribution of points in a cube and the approximate evaluation of integrals', *USSR Computational Mathematics and Mathematical Physics* **7**(4), 86–112.

Sobol', I. M. (1976), 'Uniformly distributed sequences with an additional uniform property',

USSR Computational Mathematics and Mathematical Physics **16**(5), 236–242.

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

Sobol, I. M. and Kucherenko, S. (2009), 'Derivative based global sensitivity measures and their link with global sensitivity indices', *Mathematics and Computers in Simulation* **79**(10), 3009–3017.

Sobol', I. M. and Kucherenko, S. S. (2005), 'On global sensitivity analysis of quasi-Monte Carlo algorithms', *Monte Carlo Methods and Applications* **11**(1), 83–92.

Sobol', I. M., Tarantola, S., Gatelli, D., Kucherenko, S. S. and Mauntz, W. (2007), 'Estimating the approximation error when fixing unessential factors in global sensitivity analysis', *Reliability Engineering and System Safety* **92**(7), 957–960.

Storlie, C. and Helton, J. (2008), 'Multiple predictor smoothing methods for sensitivity analysis: Description of techniques', *Reliability Engineering & System Safety* **93**(1), 28–54.

Strong, M., Oakley, J. E. and Chilcott, J. (2012), 'Managing structural uncertainty in health economic decision models: a discrepancy approach', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **61**(1), 25–45.

Sudret, B. (2008), 'Global sensitivity analysis using polynomial chaos expansions', *Reliability Engineering & System Safety* **93**(7), 964–979.

Tarantola, S., Gatelli, D. and Mara, T. (2006), 'Random balance designs for the estimation of first order global sensitivity indices', *Reliability Engineering & System Safety* **91**(6), 717–727.

Tarantola, S., Kopustinskas, V., Bolado-Lavin, R., Kaliatka, A., Ušpuras, E. and Vaišnoras, M. (2012), 'Sensitivity analysis using contribution to sample variance plot: Application to a water hammer model', *Reliability Engineering & System Safety* **99**, 62–73.

Tebaldi, C. and Knutti, R. (2007), 'The use of the multi-model ensemble in probabilistic climate projections', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365**(1857), 2053–2075.

Tissot, J.-Y. and Prieur, C. (2012), 'Bias correction for the estimation of sensitivity indices based on random balance designs', *Reliability Engineering & System Safety* **107**, 205–213.

Tortorelli, D. A. and Michaleris, P. (1994), 'Design sensitivity analysis: overview and review', *Inverse problems in Engineering* **1**(1), 71–105.

Watson, G. (1961), 'A study of the group screening method', *Technometrics* **3**(3), 371–388.

Watson, G. (1964), 'Smooth regression analysis', *Sankhyā: The Indian Journal of Statistics, Series A* pp. 359–372.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D. (1992), 'Screening, predicting, and computer experiments', *Technometrics* **34**(1), 15–25.

Xu, C. and Gertner, G. (2011), 'Understanding and comparisons of different sampling approaches for the Fourier amplitudes sensitivity test (FAST)', *Computational Statistics and Data Analysis* **55**, 184–198.

Xu, C. and Gertner, G. Z. (2008), 'Uncertainty and sensitivity analysis for models with correlated parameters', *Reliability Engineering & System Safety* **93**(10), 1563–1573.

Youden, W., Kempthorne, O., Tukey, J. W., Box, G. and Hunter, J. (1959), 'Discussion of the papers of Messrs. Satterthwaite and Budne', *Technometrics* **1**(2), 157–184.

Zuniga, M. M., Kucherenko, S. and Shah, N. (2013), 'Metamodelling with independent and dependent inputs', *Computer Physics Communications* **184**(6), 1570–1580.