

SENSITIVITY ANALYSIS

- Variance based sensitivity analysis (Sobol' indices)
 - global method
 - Monte Carlo evaluation → many evaluations
 - Connection with Analysis of variance
- Screening methods (Monis)
 - local method

1. Is it possible to avoid moments? (MC is heavy computationally)
2. Can we replace the original model with a surrogate? (= emulator)

Yes!

1. → Moment independent sensitivity
 - pdf
 - cdf
 2. →
 - Reduced order modelling
 - Gaussian Process regression
 - Polynomial chaos
- } different options

In both cases we can speed up the evaluation of Sobol' indices

(model → Sobol' indices ✗ (Monte Carlo)
surrogate model → Sobol' indices ✓)

Chapter 7

Sensitivity Analysis

7.1 Introduction

Sensitivity analysis (SA) quantifies the effects of input parameters variation on outputs of interest, providing a criterium to rank the most influential input parameters. Available approaches in literature to perform SA can be grouped into two main classes, *local* and *global* methods:

- *local approaches* aim at evaluating the impact of small input perturbations on the model output, which occur around nominal values (the mean of a random variable for instance). This deterministic approach consists in calculating or estimating the partial derivatives of the model at a specific point, thus providing only local information – that is, restricted to a neighborhood of the selected nominal value of the input.
- *global* approaches are based on sampling techniques on the entire parameter space. Besides basic tools such as the scatter-plots of the outputs against the input parameters, or regression analysis, we will focus on variance-based global sensitivity analysis (GSA), which aims at quantifying the amount of the variance shown by the output which is generated by the variation of a single parameter, as well as by interactions among the parameters.

In a broad sense, sensitivity analysis refers to the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input. A related practice is *uncertainty propagation*, which focuses rather on quantifying uncertainty in model output. Ideally, these two stages should be run in tandem: first, assessing which are the most relevant input parameters affecting the output of interest, then discarding the less relevant inputs (e.g., by fixing them at an arbitrary value) and analyzing how the uncertainty on the most relevant inputs impacts on the output of interest.

7.1.1 Models, inputs, outputs

To keep the discussion fairly general, we do not make any particular assumption about the model.

By *model*, in this chapter we mean any input-output relationship

$$Y = f(X_1, \dots, X_k)$$

we might interested in.

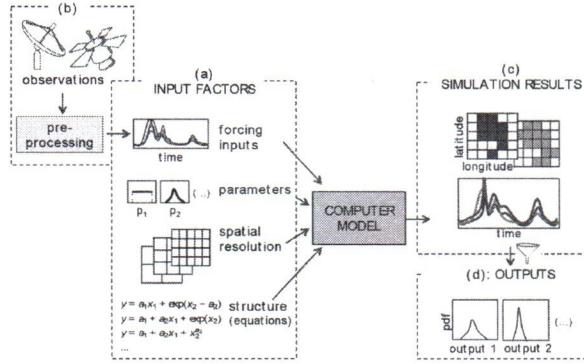


Figure 7.1: Schematic illustrating the (uncertain) ‘input factors’ and ‘outputs’ of a computer model, whose relationships are investigated by global sensitivity analysis. Taken from [34].

In particular:

- a physics-based (or physics-driven) model tries to put together accepted laws which have been attributed to the system, in order to predict its behavior, such as conservation laws and constitutive laws. A model in this case will result in a system of differential equations, under the form of ordinary differential equations (ODEs) or partial differential equations (PDEs), which then will be discretized to be numerically solved. For example, we might use Darcy’s and Ficks’ laws to understand the motion of a solute in water flowing through a porous medium;
- a data-driven model is instead purely based on observed data. Regarding the former example, we would try in this case to treat the solute as a signal and to derive its properties statistically.

The **input** is that which is allowed to vary in order to study its effect on the **output**. This latter might be described by a function of the solution of a PDE (or ODE) system.

Input factors can be broadly divided into four groups (see Fig. 7.1 for a possible example in the description of the behavior of earth system components, typically via numerical integration of differential equations over a space and time domain):

1. The equations implemented in the model to represent physical processes, for which our often-incomplete scientific knowledge might offer multiple options.
2. Set-up choices that are needed for the execution of the model on a computer, for example the selection of temporal or spatial resolutions for numerical integration of the model equations.
3. The numerical values to be attributed to the *parameters* appearing in the model equation, which are often *effective* parameters i.e. quantities that cannot directly be measured due to a scale mismatch between model element and instrument footprint. These parameters are called ‘effective’ since they are typically set to values that make the model component, e.g. the electrical conductivity of cardiac cells, approximate the behavior of the real-world system without representing the full heterogeneity of that system.
4. Any input data (system forcing, initial conditions and boundary conditions), which may be uncertain due to errors in both measurement and pre-processing. Examples of pre-processing errors include, e.g., the manipulation of raw observations (such as remote sensing data)

- time stepping,
mesh sizes,
tolerances, ..

to transform them into the actual variable needed as input to the computer model. The importance of initial and boundary conditions varies significantly with the type of model: for example, the outcome of an atmospheric model might be very sensitive to uncertainty in initial conditions, while that of a groundwater model will depend more strongly on the boundary conditions. The impact of initial conditions will also grow over the simulation period for some models (think to, e.g., numerical weather prediction models).

We will mostly focus on inputs given by parameters and, possibly, input data (system forcing, initial conditions and boundary conditions).

The specific goal of sensitivity analysis is to investigate the relative influence that input factors have on one or more model outputs. If the relationship between input factors and output is nonlinear, then small variations of an input factor may induce large variations in the output, while large variations of another input factor may induce much lower variations in the output.

When the relationships between input factors and outputs are complex, sensitivities can only be discovered ‘empirically’, i.e. by running the model against different combinations (samples) of the input factors and by analyzing the statistical properties of the input-output sample (sampling-based sensitivity analysis).

7.1.2 Sensitivity Analysis Tasks

A sensitivity analysis will in turn instruct the modelers as to the relative importance of the inputs in determining the output. An obvious consequence of this is that the modeler will remain ignorant of the importance of those variables which have been kept fixed. Clearly, the more variables we promote to the rank of input, and allow to vary, the greater the variance to be expected in the model prediction. This could lead to a situation in which we discover that, having incorporated all uncertainties, the model prediction varies so wildly as to be of no practical use.

The most relevant tasks of sensitivity analysis are :

1. identify and prioritize the most influential inputs (or *factor prioritization*).

The goal is to identify the key drivers of model behavior, that is, to identify a factor (or group of factors) which, when fixed to its true value, leads to the greatest reduction in the variance of the output. It allows to (*i*) detect and rank those factors which need to be measured in order to reduce the output variance; (*ii*) to detect the factors that have a better chance of being estimated in a subsequent numerical or experimental estimation process.

2. identify non-influential inputs in order to fix them to nominal values (or *factor fixing*).

The aim is to identify factors in the model which, left free to vary over their range of uncertainty, make no significant contribution to the variance of the output. The identified factors can then be fixed.

3. map the output behavior in function of the inputs by focusing on a specific domain of inputs (or *factor mapping*).

For example, one may want to highlight model realizations falling above the 95th percentile because these correspond to risky conditions in an industrial plant. A variance-based analysis is not suitable for this setting.

4. reduce the output variance below a given tolerance (or *variance cutting*).

This may be desirable in risk or reliability analysis, where the analyst is interested in making sure, for example, that the uncertainty of the reliability of a given system component is below a given tolerance.

5. calibrate some model inputs using some available information (real output observations, constraints, etc.).

7.2 Local vs. Global Methods

Local methods are based on the evaluation of the partial derivatives of the output with respect to each input parameter. The term *local* is related to the fact that derivatives are evaluated at a given point in the parameter space, thus providing information related to a neighborhood of that point. Evaluating partial derivatives of complex outputs is not a straightforward task: for instance, in the case of outputs depending on the solution to time-dependent nonlinear PDEs, computing output derivatives requires the solution of the so-called adjoint problem, which is a linear but time-dependent PDE, backward in time. For this reason, the use of local methods are usually restricted to small subsets of the parameter space, especially when dealing with high-dimensional parameter spaces.

Moreover, local methods are not suitable in those situations in which *mixed* effects or interactions among different parameters on the output of interest are present. Indeed, if the estimation of first-order effects requires the calculation of p first-order partial derivatives, estimating second-order effects already requires order of p^2 second-order partial derivatives, thus making local methods remarkably expensive whenever the solution of the underlying model is involved to compute.

Let us consider a model under the form

$$Y = Y(Z_1, \dots, Z_k).$$

It is not by chance that most of the sensitivity analyses met in the literature are based on derivatives. Indeed the *output (or system) derivative*

$$S_{Z_i}^p = \frac{\partial Y}{\partial Z_i}, \quad i = 1, \dots, k \quad (7.1)$$

of an output Y versus an input Z_i can be thought of as a mathematical definition of the sensitivity of Y versus Z_i . However, derivatives are only informative at the base point where they are computed and do not provide for an exploration of the rest of the space of the input factors.

When aiming at quantifying uncertainty in the presence of uncertain inputs, we shall make use of methods based on exploring the space of the input factors, based on the consideration that a handful of data points judiciously thrown into that space is far more effective, in the sense of being informative and robust, than estimating derivatives at a single data point in the centre of the space.

- **Example 7.2.1.** To better understand this fact, let us consider the simple case of a model which has a linear error-free form

$$Y = \sum_{i=1}^k \Omega_i Z_i \quad (7.2)$$

where the input factors are $\mathbf{X} = (\Omega_1, \dots, \Omega_k, Z_1, \dots, Z_k)$, with a single output variable. Let us assume first that the Ω 's are fixed coefficients, so that the true (active) factors for the model are just the Z_1, \dots, Z_k , whereas $\Omega_1, \dots, \Omega_k$ play the role of weights.

For the sake of the example we consider the weights fixed, while the individual variables have been characterized as independent and distributed normally with mean zero, i.e. $Z_i \sim N(0, \sigma_{Z_i}^2)$, $i = 1, \dots, k$, assuming that $\sigma_{Z_1} < \sigma_{Z_2} < \dots < \sigma_{Z_k}$. Then Y will also be normally distributed with mean \bar{y} and variance Ω_i^2 , with

$$\mu_Y = \sum_{i=1}^k \Omega_i \mu_{Z_i} = 0, \quad \sigma_Y^2 = \sum_{i=1}^k \Omega_i^2 \sigma_{Z_i}^2.$$

Assume for the sake of simplicity that $\Omega_1 = \dots = \Omega_k = \text{constant}$.

Let us now perform a Monte Carlo experiment, producing an input sample

$$\mathbf{M} = \begin{bmatrix} z_1^{(1)} & \dots & z_k^{(1)} \\ \dots & \dots & \dots \\ z_1^{(N)} & \dots & z_k^{(N)} \end{bmatrix}$$

The goal is to prove that if we take a MC sample of the realization of the inputs and we take the corresponding output vector and we try to decide about the importance of factors by looking at derivatives, we find a wrong conclusion. This is because, evaluating the derivative of Y w.r.t. Z_i will lead us to the conclusion that all factors should be equally important.

However, even just by drawing a simple scatterplot we can see that this is not true. So, in this context, it's not a good idea to look only at the partial derivatives.

sampling N factors' values from their distribution, and computing the output vector

$$\mathbf{Y} = \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(N)} \end{bmatrix}.$$

With this sample of model input and output one can produce k scatterplots by projecting in turn the N values of the selected output Y (assumed here to be a scalar) against the N values of each of the k input factors.

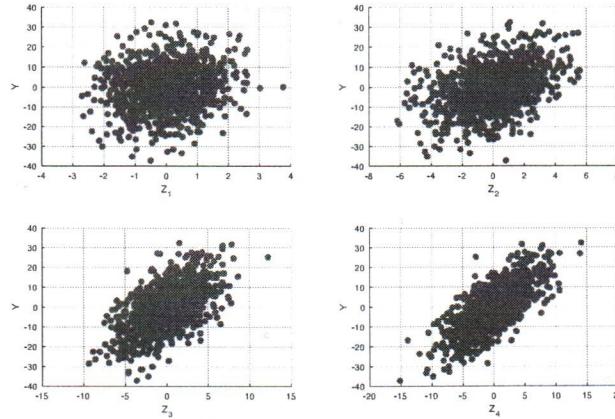


Figure 7.2: Scatterplots of Y versus Z_1, \dots, Z_4 .

These scatterplots can be used to investigate the behavior of models. The scatterplots show that Y is more sensitive to Z_4 than it is to Z_3 , and that the ordering of the input factors by their influence on Y is $Z_4 > Z_3 > Z_2 > Z_1$; however, if we decided upon the relative importance of the Z_i 's using output derivatives,

$$S_{Z_i}^p = \frac{\partial Y}{\partial Z_i}$$

we would obtain $S_{Z_i}^p = \Omega_i$, and conclude that all factors are equally important.

Input/output scatterplots are in general a very simple and informative way of running a sensitivity analysis. The challenge for sensitivity analysis, in situations with many input factors, is how to rank the factors rapidly and automatically without having to look at many separate scatterplots.

Compared to output derivatives, a possible improvement is to use a *sigma-normalized output derivatives*

$$S_{Z_i}^\sigma = \frac{\sigma_{Z_i}}{\sigma_Y} \frac{\partial Y}{\partial Z_i},$$

in which the derivatives are normalized by the factors' standard deviations (hence the σ in the superscript).

- **Example 7.2.2.** Applied to our previous model this would give

$$S_{Z_i}^\sigma = \frac{\sigma_{Z_i}}{\sigma_Y} \Omega_i \quad \text{so that} \quad \Omega_i = S_{Z_i}^\sigma \frac{\sigma_Y}{\sigma_{Z_i}}.$$

and then

$$\sigma_Y^2 = \sum_{i=1}^r \Omega_i^2 \sigma_{Z_i}^2 = \sigma_Y^2 \sum_{i=1}^k (S_{Z_i}^\sigma)^2 \quad \Rightarrow \quad \sum_{i=1}^k (S_{Z_i}^\sigma)^2 = 1,$$

The relative ordering of the Z_i 's now depends on both vectors, σ and Ω just as it should; and, more importantly, the sensitivity measures are neatly normalized to one.

- **Remark 7.2.1.** Note that care has to be taken so that each column in the matrix \mathbf{M} is a sample from the respective distribution $Z_i \sim N(\bar{z}_i, \sigma_{Z_i}^2)$. In general, and unless otherwise specified, we assume that the input factors are independent of each other, so that each one can be independently sampled from its marginal distributions.

In order to summarize the results reported in a scatterplot, the most popular method for this is to try a simple linear regression on the data of matrix \mathbf{M} and vector \mathbf{Y} , of the form

$$y^{(i)} = b_0 + \sum_{j=1}^r b_{Z_j} z_j^{(i)} \quad (7.3)$$

where the coefficients b_0, b_{Z_j} are determined by least-square computation, based on the squared differences between the y -values produced by the regression (meta)model and the actual model outputs (in our case, produced by a Monte Carlo simulation).

- **Example 7.2.3.** In the case of Example 7.2.1, since the points have been generated using a linear model, we expect that the linear regression will re-discover it, i.e. we would expect that $b_0 \approx 0$, $b_{Z_i} \approx \Omega_i, i = 1, \dots, k$.

If we suppose that the model input-output mapping can be accurately fitted by a linear model, a natural measure of the sensitivity of the output to the input Z_i is represented by the standardized regression coefficient (SRC)

$$\hat{\beta}_{Z_i} = \hat{b}_{Z_i} \frac{\sigma_{Z_i}}{\sigma_Y}, \quad i = 1, \dots, k$$

where σ_{Z_i} is the standard deviation in model input Z_i and σ_Y is the standard deviation of Y . For our model, the regression coefficients, for $N \rightarrow \infty$, will tend to

$$\hat{\beta}_{Z_i} = \hat{b}_{Z_i} \frac{\sigma_{Z_i}}{\sigma_Y} \approx \Omega_i \frac{\sigma_{Z_i}}{\sigma_Y}$$

so that, in the special case of our model (7.2) we find

$$\hat{\beta}_{Z_i} = S_{Z_i}^\sigma.$$

As a result, when the model is linear, it will also be true for the β 's that

$$\sum_{i=1}^r \hat{\beta}_{Z_i}^2 = 1.$$

If (7.3) holds and the model inputs are independently distributed, then the variance of the model output Y is given by

$$\sigma_Y^2 = \sum_{j=1}^r \hat{b}_{Z_j}^2 \sigma_{Z_j}^2;$$

hence, the fractional contribution to the variance of the model output given by Z_i is

$$\frac{b_{Z_i}^2 \sigma_{Z_i}^2}{\sum_{j=1}^r \hat{b}_{Z_j}^2 \sigma_{Z_j}^2} = \hat{\beta}_{Z_i}^2$$

Therefore, in a linear or almost linear frame, $\hat{\beta}_{Z_i}$ is an approximation of the fractional contribution to the variance of the model output associated with model input Z_i . An alternative sensitivity measure is represented by the correlation coefficient:

$$\text{Corr}(Y, Z_i) = \frac{\text{Cov}(Y, Z_i)}{\sigma_{Z_i} \sigma_Y}$$

where $\text{Cov}(Y, Z_i)$ is the covariance between Y and Z_i . If the model inputs are independent, then $\text{Corr}(Y, Z_i) = \hat{\beta}_{Z_i}$ for all $i = 1, \dots, r$.

Moreover, in case of a linear model, the *standardized regression coefficients* coincide with the sigma-normalized derivatives, that is,

$$\hat{\beta}_{Z_i} = \hat{b}_{Z_i} \frac{\sigma_{Z_i}}{\sigma_Y} = \Omega_i \frac{\sigma_{Z_i}}{\sigma_Y} = S_{Z_i}^\sigma.$$

The fact that the two measures coincide for our model can be generalized only to linear models and no further. If the model is nonlinear¹, the two measures will be different, and the β 's will be a more robust and reliable measure of sensitivity even for nonlinear models. Unlike S_{Z_i} , which is computed at the midpoint of the distribution of Z_i while keeping all other factors fixed at their midpoint, $\hat{\beta}_{Z_i}$ is the result of an exploration of the entire space of the input factors – the limit being in the dimension N of the sample.

Note that the SRC are based on a linear regression model for the output function, and are computed from the least square regression analysis applied to the output of a Monte Carlo simulation. It is also possible to show that the (model) coefficient of determination, R_Y^2 usually adopted to assess the goodness of fit of a linear model, is such that

$$R_Y^2 = \sum_{i=1}^r (\hat{\beta}_{Z_i})^2,$$

giving the percentage of the data variance which is explained by the regression model (7.3). The validity of the SRC's as a measure of sensitivity is conditional to the degree to which the regression model fits the data, i.e., to R_Y^2 . A low value for R_Y^2 indicates a poor regression model, and that it is hence unrealistic to assess the influence of the input variables based on the SRC's [11].

When $R_Y^2 = 1$,

$$\sum_{i=1}^r (\hat{\beta}_{Z_i})^2 = 1 = \sum_{i=1}^r \left(\hat{b}_{Z_i} \frac{\sigma_{Z_i}}{\sigma_Y} \right)^2$$

so that, denoting by $\text{Var}[Y]$ the variance of Y ,

$$\sum_{i=1}^r \left(\hat{b}_{Z_i} \sigma_{Z_i} \right)^2 = \sigma_Y^2 = \text{Var}[Y].$$

Summarizing,

Sigma-normalized derivatives allow to obtain a *variance decomposition* formula, and decompose the variance of the model output of interest according to the input factors. Yet we would like to do this for all models, independently of their degree of linearity; that is, we would like to decompose the variance of Y even for models with a low R_Y^2 . One such 'model-free' sensitivity measure is based on averaged partial variances, which we now move on to describe following two possible paths.

7.3 Global methods

From the late 1980s, to overcome the limitations of local methods (linearity and normality assumptions, local variations), a new class of methods has been developed in a statistical framework. In

¹Indeed, in several situations, the linear approximation does not hold. Analysts then resort to transformations (typically rank transformations) to improve the regression fit. A rank transformation involves substituting the raw data with the corresponding ranks. In other words, we take the sample and order the variables with their values.

contrast to local sensitivity analysis, it is referred to as *global sensitivity analysis* because it considers the whole variation range of the inputs.

As a way of introducing variance-based methods, we could ask the following question: Is it possible to introduce sensitivity indicators that explain the variance of the model output rather than only the fraction of the variance associated with the linear surrogate model? The answer is found in variance-based sensitivity measures, which we describe next.

7.3.1 First order effects

Considering a generic model

$$Y = f(X_1, \dots, X_k)$$

we wish to determine what would happen to the uncertainty of Y if we could fix a factor X_i ; let us denote by \mathbf{X} a vector of k uncertain model inputs X_1, \dots, X_k .

Imagine that we fix factor X_i at a particular value x_i^* . Let $\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i = x_i^*]$ be the resulting variance of Y , taken over $\mathbf{X}_{\sim i}$ (all factors but X_i , that is, fixed to the value x_i^*). We call this a *conditional variance*, as it is conditional on X_i being fixed to x_i^* . We would imagine that, having frozen one potential source of variation (X_i), the resulting variance $\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i = x_i^*]$ will be less than the corresponding total variance $\text{Var}[Y]$.

One could therefore conceive of using

$$\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i = x_i^*]$$

as a measure of the relative importance of X_i , reasoning that the smaller $\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i = x_i^*]$, the greater the influence of X_i . To make this measure not dependent on x_i^* , we take an average of this measure over all possible points x_i^* ,

$$\mathbb{E}_{X_i}[\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i]].$$

According to the *law of total variance* (or *variance decomposition formula*)

$$\mathbb{E}_{X_i}[\text{Var}_{\mathbf{X}_{\sim i}}[Y | X_i]] + \text{Var}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y | X_i]] = \text{Var}[Y]. \quad (7.4)$$

Note that, by equation (7.4),

$$\text{Var}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y | X_i]] \leq \text{Var}[Y].$$

The *conditional variance*, given by $\text{Var}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y | X_i]]$, is called the *first-order effect* of X_i on Y . Dividing this quantity by $\text{Var}[Y]$ we obtain a sensitivity measure which goes under the name of *first-order sensitivity index*.

Definition 7.3.1. The first-order sensitivity index (or first Sobol' index) of the input X_i on Y is given by

$$S_i = \frac{\text{Var}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y | X_i]]}{\text{Var}[Y]}, \quad i = 1, \dots, k.$$

The first sensitivity index is well suited for a Factor Prioritization setting.

Example 7.3.1. In the case of the linear model introduced in Example 7.2.1, we obtain

$$S_{Z_i} = \frac{\text{Var}_{Z_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y | Z_i]]}{\text{Var}[Y]} = \beta_{Z_i}^2;$$

this identity holds for linear models, as we would expect given that S_{Z_i} is a model-free generalization of $\beta_{Z_i}^2$; for nonlinear models, the two measures will differ.

Copyright © Andrea Manzoni, 2020

the highest, the more important the factor i (X_i) (because it means that we can explain a large % of the variability of the output due to X_i)

More quantitatively:
 $S_i \uparrow \Rightarrow 1 - S_i \downarrow$
 because of (7.4)
 we have that

$1 - S_i = \frac{\text{Var}_{\mathbf{X}_{\sim i}}(Y | X_i)}{\text{Var}(Y)}$

\Rightarrow if $S_i \uparrow$ we're saying that this quantity is low, namely the (average) variance of the model once we fix X_i over the whole variance of the model

(\rightarrow if this quantity is low, we're saying that the Var of X_i makes the Var of Y)

* In particular:

- additive model : $\sum_{i=1}^k S_i = 1$
- non-additive model : $\sum_{i=1}^k S_i \leq 1$

After computing these indices:

- factor prioritization: rank S_1, \dots, S_k in decreasing order
- factor fixing: fix the value of X_i for which $S_i < \epsilon$ (for a given threshold $\epsilon > 0$)

The idea is to introduce sensitivity indicators that explain the variance of the output in terms of:

- first order effects
- total effects

Moreover, while

$$\sum_{i=1}^k \hat{\beta}_{Z_i}^2 = 1$$

only for linear models, the relationship

$$\sum_{i=1}^k S_{Z_i} = 1$$

holds for any additive model. By definition, a model is additive when

$$Y = f(X_1, \dots, X_k) = \sum_{i=1}^k f_i(X_i),$$

so that it is possible to separate the effects of its input variables in a variance decomposition framework. For instance, $Y = \sum_i Z_i^2$ is a nonlinear, additive model in the Z 's.

For nonadditive models,

$$\sum_{i=1}^r S_{Z_i} \leq 1$$

that is, first-order terms do not add up to one. A nonadditive model would be obtained by allowing, in (7.2), both the Z 's and the Ω 's to become factors (that is, by treating the Ω 's no longer as constants).

For non-additive models we should look for higher order interactions to capture that part of the response of Y to X_i, X_j that cannot be written as a superposition of effects separately due to X_i and X_j . Equivalently, two factors are said to interact when their effect on Y cannot be expressed as a sum of their single effects.

Let us consider, for instance, two factors instead of one, and take

$$\frac{\text{Var}_{X_i X_j} [\mathbb{E}_{\mathbf{X}_{\sim i}, \mathbf{X}_{\sim j}} [Y | X_i, X_j]]}{\text{Var}[Y]}$$

with $i \neq j$ and rewrite the numerator – involving the variance taken over X_i, X_j and the average taken over all-but- (X_i, X_j) – skipping the indices, as $E(Y | X_i, X_j)$. The following relation holds, involving the sum of the first order and of the second order sensitivity index,

$$\text{Var}[\mathbb{E}[Y | X_i, X_j]] = V_i + V_j + V_{ij}$$

where

$$V_i = \text{Var}[\mathbb{E}[Y | X_i]], \quad V_j = \text{Var}[\mathbb{E}[Y | X_j]], \quad V_{ij} = \text{Var}[\mathbb{E}[Y | X_i, X_j]] - V_i - V_j.$$

The term V_{ij} is the *interaction* term between factors X_i, X_j and captures that part of the response of Y to X_i, X_j , that cannot be written as a superposition of effects separately due to X_i and X_j . If X_i and X_j have an additive relation in the model, then $\text{Var}[\mathbb{E}[Y | X_i, X_j]] = V_i + V_j$.

Considering a nonlinear, additive model and a nonlinear, nonadditive model, the latter model will have nonzero second-order terms, while the former model will not. This means that even for a nonadditive model we have found a way to recover (that is, to understand) 100% of the variance of Y . In a model having k factors, a full sensitivity analysis is composed of

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \sum_i \sum_{j>i} \sum_{l>j} S_{ijl} + \dots + S_{123\dots k} = 1 \tag{7.5}$$

The problem is that this summation has as many as $2^k - 1$ terms; for instance, for $k = 10$ it has 1023 terms, too many to look at in practice. The variance-based analysis can help us in these circumstances, by computing for each factor a *total effect term*, which we describe next.

7.3.2 Total Effects

In order not to be forced to compute all the interaction terms, *total effects* may be computed. We consider

$$\frac{\text{Var}_{\mathbf{X}_{\sim i}}[\mathbb{E}_{X_i}[Y \mid \mathbf{X}_{\sim i}])}{\text{Var}[Y]} = \frac{\text{Var}[\mathbb{E}[Y \mid X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_k]]}{\text{Var}[Y]}$$

As the sum of all possible sensitivity terms must be 1 according to (7.5), the difference

$$1 - \frac{\text{Var}_{\mathbf{X}_{\sim i}}[\mathbb{E}_{X_i}[Y \mid \mathbf{X}_{\sim i}])}{\text{Var}[Y]}$$

must be made up of all terms of any order that include X_i . Again for the variance decomposition formula, since

$$\mathbb{E}_{\mathbf{X}_{\sim i}}[\text{Var}_{X_i}[Y \mid \mathbf{X}_{\sim i}]] + \text{Var}_{\mathbf{X}_{\sim i}}[\mathbb{E}_{X_i}[Y \mid \mathbf{X}_{\sim i}]] = \text{Var}[Y],$$

we can define the *total effect* as follows.

Definition 7.3.2. The total effect (or total sensitivity index) of the input X_i on Y is given by

$$S_{T_i} = 1 - \frac{\text{Var}_{\mathbf{X}_{\sim i}}[\mathbb{E}_{X_i}[Y \mid \mathbf{X}_{\sim i}])}{\text{Var}[Y]} = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}}[\text{Var}_{X_i}[Y \mid \mathbf{X}_{\sim i}]]}{\text{Var}[Y]}, \quad i = 1, \dots, k.$$

S_{T_i} measures the contribution to the output variance of X_i , including all variance caused by its interactions, of any order, with any other input variables.

In other words, S_{T_i} is a measure of the overall effect of a factor on the output (inclusive of interactions) and corresponds to the expected variance that is left when all factors but X_i are fixed. In particular:

- $S_{T_i} = 0$ is a necessary and sufficient condition for X_i for being noninfluential;
- the smaller S_{T_i} , the less the X_i influences the model.

This tool may be employed in the factor fixing setting, in order to reduce the number of input parameters for the system. This could help to simplify a model in a greater sense, since we might be able to condense (lump) an entire section of our model if all factors entering that section are noninfluential.

} we can drop non-relevant parameters by fixing their values.
Note: less relevant parameters will be the most difficult to identify (if we want to rely on observations of Y)

- Remark 7.3.1.** An additional interesting feature of variance-based methods is that they allow for a concise treatment of the sensitivity of sets of factors. For example, we can compute a variance measure $S_{\mathbf{X}}$ conditioned on a subset of the input factors \mathbf{X} , that is

$$S_{\mathbf{X}} = \frac{\text{Var}[\mathbb{E}[Y \mid \mathbf{X}]]}{\text{Var}[Y]}.$$

This will include all first-order terms related to \mathbf{X} plus second- and higher-order product terms including only members of \mathbf{X} . Referring again to the model (7.2), we can imagine computing a variance measure conditioned on a subset of the input factors, e.g. on the set Ω , assuming that parameters \mathbf{Z} are determined by observations and parameters Ω have been determined by experts. In this case,

$$S_{\Omega} = \frac{\text{Var}[\mathbb{E}[Y \mid \Omega]]}{\text{Var}[Y]}$$

will include all first-order terms related to S_{Ω} plus second- and higher-order product terms including only members of S_{Ω} ; on the other hand, $S_{\mathbf{Z}} = \text{Var}[\mathbb{E}[Y \mid \mathbf{Z}]]/\text{Var}[Y]$ contains all nonzero first-order terms plus the null second- and higher-order terms internal to \mathbf{Z} . Hence, assuming that the

nonzero sensitivity indices are the first and the second, by computing $S_{\Omega, Z} = \text{Var}[Y] - S_{\Omega} - S_Z$ it is possible to assess the effects only due to the interaction terms. Treating sets of parameters can be extremely useful when we want to distinguish data uncertainty from experts' uncertainties, for instance.

Remark 7.3.2. We have shown the equivalence of sigma-normalized coefficients S_i^{σ} , regression coefficients β_i and variance-based first-order sensitivity indices S_i for linear models, as well as how S_i represents a model-free extension of the variance decomposition scheme to models of unknown linearity. In general, for a model of unknown linearity, monotonicity and additivity, variance-based measures constitute a good means of tackling settings such as factor fixing and factor prioritization. The main problem with variance-based measures is their computational cost. Estimating the sensitivity coefficients takes many model runs, as we will see.

Finally, the general advise is to work on uncorrelated samples as much as possible, e.g. by treating dependencies as explicit relationships with a noise term. Indeed, dependent input samples are more laborious to generate and the sample size needed to compute sensitivity measures for non-independent samples is much higher than in the case of uncorrelated samples.

7.4 Screening methods

local methods but they're not relying on derivatives (not in an analytical way, at least)

Screening methods are qualitative methods for studying sensitivities on models containing several tens of input variables. They are based on a discretization of the inputs in levels, allowing a fast exploration of the model (or code) behavior. These methods are adapted to a large number of inputs; practice has often shown that only a small number of inputs are influential. The aim of this type of method is to identify the non-influential inputs with a small number of model calls while making realistic hypotheses on the model complexity. The model is therefore simplified before using other SA methods, more subtle but more costly.

The most engineering-used screening method is based on the so-called *One-At-a-Time* (OAT) design, where each input is varied while fixing the others (see Fig. 7.3). Among several possible choices, here we only sketch some basic ideas behind the *Elementary Effects* (EE) (or *Morris*) method; when the number of experiments is of the same order than the number of inputs, the classical theory of experimental design applies, for example with the so-called factorial fractional design.

7.4.1 Construction

Despite its weakness, the EE method is one of the most used screening methods in sensitivity analysis. It is applied to identify non-influential inputs for a computationally costly mathematical model or for a model with a large number of inputs, where the costs of estimating other sensitivity analysis measures such as the variance-based measures is not affordable. Like all screening, the EE method provides qualitative sensitivity analysis measures, i.e. measures which allow the identification of non-influential inputs or which allow to rank the input factors in order of importance, but do not quantify exactly the relative importance of the inputs.

Indeed, the EE method allows to classify the inputs in three groups:

- inputs having negligible effects,
- inputs having large linear effects without interactions,
- inputs having large non-linear and/or interaction effects.

The method consists in discretizing the input space for each variable, then performing a given number of OAT designs. Such designs of experiments are randomly chosen in the input space, and the variation direction is also random. The repetition of these steps allows the estimation of elementary effects for each input. From these effects, sensitivity indices are derived.

Idea:

Instead of covering the input parameters space by samples and then perform a suitable Monte Carlo approximation of those expectation to compute Sobol indices, the idea is to perform parameter variation one-at-the-time. We keep all the parameters fixed and then we change the value of a single parameter component (a single factor) and then we decide, on the basis of incremental ratios, whether or not that parameter is relevant. Performing this is, in a sense, an approximation of the derivatives.

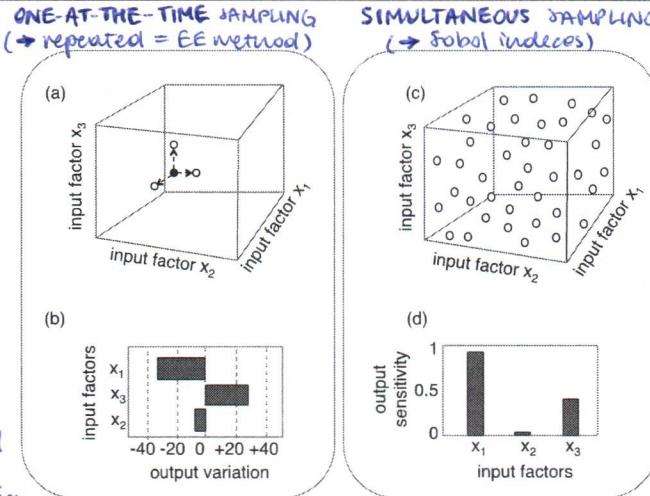
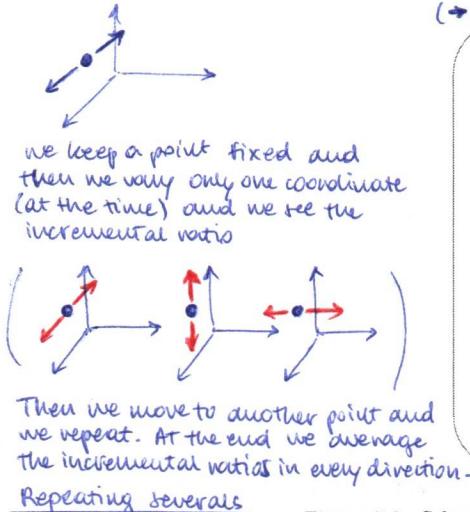


Figure 7.3: Schematic illustrating the difference between One-At-the-Time (OAT) sampling (a) and associated SA results (b) against All-At-the-Time (simultaneous) sampling (c) and corresponding sensitivity indices (d). Taken from [34].

Repeating several OAT designs, we can extract elementary effects for each point.

Averaging over all the points will give us an idea on if the input X_i is important or not. The variance among these values will give us an idea on the role of the parameters being uniform in the space or not.

The variance is somehow the difference between screening method and methods based on the derivatives
However: curse of dimensionality. If the dimension of the space is too high, we get troubles in exploring the whole parameters space.
(in extreme cases)

numerical approximation of the partial derivative

$$\frac{\partial Y}{\partial X_j} \text{ (or } \frac{\partial f}{\partial X_j})$$

Let us assume to consider a mathematical model with k input factors, and denote by Y be the output of interest (a scalar for simplicity):

$$Y = f(X_1, X_2, \dots, X_k).$$

The original EE method of Morris provides two sensitivity measures for each input factor:

- the measure μ^* , assessing the overall importance of an input factor on the model output; (μ_j^*)
- the measure σ , describing nonlinear effects and interactions. (σ_j)

These two measures are obtained through a design based on the construction of a series of trajectories in the space of the inputs, where inputs are randomly moved One-At-a-Time.

In this design, inputs are assumed to vary in the k -dimensional unit cube $[0, 1]^k$, across p selected levels. The region of experimentation Ω is thus a k -dimensional p -level grid. For a given value of \mathbf{X} , the elementary effect of the j -th input factor is defined as

$$EE_j = \frac{f(X_1, \dots, X_{j-1}, X_j + \Delta, X_{j+1}, \dots, X_k) - f(X_1, \dots, X_k)}{\Delta} = \frac{f(\mathbf{X} + \Delta \mathbf{e}_j) - f(\mathbf{X})}{\Delta}$$

where p is the number of levels, Δ is a predetermined integer value in $\{1/(p-1), \dots, 1-1/(p-1)\}$, $\mathbf{X} = (X_1, \dots, X_k)$ is any selected value in Ω such that the transformed point $\mathbf{X} + \mathbf{e}_i \Delta$ is still in Ω for each index $i = 1, \dots, k$ and \mathbf{e}_i is the unitary vector of the canonical basis. Note that the system (or output) derivative (7.1) is nothing but the limit version of the elementary effect EE_i when Δ tends to zero. This is a OAT design.

Remark 7.4.1. Each trajectory is then made by $(k+1)$ points since input factors move one by one of a step Δ in $\{0, 1/(p-1), 2/(p-1), \dots, 1\}$ while all the others remain fixed. Note that all points generated along this trajectory belong to the surface of the sphere centered at \mathbf{X} with radius Δ .

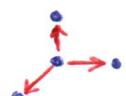
The EE method consists in discretizing the input space for each variable, then performing a given number of OAT design. Such designs of experiments are randomly chosen in the input space,

— we discretize each one of the k directions in p subintervals

— $\Delta \mathbf{e}_j$ = step in the direction \mathbf{e}_j of amplitude Δ

taking Δ in here we are constraining to remain on the discretized grid

for example in 3D we need all the $4 \cdot (3+1)$ blue points to "define" a trajectory:



and the variation direction is also random. The repetition of these steps allows the estimation of elementary effects for each input. From these effects, sensitivity indices are derived.

Let us denote by r the number of OAT designs (a frequent choice is to set parameter r between 4 and 10). Let us discretize the input space with a k -dimensional grid with p levels for each input, and denote by $EE_j^{(i)}$ the elementary effect of the j -th variable obtained at the i -th replica,

$$EE_j^{(i)} = \frac{f(\mathbf{X}^{(i)} + \Delta \mathbf{e}_j) - f(\mathbf{X}^{(i)})}{\Delta}, \quad j = 1, \dots, k$$

(we can also consider Δ_i if we want to change replica by replica)

for $i = 1, \dots, r$. Indices are then obtained as follows:

$$\mu_j^* = \frac{1}{r} \sum_{i=1}^r |EE_j^{(i)}|, \quad j = 1, \dots, k,$$

the sample mean of the absolute value of the elementary effects, and

$$\sigma_j = \sqrt{\frac{1}{r} \sum_{i=1}^r \left(EE_j^{(i)} - \frac{1}{r} \sum_{i=1}^r EE_j^{(i)} \right)^2}, \quad j = 1, \dots, k,$$

the standard deviation of the elementary effects.

The interpretation of the indices is the following:

- μ_j^* is a measure of influence of the j -th input on the output. The larger μ_j^* is, the more the j -th input contributes to the dispersion of the output. The use of μ_j^* instead of

$$\mu_j = \frac{1}{r} \sum_{i=1}^r EE_j^{(i)}, \quad j = 1, \dots, k,$$

solves the problem of the effects of opposite signs which occurs when the model is non-monotonic and which can cancel each other out, thus resulting in a low value for μ_j ;

- σ_j is a measure of nonlinear and/or interaction effects of the j -th input. If σ_j is small, elementary effects have low variations on the support of the input. Thus the effect of a perturbation is the same all along the support, suggesting a linear relationship between the studied input and the output. On the other hand, the larger σ_j is, the less likely the linearity hypothesis is. Thus a variable with a large σ_j will be considered having non-linear effects, or being implied in an interaction with at least one other variable.

In its simplest form, estimating the indices above requires the evaluation of Y twice, once at the selected values and again after increasing X_j by the quantity Δ . Thus, the total computational cost of the experiment would be $n = r(k+1)$ runs, where r is the selected size of each sample.

7.4.2 The rationale behind

The basic idea of the EE method is that the distribution of elementary effects associated with the j -th input factor is obtained by randomly sampling different \mathbf{X} from Ω , and is denoted by F_j , that is, $EE_j \sim F_j$. The F_j distribution is finite and, if p is even and Δ is chosen to be equal to $p/(2(p-1))$, the number of elements of F_j is $p^{k-1}[p - \Delta(p-1)]$.

The purpose of the experimental plan is then collecting random samples from each distribution F_j of elementary effects associated with each input factors. Analysis of those distributions will assess the relative importance of the input factors. In particular: a large measure of the distribution mean value indicates an input factor with an high *overall* influence on the output, while a large measure of spread (i.e., an high value of standard deviation) indicates an input factor that is involved in interactions with other factors or whose effect is nonlinear.

The sensitivity measures μ_j and σ_j , proposed by Morris, are respectively the estimates of the mean and the standard deviation of the distribution F_j . In particular, μ_j assesses the overall influence of the factor X_j on the output, while σ_j estimates the ensemble of the factor's effects, whether nonlinear and/or due to interactions with other factors. The use of the mean μ_j , as we have seen, can be replaced with μ_j^* , which is defined as the estimate of the mean of the distribution G_j of the absolute values of the elementary effects, i.e. $|EE_j| \sim G_j$. Indeed, Morris approach may be problematic in the case of large models with multiple outputs, for which the computation of μ_j^* might be preferable. It has also been shown that μ_j^* is a good proxy of the total sensitivity index² S_{Tj} .

On the other hand, an intuitive explanation of the meaning of σ is the following. Assume that for the factor X_j we obtain a high value of σ . The elementary effects relative to this factor thus differ notably from one another, implying that the value of an elementary effect is strongly affected by the choice of the sample point at which it is computed, i.e. by the choice of the other factors' values. By contrast, a low value of σ_j indicates very similar values among the elementary effects, implying that the effect of X_j is almost independent of the values taken by the other factors.

7.4.3 Why might it fail? — CURSE OF DIMENSIONALITY: selecting \bar{X} , all the points that are related with the evaluation of this one-at-a-time design will belong to the surface of the sphere centered in \bar{X} of radius Δ .

Hyperspheres included in (and tangent to) the hypercube have volumes that quickly decrease for increasing k .
For instance:



the red zones are the ones we'll never be able to reach through $\bar{X} + \Delta^2$

(when we're in \bar{X} we move in any direction \vec{e}_i but we remain only in the sphere centered in \bar{X} and of ray Δ)

What is the area we are not reaching?
 $r(2) = \frac{\text{ratio sphere}}{\text{ratio unit cube}} = \frac{\pi(1/2)^2}{1} \approx 0.78$

area ($\textcolor{red}{-}$) = $1 - 0.78 = 0.22$.

What happens in higher dimension?



$$r(3) = 0.52$$

$$\text{area}(\textcolor{red}{-}) \approx 0.48$$

The unreachable area grew a lot!

If, for example, we go in a space of 12 dimensions:

$$r(12) \approx 0.000326,$$

the resulting unreachable area is $1 - r(12)$, unacceptable.

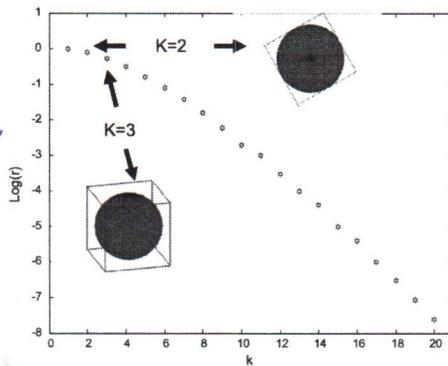


Figure 7.4: The curse of dimensionality. In $k = 3$ dimensions the volume of the sphere internal to a cube and tangent to its face is $r \sim 0.5$. r goes rapidly to zero with increasing k .

In a system with just two factors ($k = 2$) the area of the circle inscribed in the unit square, and hence the ratio of the 'partially' explored to the total area, is

$$r(2) = \pi(1/2)^2 \approx 0.78,$$

whereas in three dimensions,

$$r(3) = (4\pi/3)(1/2)^3 \approx 0.52.$$

²Recall that the total sensitivity index S_{Tj} is a measure of the overall effect of a factor on the output (inclusive of interactions) and corresponds to the expected variance that is left when all factors but X_j are fixed.

The general formula for the volume of the hypersphere of radius $1/2$ in k dimensions is:

$$r(k) = \frac{\pi^k / 2}{\Gamma(k/2 + 1)} \cdot \frac{1}{2^k}$$

where

$$\Gamma(n) = (n - 1)! \text{ for } n \in \mathbb{N}, \quad \Gamma(x + 1) = x\Gamma(x) \text{ for } x \in \mathbb{R} \setminus \mathbb{N},$$

with $\Gamma(1) = 1$ and $\Gamma(1/2) = \sqrt{\pi}$. For instance, in 12 dimensions, the fraction of the hyperspace explored is equal to $r \approx 0.000326$, less than one-thousandth of the factors' space. This is one of the many possible ways to illustrate the so-called curse of dimensionality. An OAT sensitivity analysis thus designed is therefore perfunctory in a model free setting. In a high dimensional space, a derivative based analysis and an OAT based one are practically equivalent, i.e. they are both local and thus by definition non-explorative. Even by considering n OAT designs to perform a screening based on Morris' method could potentially lead to wrong results. See, e.g., [41] for further details.

7.5 Variance-based methods

Let us now go back to the first-order and total sensitivity indices introduced in Sect. 7.3.1–7.3.2 and show how to evaluate them from a practical point of view. The main drawback of variance-based measures, as we will see by the end of this section, is indeed their computational cost.

7.5.1 Sobol' indices

The most relevant tool among variance-based SA are Sobol' indices, which roughly speaking coincide with the sensitivity indices introduced in Sect. 7.3.1. From a black box perspective, any model may be viewed as a function $Y = f(\mathbf{X})$, where \mathbf{X} is a vector of k uncertain model inputs X_1, \dots, X_k and Y is a chosen univariate model output (note that this approach examines scalar model outputs, but multiple outputs can be analysed by multiple independent sensitivity analyses). Furthermore, it will be assumed that the inputs are independently and uniformly distributed within Ω^k , the k -th dimensional unit hypercube,

$$\Omega^k = \{\mathbf{X} \in \mathbb{R}^k : 0 \leq X_i \leq 1, i = 1, \dots, k\}.$$

This incurs no loss of generality because any input space can be transformed onto this unit hypercube. $f(\mathbf{X})$ may be decomposed in the following way

$$Y = f_0 + \sum_{i=1}^k f_i(X_i) + \sum_{i < j}^k f_{ij}(X_i, X_j) + \dots + f_{1,2,\dots,k}(X_1, X_2, \dots, X_k) \quad (7.6)$$

In which each individual term is a function of the factors in its index, that is, $f_i = f_i(X_i)$, $f_{ij} = f_{ij}(X_i, X_j)$ and so on. This decomposition is not a series decomposition, as it has a finite number of terms. Of the $2^k - 1$ terms, one is constant (f_0), k are first-order functions (f_i), $\binom{k}{2}$ are second order functions (f_{ij}), and so on.

This expansion, called *High-Dimensional Model Representation* (HDMR) – also known as *functional Analysis of Variance* (FANOVA) decomposition – is not unique, meaning that, for a given model f , there could be infinite choices for its terms. If each term in the expansion above is such that the integral of each summand $f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s})$ over any of its independent variables is zero, i.e.

$$\int_0^1 f_{i_1, \dots, i_s} dX_{i_k} = 0 \quad \text{for } 1 \leq k \leq s \quad (*)$$

then all the terms of the decomposition are orthogonal in pairs, i.e.

Sobol proved that if each element has zero mean (*), then all terms are pairwise ⊥:

$$\int f(x_i) dx_i = 0$$



$$\int f(x_i) f(x_j) dx_i dx_j = 0$$

$$\int_{\Omega^k} f_{i_1, \dots, i_s} f_{j_1, \dots, j_t} d\mathbf{X} = 0 \quad \text{for } i_1, \dots, i_s \neq j_1, \dots, j_t.$$

With the above assumptions, the decomposition in (7.6) is unique whenever $f(\mathbf{X})$ is integrable over Ω^k . Moreover, the terms in the decomposition may be derived analytically.

This leads to definitions of the terms of the functional decomposition in terms of conditional expected values,

$$f_0 = \int_{\Omega^k} f(\mathbf{X}) d\mathbf{X} = \mathbb{E}[Y]$$

$$f_i(X_i) = \int_{\Omega^{k-1}} f(\mathbf{X}) d\mathbf{X}_{\substack{\circlearrowleft \\ \sim i}} - f_0 = \mathbb{E}[Y|X_i] - f_0 \quad (7.7)$$

$$f_{ij}(X_i, X_j) = \int_{\Omega^{k-2}} f(\mathbf{X}) d\mathbf{X}_{\substack{\circlearrowleft \\ \sim i, j}} - f_i(X_i) - f_j(X_j) - f_0 = \mathbb{E}[Y|X_i, X_j] - f_0 - f_i - f_j \quad (7.8)$$

and so on.

from which it can be seen that f_i is the effect of varying X_i alone (known as the main effect of X_i), and f_{ij} is the effect of varying X_i and X_j simultaneously, additional to the effect of their individual variations. This is known as a second-order interaction. Higher-order terms have analogous definitions.

Now, further assuming that $f(\mathbf{X})$ is square-integrable, the functional decomposition may be squared and integrated to give

$$\int_0^1 f^2(\mathbf{X}) d\mathbf{X} - f_0^2 = \sum_{s=1}^k \sum_{i_1 < \dots < i_s} \int f_{i_1 \dots i_s}^2 dX_{i_1} \dots dX_{i_s}$$

Notice that the left hand side is equal to the variance of Y , and the terms of the right hand side are variance terms, now decomposed with respect to sets of the X_i .

This finally leads to the decomposition of variance expression (or ANOVA-HDMR decomposition),

$$\text{Var}[Y] = \sum_{i=1}^k V_i + \sum_{i < j}^k V_{ij} + \dots + V_{12\dots k} \quad (7.9)$$

thanks to the orthogonality properties between any pair of terms in the expansion, where

$$V_i = \text{Var}_{X_i} [\mathbb{E}_{\mathbf{X}_{\sim i}} [Y | X_i]],$$

$$V_{ij} = \text{Var}_{X_{ij}} [\mathbb{E}_{\mathbf{X}_{\sim ij}} [Y | X_i, X_j]] - V_i - V_j \quad \text{and so on.}$$

we can write it because we have orthogonality between any pair of terms in the expansion

and so on. In the latter equation, $\text{Var}_{X_{ij}} [\mathbb{E}_{\mathbf{X}_{\sim ij}} [Y | X_i, X_j]]$ measures the joint effect of the pair (X_i, X_j) on Y ; V_{ij} is instead the joint effect of X_i and X_j minus the first-order effects for the same factors. The above variance decomposition shows how the variance of the model output can be decomposed into terms attributable to each input, as well as the interaction effects between them. Together, all terms sum to the total variance of the model output.

First-order indices

A direct variance-based measure of sensitivity S_i , called the *first-order sensitivity index*, or *main effect index*, is given by

$$S_i = \frac{V_i}{\text{Var}[Y]} = \frac{\text{Var}_{X_i} [\mathbb{E}_{\mathbf{X}_{\sim i}} [Y | X_i]]}{\text{Var}[Y]}.$$

This is the contribution to the output variance of the main effect of X_i , therefore it measures the effect of varying X_i alone, but averaged over variations in other input parameters. It is standardized by the total variance to provide a fractional contribution. Higher-order interaction indices S_{ij} , S_{ijk} and so on can be formed by dividing other terms in the variance decomposition by $\text{Var}(Y)$. Note that this has the implication that

$Y = f(\vec{x})$
 \rightarrow HDMR / FANOVA not unique
 \rightarrow $H_p : f(\vec{x})$ integrable
 Element of the decompos.
 was zero mean
 \Rightarrow unique decomposition +
 all elements are pairwise \perp
 \rightarrow $H_p : f(\vec{x})$ squared-integrable
 \Rightarrow decomposition of variance(Y)
 \rightarrow first-order vs. total-effect

Variance-based methods

$$\sum_{i=1}^k S_i + \sum_{i < j}^k S_{ij} + \dots + S_{12\dots k} = 1$$

as we can easily obtain by dividing both sides of (7.9) by $\text{Var}[Y]$.

Total-effect index

Using the S_i , S_{ij} and higher-order indices given above, one can build a picture of the importance of each variable in determining the output variance. However, when the number of variables is large, this requires the evaluation of $2^k - 1$ indices, which can be too computationally demanding. For this reason, a measure known as the *total-effect index* or *total-order index*, S_{T_i} , is used. This measures the contribution to the output variance of X_i , including all variance caused by its interactions, of any order, with any other input variables. It is given as,

$$S_{T_i} = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}} [\text{Var}_{X_i} [Y | \mathbf{X}_{\sim i}]]}{\text{Var}[Y]} = 1 - \frac{\text{Var}_{\mathbf{X}_{\sim i}} [\mathbb{E}_{X_i} [Y | \mathbf{X}_{\sim i}]]}{\text{Var}[Y]}$$

For instance, for a three-factor model the total effect of X_1 is the sum of all the terms in equation (7.10) where the factor X_1 is considered:

$$S_{T_1} = S_1 + S_{12} + S_{13} + S_{123}.$$

Note that unlike the S_i ,

$$\sum_{i=1}^k S_{T_i} \geq 1$$

due to the fact that the interaction effect between e.g. X_i and X_j is counted in both S_{T_i} and S_{T_j} . In fact, the sum of the S_{T_i} will only be equal to 1 when the model is purely additive.

As already remarked, total indices are useful in sensitivity analysis, as they give information on the nonadditive features of the model. For a purely additive model $\sum_{i=1}^k S_i = 1$, while for a given factor X_j a significant difference between S_{T_j} and S_j highlights important interaction involving that factor. In particular, the condition $S_{T_i} = 0$ is necessary and sufficient for X_i to be a non-influential factor. If $S_{T_i} = 0$, then X_i can be fixed at any value within its range of uncertainty without appreciably affecting the value of the output variance $\text{Var}[Y]$ – in other words, total indices are suitable for the factor fixing setting.

7.5.2 Numerical evaluation of Sobol' indices

For analytically tractable functions, the indices above may be calculated analytically by evaluating the integrals in the decomposition. However, in the vast majority of cases they are estimated – this is usually done by the Monte Carlo method, exploiting the so-called *Sobol' sequences*.

Indeed, at first sight, it might seem that the computational strategy for the estimation of conditional variances such as V_i and $\text{Var}_{X_{ij}} [\mathbb{E}_{\mathbf{X}_{\sim ij}} [Y | X_i, X_j]]$ would be the cumbersome, brute-force computation of the multidimensional integrals in the space of the input factors. To obtain, for example, V_i , one would first use a set of Monte Carlo points to estimate the inner expectation for a fixed value of X_i , and then repeat the procedure many times for different X_i values to estimate the outer variance. To give an indication, if 1000 points were used to get a good estimate of the conditional mean $\mathbb{E}_{\mathbf{X}_{\sim i}} [Y | X_i]$ and the procedure were repeated 1000 times to estimate the variance, then we would need 10^6 points just for one sensitivity index. This is in fact not necessary, as the computation can be accelerated via existing shortcuts.

Crude Monte Carlo thus might reveal extremely inefficient. This is the reason why we rather prefer to substitute (pseudo)random sequences with low-discrepancy sequences to improve the efficiency of the estimators, relying on quasi-Monte Carlo methods. A common option when dealing with sensitivity analysis is provided by Sobol' sequences.

$\text{Var}_{X_i} (\mathbb{E}_{\mathbf{X}_{\sim i}} [Y | X_i])$
 expectation,
 N_1 samples
 ↓
 outer variance,
 N_2 samples
 $\Rightarrow N_1 \cdot N_2$ evaluations
 of the model.
 (too much!)

Crude Monte Carlo is
 inefficient in this
 setting. We turn to low
 discrepancy sequences
 (quasi Monte Carlo)

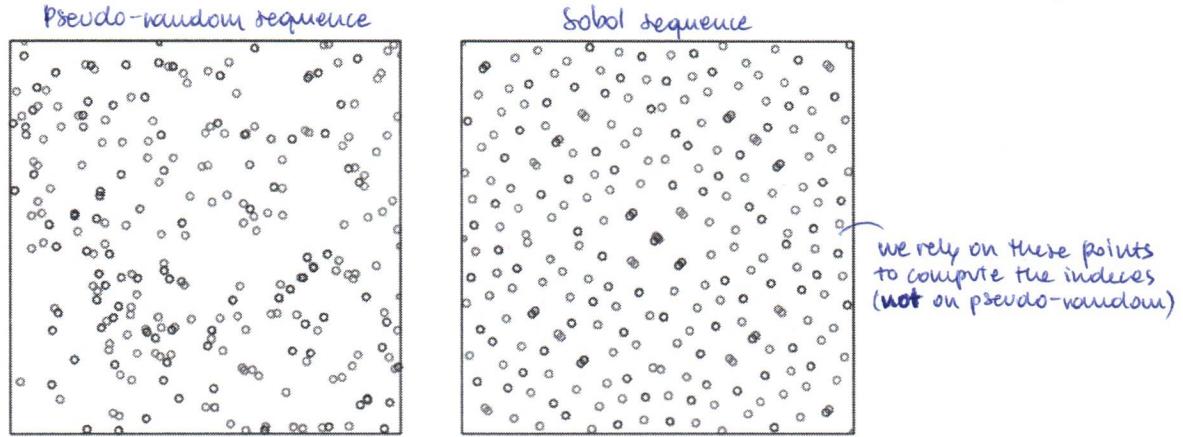


Figure 7.5: 256 points from the first 256 points for the 2,3 Sobol sequence (top) compared with a pseudorandom number source (bottom). The Sobol sequence covers the space more evenly. (red = 1, ..., 10, blue = 11, ..., 100, green = 101, ..., 256).

To calculate the indices using the (quasi) Monte Carlo method, the following steps are used³:

- generate a $N \times 2k$ matrix of random numbers (k is the number of inputs), obtained as input realizations from a Sobol' quasi-random sequence (obtained e.g. with the Matlab function `sobolset`);
- define two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times k}$ each containing half of the samples,

$$\mathbf{A} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_{k-1}^{(1)} & x_k^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_{k-1}^{(N)} & x_k^{(N)} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} x_{k+1}^{(1)} & x_{k+2}^{(1)} & \cdots & x_{2k-1}^{(1)} & x_{2k}^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ x_{k+1}^{(N)} & x_{k+2}^{(N)} & \cdots & x_{2k-1}^{(N)} & x_{2k}^{(N)} \end{bmatrix};$$

each row (both in \mathbf{A}, \mathbf{B}) is a point in the parameter space (dim = k)

N is called a base sample; to give an order of magnitude, N can vary from a few hundreds to a few thousands.

- construct k matrices of input realizations $\mathbf{C}_i \in \mathbb{R}^{N \times k}$, $i = 1, \dots, k$, using all columns of \mathbf{B} except the i -th column taken from \mathbf{A} : = all components, but the i -th, are fixed

$$\mathbf{C}_i = \begin{bmatrix} x_{k+1}^{(1)} & \cdots & \boxed{x_i^{(1)}} & \cdots & x_{2k}^{(1)} \\ \vdots & & \vdots & & \vdots \\ x_{k+1}^{(N)} & \cdots & \boxed{x_i^{(N)}} & \cdots & x_{2k}^{(N)} \end{bmatrix}; \quad \mathbf{B}_{-i} \quad \mathbf{A}_i$$

- compute the model output for all the vectors of parameters given by the rows of \mathbf{A} , \mathbf{B} and \mathbf{C}_i . The results are $k+2$ vectors of model outputs $\mathbf{y}_A = f(\mathbf{A})$, $\mathbf{y}_B = f(\mathbf{B})$ and $\mathbf{y}_{C_i} = f(\mathbf{C}_i)$ of dimension $N \times 1$, giving a total of $N(k+2)$ model evaluations.

These vectors are all we need to compute the first- and total-effect indices S_i and S_{T_i} , $i = 1, \dots, k$. There are a number of possible Monte Carlo estimators available for both indices. Two that are currently in general use are

$k+2$ because
 A, B are 2
 C_i are k ($i=1, \dots, k$)

³The method offered here is attributable to Saltelli (2002) and represents an extension of the original approach provided by Sobol' (1990) and Homma and Saltelli (1996).

$$S_i = \frac{\text{Var}_{X_i} [\mathbb{E}_{\mathbf{X} \sim i} [Y | X_i]]}{\text{Var}[Y]} \frac{\frac{4}{N} \mathbf{y}_A^\top \mathbf{y}_{C_i} - \bar{y}_A^2}{\frac{4}{N} \mathbf{y}_A^\top \mathbf{y}_A - \bar{y}_A^2}, \quad i = 1, \dots, k. \quad (7.11)$$

where

$$\bar{y}_A = \frac{1}{N} \sum_{j=1}^N \mathbf{y}_A^{(j)}$$

denotes the sample mean of the vector \mathbf{y}_A , and provides an estimator of $f_0 = E(Y)$.

Note that in the scalar product $\mathbf{y}_A \cdot \mathbf{y}_{C_i}$ values of Y computed from A are multiplied by values of Y for which all factors but X_i are resampled while the values of X_i remain fixed. If X_i is noninfluential, then high and low values of \mathbf{y}_A and \mathbf{y}_{C_i} are randomly associated. If X_i is influential, then high (or low) values of \mathbf{y}_A will be preferentially multiplied by high (or low) values of \mathbf{y}_{C_i} increasing the value of the resulting scalar product.

Similarly, the method estimates total-effect indices as follows:

$$S_{T_i} = 1 - \frac{\text{Var}_{\mathbf{X} \sim i} [\mathbb{E}_{X_i} [Y | \mathbf{X} \sim i]]}{\text{Var}[Y]} = 1 - \frac{\frac{4}{N} \mathbf{y}_B^\top \mathbf{y}_{C_i} - \bar{y}_A^2}{\frac{4}{N} \mathbf{y}_A^\top \mathbf{y}_A - \bar{y}_A^2}, \quad i = 1, \dots, k. \quad (7.12)$$

The accuracy of the estimators is of course dependent on N . The value of N can be chosen by sequentially adding points and calculating the indices until the estimated values reach some acceptable convergence. For this reason, when using low-discrepancy sequences, it can be advantageous to use those that allow sequential addition of points (such as the Sobol sequence), as compared to those that do not (such as Latin hypercube sequences).

Because there are k factors, estimating the S_i and the S_{T_i} for all input variables requires $N + N$ runs of the model for matrices A, B , plus kN runs to estimate k times the output vector of matrix C_i . The total cost is hence $N(k+2)$, much lower than the N^2 runs of the brute force method (fix N times the value of one parameter and consider N times the values assumed by the other parameters). Therefore, computational expense can quickly become a problem when the model is expensive to evaluate. In such cases, we can rely on emulators to reduce the computational cost of estimating sensitivity indices.

7.5.3 An example: the Ishigami-Homma function

We apply the variance-based sensitivity analysis technique described so far to the case of the Ishigami-Homma function,

$$f(\mathbf{X}) = \sin(X_1) + a \sin^2(X_2) + b X_3^4 \sin(X_1)$$

where all $X_i \sim \mathcal{U}(-\pi, \pi)$. It can be noticed that: (i) X_1 seems to be the most influential input; (ii) X_2 seems to be non-influential. See, for instance, the scatterplots of the values of Y against X_1, X_2, X_3 reported in Figure 7.6.

This is confirmed by variance-based SA: in fact, the total effects index of X_1 is the highest and the total effects of X_2 is almost zero; these numbers can be also derived analytically. Computed Sobol' indices are reported and, in this case, compared with the analytical expressions of those indices, which can be computed analytically for the case at hand.

```
main      total
X1:  0.3778  1.0677
X2: -0.0116 -0.0477
X3: -0.0259  0.6175

sum:  0.3403  1.6375
```

We can then plot main and total effects and compare the values estimated numerically, also analyzing convergence of sensitivity indices; results are reported in Figures 7.7–7.8.

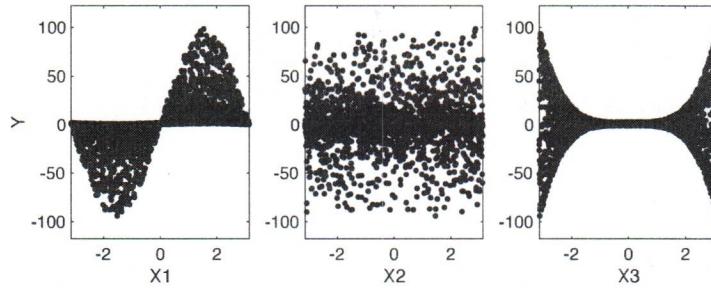


Figure 7.6: Scatterplot of the function values against each of the input factors.

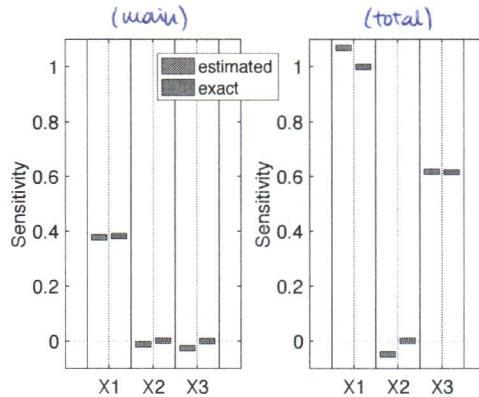


Figure 7.7: Computed Sobol' indices and analytical expressions (available for the case at hand).

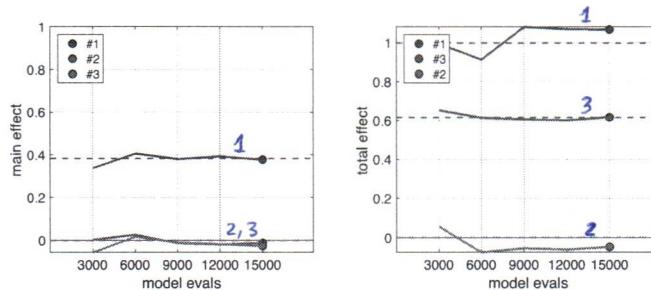


Figure 7.8: Convergence of sensitivity indices with respect to the number of output evaluations.

7.5.4 The case of dynamical systems

We sketch a possible way to extend a variance-based sensitivity analysis to the case of time-dependent outputs relying on the recent paper [1]. This is also a possible way to further highlight several interactions among different techniques already addressed during the course.

In the case the model is described by a dynamical systems, a further generalization of the Sobol'

indices is possible. The ability to make reliable predictions from time-dependent mathematical models of the form

$$Y = f(t, \mathbf{p}), \quad t \in [0, T],$$

where $\mathbf{p} \in \mathbb{R}^{N_p}$ is a vector of uncertain model parameters, relies crucially on understanding and quantifying the impact of \mathbf{p} on f . The Sobol' approach can be extended in a quite simple way to address this case.

- **Remark 7.5.1.** This situation might be of interest whenever f is (a function of) the solution of a dynamical system,

$$\begin{cases} \dot{\mathbf{X}}(t) = \Phi(t, \mathbf{X}(t); \mathbf{p}) & t \in (0, T) \\ \mathbf{X}(0) = \mathbf{q} \end{cases}$$

where both $\mathbf{p} \in \mathbb{R}^{N_p}$ and $\mathbf{q} \in \mathbb{N}$ might be random variables; we refer to both the coefficients affecting the physical law expressed by $\Phi(\cdot, \cdot; \mathbf{p}) : \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}$ and the initial condition as parameters.

A possible case is then the one where

$$Y = X_i(t; \mathbf{p}), \quad i = 1, \dots, N$$

or

$$Y = f(\mathbf{X}(t, \mathbf{p}); \mathbf{p}).$$

Most of the literature on global sensitivity analysis considers scalar outputs as opposed to the functional framework corresponding to the dynamical system above. This amounts, for instance, to analyzing the sensitivity of

$$f(t_0, \mathbf{p}) \quad \text{for a fixed } t_0$$

or to the study of integrated quantities such as

$$y(\mathbf{p}) = \int_0^T f(t, \mathbf{p}) dt.$$

While it is possible to apply Sobol's approach pointwise in time, for instance at the nodes of a grid,

$$0 = t_0 < t_1 < \dots < t_{n-1} < t_n = T,$$

this approach presents two shortcomings. First, treating the $f(t_k, \mathbf{p})$'s, $k = 1, \dots, n$, independently of one another ignores the temporal correlation structure of the process. Second, the variance of the process itself varies in time therefore skewing relative importance measurements across time.

As an illustrative example, consider an underdamped mechanical oscillator whose motion is governed by the initial value problem

$$\begin{cases} x'' + 2\alpha x + (\alpha^2 + \beta^2)x = 0, & t \in (0, T) \\ x(0) = l, \quad x'(0) = 0 \end{cases}$$

whose solution is

$$y(t; \alpha, \beta, l) = le^{-\alpha t}(\cos \beta t + \frac{\alpha}{\beta} \sin \beta t)$$

and the corresponding process is given by $f(t, \mathbf{p}) = x(t; \mathbf{p})$ where \mathbf{p} is a random vector that parameterizes the uncertainty in the parameters (α, β, l) . Figure 7.9 (left) shows the time evolution of the mean trajectory (solid line) and the two standard deviation bounds (dashed lines). The values of the traditional pointwise total Sobol' indices are reported in Figure 7.9 (right). These results are difficult to interpret as the balance of sensitivities changes multiple times.

Moreover, this standard approach is entirely unaware of the history of the process and, specifically here, of the asymptotically diminishing variance. For instance, the reported increasing influence of α is largely an artifact of the method.

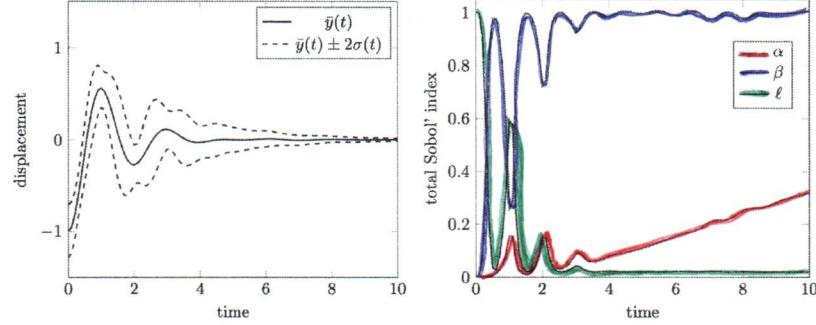


Figure 7.9: Behavior of the mechanical oscillator problem with uncertain parameters $\alpha \sim \mathcal{U}(3/8, 5/8)$, $\beta \sim \mathcal{U}(10/4, 15/4)$ and $\ell \sim \mathcal{U}(-5/4, -3/4)$. Left: mean trajectory and the two standard deviation bounds, obtained via Monte Carlo sampling in the uncertain parameter space. Right: standard Sobol' indices over time.

Pointwise indices in time such as the Sobol' indices S_i and S_{T_i} ignore all time correlations. A possible option is to take the following *generalized indices* for the first Sobol' indices:

$$\mathfrak{S}_i(f; T) = \frac{\int_0^T \text{Var}_{p_i} [\mathbb{E}_{\mathbf{p}_{\sim i}} [f | p_i](t)]}{\int_0^T \text{Var}[f(t)] dt}, \quad i = 1, \dots, N_p$$

and the total sensitivity indices:

$$\mathfrak{S}_{T_i}(f; T) = 1 - \frac{\int_0^T \text{Var}_{\mathbf{p}_{\sim i}} [\mathbb{E}_{p_i} [f | \mathbf{p}_{\sim i}](t)]}{\int_0^T \text{Var}[f(t)]} dt, \quad i = 1, \dots, N_p$$

respectively.

- **Remark 7.5.2.** The integrals in the two expressions above can be computed via a quadrature formula on $[0, T]$, with nodes $\{t_m\}_{m=1}^{N_{quad}}$ and weights $\{w_m\}_{m=1}^{N_{quad}}$.
- **Remark 7.5.3.** If a Monte Carlo (MC) sampling of f is feasible, the partial variances appearing in the expressions for the two indices can be computed using traditional MC-based algorithms for estimating the pointwise Sobol' indices. However, computing the generalized indices via direct Monte Carlo sampling is in general expensive.

To illustrate the concepts introduced so far, we return to the mechanical oscillator example and compute its generalized total Sobol' indices; see Figure 7.10 (left). Figure 7.10 (right) illustrates the evolution of the generalized Sobol' indices over successively larger intervals. These results provide a clear analysis of the relative importance of the input parameters along with a ‘history aware’ description of the evolution of these relative importance measurements. While the pointwise in time Sobol' indices show a significant growing influence of α over time, see again Figure 7.9 (right), the generalized indices stabilize quickly and provide an importance assessment of the variables that is consistent over time.

To speed up the evaluation of the generalized Sobol' indices, we can approximate $f(t, \mathbf{p})$ with a cheap-to-evaluate surrogate model $\tilde{f}(t, \mathbf{p})$: possible options in this context are polynomial chaos (PC) expansions, Gaussian processes or a *spectral approach*.

Indeed, it is possible to compute a (truncated) KL decomposition

$$f(t, \mathbf{p}) \approx f^{KL}(t, \mathbf{p}) := \sum_{j=1}^{N_{KL}} f_j(\mathbf{p}) e_j(t), \quad f_j(\mathbf{p}) = \int_0^T f(t, \mathbf{p}) e_j(t) dt,$$

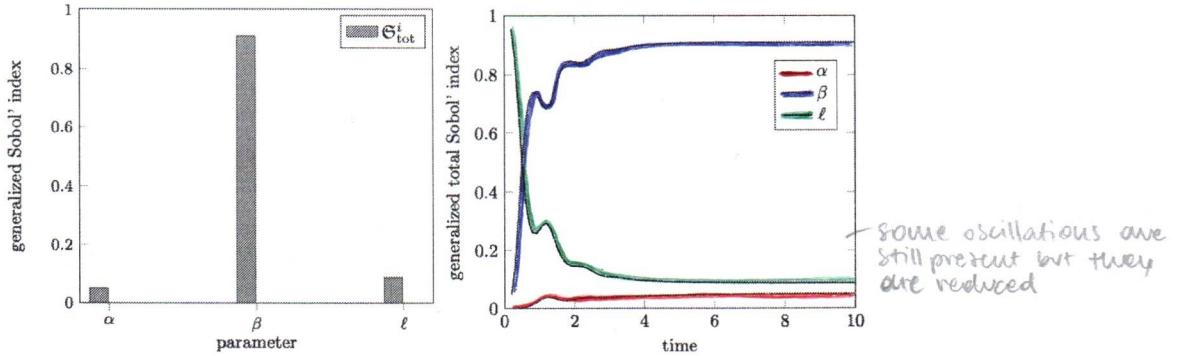


Figure 7.10: Behavior of the mechanical oscillator problem with uncertain parameters $\alpha \sim \mathcal{U}(3/8, 5/8)$, $\beta \sim \mathcal{U}(10/4, 15/4)$ and $l \sim \mathcal{U}(-5/4, -3/4)$. Left: generalized Sobol' indices $\mathfrak{S}_{T_i}(y; T)$ where $T = 10$ and y is the solution to the mechanical oscillator problem (3). Right: $\mathfrak{S}_{T_i}(y; \tau)$ with $\tau \in (0, 10)$.

of the process f , where the mean of the process f_0 is assumed to be zero for simplicity. Here the $f_i(\mathbf{p})$'s are expansion coefficients with variance $\text{Var}[f_i] = \lambda_i$ and where λ_i is an eigenvalue of the covariance operator of f with corresponding eigenvector e_i . For processes with fast decaying eigenvalues λ_i a small truncation level N_{KL} can be used, i.e., the process can be represented by a small number of modes. Recall that

$$\text{Var}[f^{N_{KL}}(t, \mathbf{p})] = \sum_{i=1}^{N_{KL}} \lambda_i e_i(t)^2 \quad \text{and} \int_0^T \text{Var}[f^{N_{KL}}(t, \mathbf{p})] dt = \sum_{i=1}^{N_{KL}} \lambda_i$$

First Sobol' indices can then be approximated as

$$\tilde{\mathfrak{S}}_i(f; T) = \frac{\sum_{i=1}^{N_{KL}} \lambda_i \text{Var}_{p_i} [\mathbb{E}_{\mathbf{p}_{-i}}[\tilde{f}_i(\mathbf{p}) | p_i](t)]}{\sum_{i=1}^{N_{KL}} \lambda_i}, \quad i = 1, \dots, N_p$$

where $f_i(\mathbf{p})$ can be replaced by a surrogate model $\tilde{f}_i(\mathbf{p})$, built e.g. through a Gaussian process or a polynomial chaos expansion.

- density (δ -sensitivity measure)
- CDF (PAWN)

7.6 Moment-Independent importance measures

The identification of key-uncertainty drivers can be also performed using moment-independent sensitivity measures. Being *moment independent*, these sensitivity measures take into account the change in the entire distribution (density) of the model output, instead of the variation of one of its particular moments (e.g., variance).

7.6.1 δ -sensitivity measure (or Borgonovo density-based sensitivity measure)

To characterize the intuition below density-based sensitivity measures, we start with an example. Consider the following input-output mapping:

$$Y = G(X_1, X_2, X_3) = X_1 X_2 + X_3$$

and suppose that the analyst is uncertain about the model inputs. She assigns X_1 uniformly and independently distributed between -1 and 1 , and X_2 and X_3 uniformly distributed between 0 and 1 . Then, what our degree of belief about Y is displayed by the distribution of Y . If Y is a continuous random variable, then we can refer to its (unconditional) density, $f_Y(y)$, which is the blue curve in the figure below.

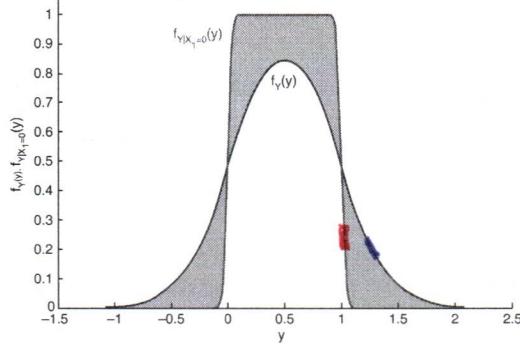


Figure 7.11: Unconditional model output density ($f_Y(y)$, blue) and conditional model output density given that $X_1 = 0$ ($f_{Y|X_1=0}(y)$, red). Taken from [17, Chap. 37].

Consider now to inspect what happens if we are able to fix X_1 , say, to its mean value (which is 0). Then, the input-output mapping becomes

$$G(0, X_2, X_3) = X_3$$

and the conditional model output density given $X_1 = 0$ is a uniform distribution between 0 and 1 . This distribution, $f_{Y|X_1=0}(y)$, is plotted in the red curve of Figure 7.11. Thus, receiving information that $X_1 = 0$ causes a change in the decision-maker's view about Y which is reflected by the change in the corresponding densities.

The shaded region in the figure displays the area enclosed between the two densities. Such area is given, in formulae, by

$$a_1(0) = \int_y |f_Y(y) - f_{Y|X_1=0}(y)| dy$$

where the subscript in $a_1(0)$ refers to the model input (X_1) and (0) reflects the value at which X_1 has been fixed. In our case, we register $a_1(0) = 0.4556$. In general, we shall write

$$a_i(x_i) = \int_y |f_Y(y) - f_{Y|X_i=x_i}(y)| dy$$

where $a_i(x_i)$ is the area enclosed between the conditional and unconditional densities, obtained by fixing the generic model input X_i at x_i . However, because X_i is a continuous random variable, there is a finite probability that X_i assumes some other values in its support. Thus, $a_i(x_i)$ is a conditional separation for any value x_i . We can make the separation unconditional by taking the expectation over all possible values of X_i in accordance with the marginal distribution of X_i . We then write

$$\delta_i = \underbrace{\left(\frac{1}{2} \mathbb{E} \left[\int_y |f_Y(y) - f_{Y|X_i=x_i}(y)| dy \right] \right)}_{\text{(average over all } x_i\text{)}} = \frac{1}{2} \int_{\mathcal{X}_i} f_{X_i}(x_i) dx_i \int_y |f_Y(y) - f_{Y|X_i=x_i}(y)| dy$$

where the factor $\frac{1}{2}$ is inserted for normalization purposes. The sensitivity measure in the previous equation is the expected separation between the conditional and unconditional model output densities provoked by fixing X_i , with the separation measured using the L^1 -norm. Introduced by Borgonovo in [7], it is referred to as δ -sensitivity measure, and fulfills the following properties:

$$\delta_i = \frac{1}{2} \mathbb{E}[a_i(x_i)]$$

(average over all x_i)

($\frac{1}{2}$ for normalization)

1. $\delta_i \in [0, 1]$ and $\delta_i = 0$ if and only if Y is independent of X_i ; a null value of δ_i then implies that the model output is independent of X_i ;
2. $\delta = 0$ is equal to 1 when we resolve uncertainty in all model inputs, that is,

$$\delta_{1,2,\dots,a} = \frac{1}{2} \mathbb{E} \left[\int_Y |f_Y(y) - f_{Y|X=x^*}(y)| dy \right] = 1;$$

3. δ is invariant by transformations.

Note, however, that the computation of the δ -sensitivity measure still suffers from the disadvantage that it requires approximating several PDFs, and expectations.

Another option to set up a moment-independent importance measure is to consider global sensitivity measures based on cumulative distribution functions (CDFs). In the CDF context, one quantifies the discrepancy between $F_Y(y)$ (the unconditional cumulative distribution function of the output Y) and $F_{Y|X_i}(y)$ (the conditional cumulative distribution function when X_i is fixed) using the distance between distributions.

Given a probability metric between distributions, $d(\cdot, \cdot)$, we define the CDF-based importance measure of X_i with respect to Y as

$$\beta_i^d = \mathbb{E} [d(F_Y(y), F_{Y|X_i}(y))],$$

where the expectation is with respect to the marginal distribution of X_i . Several possible choices for the metric $d(\cdot, \cdot)$ exist. The most well-known metrics are

- the *Kolmogorov-Smirnov* (KS) metric, which measures the distance between $F_Y(y)$ and $F_{Y|X_i}(y)$ by

$$d^{KS}(F_Y, F_{Y|X_i}) = \max_y |F_Y(y) - F_{Y|X_i}(y)|$$

thus considering the maximum over all possible values of y of the difference between $F_Y(y)$ and $F_{Y|X_i}(y)$;

- the *Kuiper* metric, which is a correction of the Kolmogorov-Smirnov metric given by:

$$d^{Ku}(F_Y, F_{Y|X_i}) = \max_y \{F_Y(y) - F_{Y|X_i}(y)\} + \max_y \{F_{Y|X_i}(y) - F_Y(y)\}.$$

the rational behind is:
if $F_Y(y) \approx F_{Y|X_i}(y)$
then removing X_i does
not affect the distr.
of $Y \Rightarrow X_i$ has no
influence on Y .
 $(d(F_Y(y), F_{Y|X_i}(y)) \approx 0)$
 $\Rightarrow \beta_i^d \approx 0$)

7.6.2 PAWN sensitivity measure

Using the KS metric, it is possible to introduce another method, called PAWN (from Pianosi and Wagener, [36]). Since $F_{Y|X_i}(y)$ accounts for what happens when the variability due to X_i is removed, its distance from $F_Y(y)$ provides a measure of the effects of X_i on Y . The limit case is when $F_{Y|X_i}(y)$ coincides with $F_Y(y)$ (case (a) in Figure 7.12): it means that removing the uncertainty about X_i does not affect the output distribution, and one can conclude that X_i has no influence on Y . If instead the distance between $F_{Y|X_i}(y)$ and $F_Y(y)$ increases, the influence of X_i increases as well (case (b) in Figure 7.12).

As a measure of distance between unconditional and conditional CDFs, we use the Kolmogorov-Smirnov statistic

$$KS(x_i) = \max_y |F_Y(y) - F_{Y|X_i=x_i}(y)|. \quad (7.13)$$

As KS depends on the value at which we fix X_i , the PAWN index T_i considers a statistic (e.g. the maximum or the median) over all possible conditioning values of x_i , i.e.

$$T_i = stat_{x_i}[KS(x_i)] \quad (7.14)$$

such that

maximum,
median, mean, ..

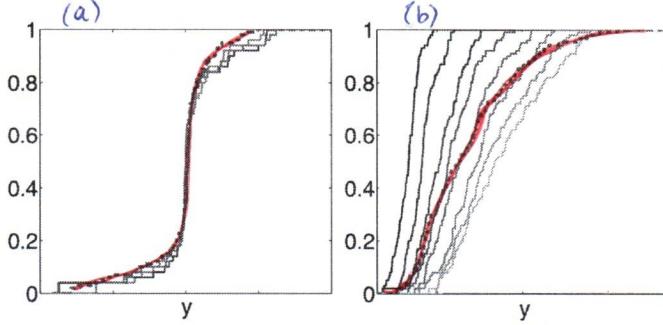


Figure 7.12: Two illustrative examples of CDFs of the model output Y . The red dashed line is the unconditional distribution function $F_Y(y)$ and the grey lines are the conditional distribution function $F_{Y|X_i}(y)$ at different fixed value of input X_i . Taken from [36].

1. T_i varies between 0 and 1;
2. the lower the value of T_i , the less influential X_i ;
3. if $T_i = 0$, then X_i has no influence on Y .

so that the PAWN index T_i can be used for both Factor Prioritisation and Factor Fixing.

If the unconditional and conditional distributions are very similar, it means that variations in the i -th factor do not significantly affect the variability of Y , and therefore, that factor has little influence. Conversely, the larger the difference between the two distributions, the more influential the input factor. This is captured by the maximum in Equation (7.13), which provides a measure of the distance between the two CDFs. The statistic in Equation (7.14) instead is used to remove the variability in the results that might arise from different choices of the nominal value x_i .

- **Remark 7.6.1.** One more property of index T_i is that it can be easily tailored to focus on a particular sub-range of the output distribution, rather than considering the entire range. To this end, it is sufficient that the maximum appearing in Equation (7.13) be taken with respect to Y values in the sub-range of interest.

From a numerical point of view, the Kolmogorov-Smirnov statistic of Equation (7.13) is approximated by

$$\hat{KS}(x_i) = \max_y |\hat{F}_Y(y) - \hat{F}_{Y|X_i=x_i}(y)|. \quad (7.15)$$

where $\hat{F}_Y(\cdot)$ and $\hat{F}_{Y|X_i=x_i}(\cdot)$ are the empirical unconditional and conditional CDFs. The former is approximated using N_u output evaluations obtained by sampling the entire input space; the latter is instead approximated using N_c output evaluations obtained by sampling the non-fixed inputs only, while keeping the value of X_i fixed. Secondly, the statistic with respect to the conditioning value of X_i is replaced by its sample version, i.e. Equation (7.14) is approximated by

$$\hat{T}_i = stat_{x_i=\bar{x}_i^{(1)}, \dots, \bar{x}_i^{(n)}} [\hat{KS}(x_i)] \quad (7.16)$$

where $\bar{x}_i^{(1)}, \dots, \bar{x}_i^{(n)}$ are n randomly sampled values for the fixed input x_i . Hence, the total number of model evaluations necessary to compute the sensitivity indices of Equation (7.16) for all the k inputs is $N_u + nN_c k$.

Remark 7.6.2. As n sets the number of conditioning values sampled from the one-dimensional space of variation of X_i , its value might reasonably be in the order of few dozens, while N_u and N_c , which set the number of samples taken in the k -dimensional and $(k-1)$ -dimensional space of all inputs and all-inputs-but- X_i , should be significantly higher.

If the specific purpose of SA is to determine non-influential inputs (Factor Fixing), the PAWN approach can be used in combination with the two-sample Kolmogorov-Smirnov test. The test rejects the hypothesis that two distributions (e.g. $F_Y(\cdot)$ and $F_{Y|X_i}(\cdot)$) are the same if

$$\hat{KS} > c(\alpha) \sqrt{\frac{N_u + N_c}{N_u N_c}}$$

$$H_0: F_Y = F_{Y|X_i}$$

$$H_1: F_Y \neq F_{Y|X_i}$$

where \hat{KS} is the KS distance between the two empirical CDFs, N_u and N_c are the number of samples used to build the empirical CDFs, α is the confidence level, and the critical value $c(\alpha)$ can be found in the literature.

In our context, rejecting the hypothesis implies that input X_i is influential. In fact, if X_i was non-influential, then the distributions $F_Y(\cdot)$ and $F_{Y|X_i}(\cdot)$ should coincide at all the conditioning values. Because in practice one cannot apply the test at all possible conditioning values, we suggest to use the frequency with which the test is passed over the n conditioning values $\bar{x}_i^{(1)}, \dots, \bar{x}_i^{(n)}$ used to compute the approximate index \hat{T}_i . If the hypothesis that the two CDFs be the same is never rejected, then input i can be considered non-influential.

Density-based methods have a number of advantages:

- they can be applied to any type of input factor, including time series of model forcing inputs or output observations; this is not possible for other GSA methods.
- as Variance-based SA methods, they can be computed regardless of the mathematical properties and meaning of the input factors. However, variance-based methods are not suitable when the output distribution is highly skewed or multimodal and variance would be a poor measure of uncertainty. Density-based methods, instead, are applicable also in those situations because they assess changes in the entire distribution of the output y , rather than in one of these moments only.
- Evaluating indices (as in the PAWN method) that are efficiently approximated by empirical distribution functions only requires a relatively limited number of model evaluations.

Hands-On 8 focuses on the application of the sensitivity analysis techniques described so far to some examples of increasing complexity.

7.6.3 An example: the Ishigami-Homma function (continued) ▶

We now apply the PAWN approach, obtaining the three sets of conditional CDFs in the panels of Figure 7.13. In all panels, the red dashed line is the empirical unconditional output CDF $\hat{F}_Y(\cdot)$ while the grey lines are the conditional CDFs $\hat{F}_{Y|X_i}(\cdot)$ ($i = 1, 2, 3$) obtained at $n = 10$ different conditioning values of X_i . These values can be read on the horizontal axis of the panels of Figure 7.14, which report the (estimated) Kolmogorov-Smirnov statistics. The horizontal line is the value $c(\alpha)\sqrt{(N_u + N_c)/(N_u N_c)}$ (with $\alpha = 0.05$) for the two-sample Kolmogorov-Smirnov test. We consider, as experimental setup: $n = 10$ conditioning points, $N_u = 150$, and $N_c = 100$.

Visual analysis of the CDFs (Figure 7.13) immediately shows that X_1 and X_3 are both much more influential than X_2 , as their conditional CDFs are more widespread around the unconditional one (red line), while those of X_2 almost overlap with the unconditional CDF. In quantitative terms, X_1 is the most influential, as shown by the analysis of the Kolmogorov-Smirnov statistic. In particular, the PAWN sensitivity indices are equal to

$$T_1 = 0.48, \quad T_2 = 0.14, \quad T_3 = 0.3 \quad \text{if stat} = \text{median}$$

$$T_1 = 0.53, \quad T_2 = 0.19, \quad T_3 = 0.35 \quad \text{if stat} = \text{max.}$$

In other words, when used for Factor Prioritisation, PAWN provides the same ranking as variance-based SA.

By applying the two-sample Kolmogorov-Smirnov test for Factor Fixing, one would conclude that X_2 is non-influential (at confidence level $\alpha = 0.05$) because its KS statistics are consistently below the threshold value (see Figure 7.14). This is again consistent with a qualitative judgement that might be formulated based on variance-based SA results where the total effects index of X_2 is very low ($S_{T_2} = 0.0009$).

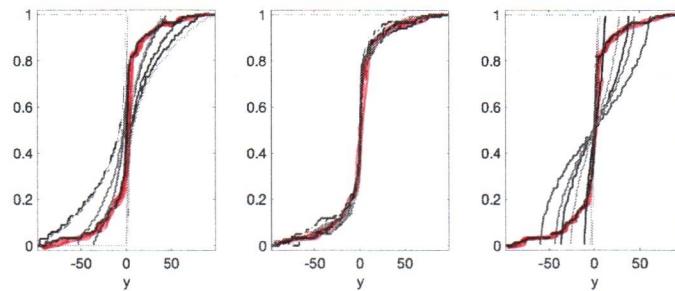


Figure 7.13: Empirical unconditional output distribution \hat{F}_Y (red dashed lines) and conditional ones $\hat{F}_{Y|X_i}$ (grey solid lines).

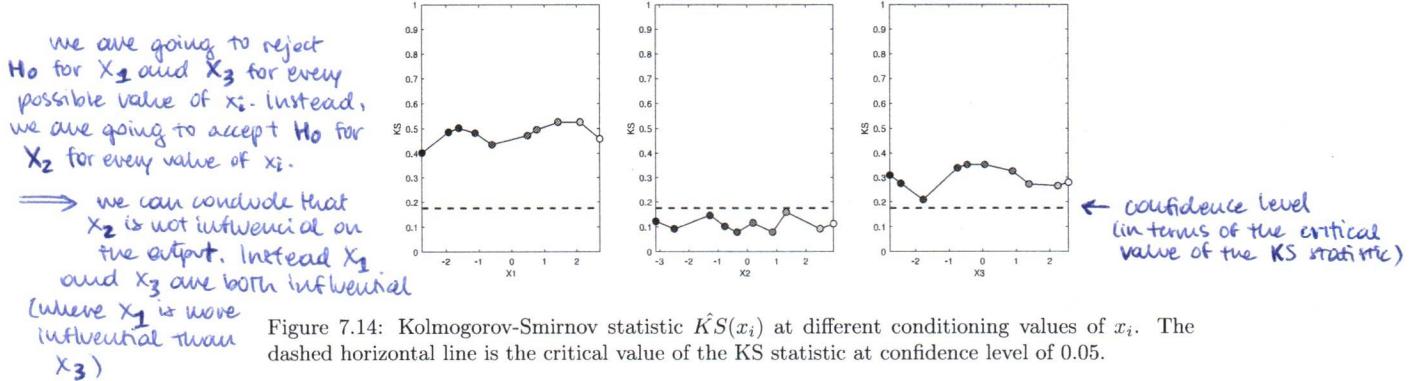
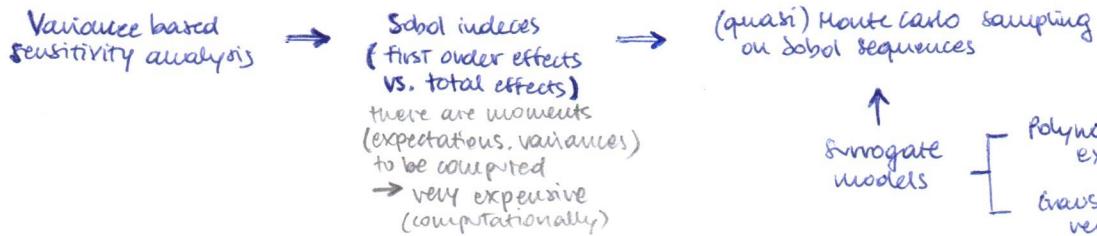


Figure 7.14: Kolmogorov-Smirnov statistic $\hat{KS}(x_i)$ at different conditioning values of x_i . The dashed horizontal line is the critical value of the KS statistic at confidence level of 0.05.

7.7 Surrogate models or emulators

So far, we have seen how global sensitivity analysis is a powerful approach for determining the key random input parameters that drive the uncertainty of model output predictions. Yet the classical computation of the so-called Sobol' indices is based on Monte Carlo simulation, which is not affordable when computationally expensive models are used, as it is the case in most applications in engineering and applied sciences. This is just a possible context where the use of surrogate models or emulators can prove to be necessary for the sake of computational efficiency; even if we do not describe how to take advantage of their use in the following chapters, we must always bear in mind that very often, without suitable emulators, forward or inverse UQ analyses involving complex models would be completely unaffordable. Another possibility when dealing with PDE-based models, which we will briefly address in the following, is to rely on suitable reduced order modeling techniques, aiming at approximating the solution of the underlying differential model, rather than only the input-output map.



For this reason, surrogate models (or metamodels, or *emulators*) such as polynomial chaos expansions (PCE) and Gaussian processes (GP), have received tremendous attention in the last few years, as they allow us to replace the original, taxing model by a surrogate which is built from an experimental design of limited size. Then the surrogate can be used to compute, for instance, the sensitivity indices, in negligible time.

In this section we introduce two possible strategies to set surrogate models, emphasizing their strengths and limitations in the context of global sensitivity analysis. In particular, Sobol' indices can be computed analytically from the PCE coefficients. In contrast, confidence intervals on sensitivity indices can be derived straightforwardly from the properties of GP.

Generally speaking, a surrogate model (or metamodel, or emulator) is an approximation of the original computational model

$$Y = G(\mathbf{X}), \quad \mathbf{X} = (X_1, \dots, X_k) \in \mathbb{R}^k$$

which is constructed relying on a limited number of runs of the true model (sometimes referred to as *experimental design*):

$$\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}.$$

Once a type of surrogate model is selected, the parameters have to be fitted based on the information contained in the experimental design \mathcal{X} and associated runs of the original computational model $\mathcal{Y} = \{y_i = G(\mathbf{x}^{(i)})\}, i = 1, \dots, N\}$.

7.7.1 Polynomial Chaos Expansions

using orthogonal polynomials to approximate input-output relations goes under the name of polynomial chaos

In generalized polynomial chaos (gPC), different kinds of orthogonal polynomials are chosen as a basis depending on the probability distribution of the random inputs. Optimal convergence can be achieved by choosing the proper basis.

The use of polynomial chaos⁴ expansions (PCE) has emerged in uncertainty quantification problems under the form of stochastic finite element methods. As we will see, in this setup the constitutive equations of the physical problem are discretized both in the physical space (using standard finite element techniques) and in the random space using PCE. This results in coupled systems of equations which require ad hoc solvers, thus the term *intrusive approach*. Nonintrusive techniques such as projection or stochastic collocation have emerged more recently as a means to compute the coefficients of PC expansions from repeated evaluations of the existing model G considered as a black-box function. In this section we focus on a particular nonintrusive approach based on a least-square approximation.

Let us consider a computational model $\mathbf{x} \in \mathcal{D}_X \subset \mathbb{R}^d \mapsto y = G(\mathbf{x}) \in \mathbb{R}$. Suppose that the uncertainty in the input parameters is modeled by a random vector \mathbf{X} with prescribed joint probability density function $f_{\mathbf{X}}(\mathbf{x})$. The resulting (random) quantity of interest $Y = G(\mathbf{X})$ is obtained by propagating the uncertainty in \mathbf{X} through G . Assuming that Y has a finite variance, the following spectral representation

$$Y = \sum_{j=0}^{\infty} y_j Z_j$$

holds, where $\{Z_j\}_{j=0}^{\infty}$ is a numerable set of random variables (a basis of the (Hilbert) space in which the problem is set), and $\{y_j\}_{j=0}^{\infty}$ are coefficients. These latter can be interpreted as the coordinates of Y in this basis. When dealing with polynomial chaos expansions, the basis terms

⁴The term *polynomial chaos* was coined by Norbert Wiener in 1938 in his work studying the decomposition of Gaussian stochastic processes – and this was long before the phenomenon of chaos in dynamical systems was known – using Hermite polynomials as an orthogonal basis.

$\{Z_j\}_{j=0}^{\infty}$ are multivariate orthonormal polynomials in the input vector \mathbf{X} , that is, $Z_j = \phi_j(\mathbf{X})$. We will discover another (maybe the most :)) relevant use of families of orthogonal polynomials from numerical analysis...

Let us first address the case $k = 1$.

Definition 7.7.1. Let X be a (continuous) random variable with CDF F_X and PDF f_X , such that the moments $\mathbb{E}[|X|^{2m}]$ are finite for every $m \in \mathbb{N}$. The generalized polynomial chaos (gPC) basis functions $\{\phi_n\}_{n \geq 0}$ are the orthogonal polynomial functions satisfying

$$\mathbb{E}[\phi_m(X)\phi_n(X)] = \int \phi_m(x)\phi_n(x)f_X(x)dx = a_n\delta_{mn}, \quad m, n \in \mathbb{N} \quad (7.17)$$

where

$$a_n = \mathbb{E}[\phi_n^2(X)], \quad n \in \mathbb{N}.$$

The set $\{\phi_n(x)\}$ is a family of orthogonal polynomials of $x \in \mathbb{R}$ with respect to the weight function $f_X(x)$, which is the probability function of the random variable X . This establishes a correspondence between the distribution of the random variable X and the type of orthogonal polynomials of its gPC basis.

- **Example 7.7.1.** Let $X \sim N(0, 1)$, with

$$f_X(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}.$$

The orthogonality (7.17) then defines the Hermite orthogonal polynomials $\{H_n(X)\}$, for which

$$H_0(X) = 1, \quad H_1(X) = X, \quad H_2(X) = X^2 - 1, \quad H_3(X) = X^3 - 3X, \dots$$

This is the classical Wiener-Hermite polynomial chaos basis.

- **Example 7.7.2.** Let $X \sim \mathcal{U}(-1, 1)$ with PDF $f_X(x) = 1/2$, and is a constant. The orthogonality (7.17) then defines the Legendre orthogonal polynomials $\{L_n(X)\}$, for which

$$L_0(X) = 1, \quad L_1(X) = X, \quad L_2(X) = \frac{3}{2}X^2 - \frac{1}{2}, \dots$$

The obtained families are orthogonal, but not orthonormal. An orthonormal family $\{\psi_n\}_{j=0}^{\infty}$ can be obtained as

$$\psi_n(x) = \frac{\phi_n(x)}{\sqrt{a_n}}, \quad n \in \mathbb{N}.$$

In general:

variable	distribution	orthogonal polynomials	Hilbert basis $\psi_k(x)$
Uniform $U(-1, 1)$	$\mathbf{1}_{[-1,1]}(x)/2$	Legendre $L_n(x)$	$P_n(x)/\sqrt{\frac{1}{2n+1}}$
Gaussian $N(0, 1)$	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	Hermite $H_n(x)$	$H_n(x)/\sqrt{n!}$
Gamma $\Gamma(a, 1)$	$x^a e^{-x} \mathbf{1}_{\mathbb{R}^+}(x)$	Laguerre $L_n^a(x)$	$L_n^a(x)/\sqrt{\frac{\Gamma(n+a+1)}{n!}}$

The orthogonality (7.17) ensures that the polynomials can be used as basis functions to approximate functions in terms of the random variable X .

Definition 7.7.2. Let $G(X)$ be a function of a random variable X , whose support is \mathcal{D}_X . A generalized polynomial chaos approximation in a strong sense is $G_N(X) \in \mathbb{P}_N$, where $\mathbb{P}_N(X)$ is the space of polynomials of X of degree up to $N \geq 0$, such that (in a proper norm)

$$\|G(X) - G_N(X)\| \rightarrow 0 \quad \text{as } N \rightarrow \infty.$$

When taking the $L^2_{f_X}(\mathcal{D}_X)$ space, with norm $\|G\|_{L^2_{f_X}(\mathcal{D}_X)} = \sqrt{\mathbb{E}[G^2]}$, we can define the N -th degree gPC orthogonal projection

$$P_N G = \sum_{k=0}^N \hat{G}_k \phi_k(Z), \quad \hat{G}_k = \frac{1}{a_k} \mathbb{E}[G(X) \phi_k(X)]$$

being \hat{G}_k the k -th Fourier coefficient of G . The existence and convergence of the projection follow directly from the classical approximation theory, since

$$\|G - P_N G\|_{L^2_{f_X}} \rightarrow 0, \quad N \rightarrow \infty$$

which is also often referred to as mean-square convergence. When a gPC expansion $G_N(X)$ of a function $G(X)$ converges to $G(X)$ in a strong norm, such as the mean-square norm, it implies that $G_N(X)$ converges to $G(X)$ in probability, which further implies the convergence in distribution.

Remark 7.7.1. Recall that, by Parseval inequality,

$$\min_{H \in \mathbb{P}_N} \|G - H\|_{L^2_{f_X}}^2 = \|G - P_N G\|_{L^2_{f_X}}^2 = \sum_{p=N+1}^{\infty} \hat{G}_p^2.$$

Truncating the gPC expansion up to degree N provides a finite-dimensional representation of any function $G(X)$ of the random input X . However, computing the N -th degree gPC orthogonal projection is an intrusive task, requiring the evaluation of the integrals defining \hat{G}_k , $k = 0, \dots, N$, and the knowledge of the explicit form of G in terms of Z .

Let us now consider the case $k > 1$. When more than one independent random variables are involved, multivariate gPC expansions are required. Let $\mathbf{X} = (X_1, \dots, X_k)$ be a random vector with mutually independent components, so that the PDF $f_{\mathbf{X}}(\mathbf{x})$ can be expressed as

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{d=\mathbb{k}} f_{X_i}(x_i).$$

Also, let $\{\phi_n(X_i)\}_{n=0}^{\infty}$ be the univariate gPC basis functions in X_i , that is,

$$(\phi_n, \phi_m)_i := \mathbb{E}[\phi_n(X_i) \phi_m(X_i)] = \int_{\mathcal{D}_{X_i}} \phi_n(x) \phi_m(x) f_{X_i}(x) dx = \delta_{mn} a_n^{(i)}, \quad m, n \in \mathbb{N}$$

where δ_{mn} is the Kronecker operator, and

$$a_n^{(i)} := \|\phi_n^{(i)}\|_i^2 = (\phi_n^{(i)}, \phi_n^{(i)})_i.$$

Also in this case, the obtained families are orthogonal, but not orthonormal. An orthonormal family $\{\psi_n^{(i)}\}_{j=0}^{\infty}$ can be obtained as

$$\psi_n^{(i)}(x_i) = \frac{\phi_n^{(i)}(x_i)}{\sqrt{a_n^{(i)}}}, \quad i = 1, \dots, k, \quad n \in \mathbb{N}.$$

e.g. $\vec{X} = (X_1, X_2) \Rightarrow \begin{cases} X_1 \rightarrow \{\psi_n^{(1)}\}_n \\ X_2 \rightarrow \{\psi_n^{(2)}\}_n \end{cases}$

we're gonna need a coefficient for every possible combination of the basis $\psi_j^{(1)}, \psi_i^{(2)}$ $i, j \in \mathbb{N}$

\rightarrow

(126)

Chapter 7. Sensitivity Analysis

From the sets of univariate orthonormal polynomials, we can build multivariate orthonormal polynomials with a tensor product construction. For this purpose, we can define the multi-indices $\alpha \in \mathbb{N}^k$, which are ordered lists of integers, $\alpha = (\alpha_1, \dots, \alpha_k)$, $\alpha_i \in \mathbb{N}$, with $|\alpha| = \alpha_1 + \dots + \alpha_k$. Then, we can associate a multivariate polynomial Ψ_α to any multi-index α by:

$$\Psi_\alpha(\mathbf{x}) = \prod_{i=1}^k \psi_{\alpha_i}^{(i)}(x_i)$$

where the univariate polynomials $\{\psi_n^{(i)}, n \in \mathbb{N}\}$ are the ones defined above. In other words, when $0 \leq |\alpha| \leq N$, the k -variate N -th degree gPC basis functions are the products of the univariate gPC polynomials of total degree less than or equal to N .

The set of all multivariate polynomials in the input random vector \mathbf{X} forms a basis of the Hilbert space in which $Y = G(\mathbf{X})$ can be represented:

$$Y = \sum_{\alpha \in \mathbb{N}^k} y_\alpha \Psi_\alpha(\mathbf{X}).$$

Remark 7.7.2. In practical sensitivity analysis problems, the input variables may not necessarily have standardized distributions, whence the need of transforming them. For instance, when dealing with independent uniform distributions with support $\mathcal{D}_{X_i} = [a_i, b_i]$, $i = 1, \dots, k$, we can take

$$X_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} U_i, \quad U_i \sim \mathcal{U}(-1, 1).$$

In the case of Gaussian independent variables $X_i \sim N(\mu_i, \sigma_i^2)$,

$$X_i = \mu_i + \sigma_i U_i, \quad U_i \sim N(0, 1).$$

In the general case, when the input variables are non-Gaussian but follow a known distribution, the one- to-one mapping may be obtained as follows:

$$X_i = F_{X_i}^{-1}(\Phi(U_i)), \quad U_i \sim N(0, 1)$$

where F_{X_i} is the CDF of X_i and Φ is the CDF of a standard Gaussian random variable.

This transform approach also allows one to address problems with dependent variables. For instance, if the input vector \mathbf{X} is defined by a set of marginal distributions and a Gaussian copula, it can be transformed into a set of independent standard normal variables; see Section 2.5.3.

The representation of the random response as

$$Y = \sum_{n=1}^{\infty} y_n \Psi_n(X)$$

if $k = 1$, or

$$Y = \sum_{\alpha \in \mathbb{N}^k} y_\alpha \Psi_\alpha(\mathbf{X}).$$

if $k > 1$, is exact when the infinite series is considered. However, in practice, only a finite number of terms may be computed. For this purpose, a truncation scheme has to be selected. Since the polynomial chaos basis consists of multivariate polynomials, it is natural to consider a truncated series all the polynomials up to a given maximum degree. The standard truncation scheme consists in selecting all polynomials such that

$$|\alpha| = \sum_{i=1}^k \alpha_i \leq p, \quad \text{for a given } p.$$

This leads to a set of polynomials \mathbb{P}_p^k

$$\mathbb{P}_p^k = \{p : \mathcal{D}_X \rightarrow \mathbb{R} : p(\mathbf{X}) = \sum_{|\alpha| \leq p} y_\alpha \Psi_\alpha(\mathbf{X})\},$$

associated to the truncation scheme $\mathcal{A}^{k,p} = \{\alpha \in \mathbb{N}^k : |\alpha| \leq p\}$, of cardinality

$$\text{card } \mathcal{A}^{k,p} = \binom{k+p}{p} = \frac{(k+p)!}{k!p!}.$$

The maximal polynomial degree p may typically be equal to 3 - 5 in practical applications. Note that the number of terms in the truncated sum (and the number of coefficients to be computed) increases exponentially with k and p , and then increases dramatically when k is large, say $k > 10$ (curse of dimensionality). This issue may be solved using specific algorithms to compute sparse PCE, a matter that goes beyond the scope of this course.

Computation of the coefficients

The computation of the PCE coefficients may be cast as a least-square minimization problem (or *regression* problem) as follows: once a truncation scheme $\mathcal{A} \subset \mathbb{N}^k$ is chosen (for instance, $\mathcal{A} = \mathcal{A}^{k,p}$) the infinite series is recast as the sum of the truncated series and a residual:

$$Y = G(\mathbf{X}) = \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X}) + \varepsilon,$$

in which ε corresponds to all those PC polynomials whose index α is not in the truncation set \mathcal{A} .

The least-square minimization approach consists in finding the set of coefficients $\mathbf{y} = \{y_\alpha, \alpha \in \mathcal{A}\}$ which minimizes the mean-square error:

$$\mathbb{E} [\varepsilon^2] := \mathbb{E} \left[\left(G(\mathbf{X}) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X}) \right)^2 \right].$$

The set of coefficients \mathbf{y} is computed at once by solving

$$\mathbf{y} = \arg \min_{\mathbf{y} \in \mathbb{R}^{\text{card}(\mathcal{A})}} \mathbb{E} \left[\left(G(\mathbf{X}) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X}) \right)^2 \right].$$

In practice, the discretized version of the problem is obtained by replacing the expectation operator in the equation above by the empirical mean over a sample set:

Find

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y} \in \mathbb{R}^{\text{card}(\mathcal{A})}} \frac{1}{N} \sum_{i=1}^N \left(G(\mathbf{x}^{(i)}) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{x}^{(i)}) \right)^2,$$

where $\mathcal{X} = \{\mathbf{x}^{(i)}, i = 1, \dots, N\}$ is a sample set of points called experimental design (ED) that is typically obtained by Monte Carlo simulation of the input random vector \mathbf{X} .

To solve this least-square minimization problem, the computational model G is first run for each point in the ED, and the results are stored in a vector $\mathcal{Y} = \{y^{(1)}, \dots, y^{(N)}\}$, with $y^{(i)} = G(\mathbf{x}^{(i)})$, $i = 1, \dots, N$.

Then, the solution is obtained by solving the following system of normal equations,

$$(A^T A) \hat{\mathbf{y}} = A^T \mathcal{Y}$$

where the information matrix A is calculated from the evaluation of the basis polynomials onto each point in the ED, that is,

$$(A)_{ij} = \Psi_j(\mathbf{x}^{(i)}), \quad i = 1, \dots, N, \quad j = 1, \dots, \text{card}(\mathcal{A}).$$

The points used in the experimental design may be obtained from crude Monte Carlo simulation. However other types of designs are of common use, especially Latin hypercube sampling (LHS), or quasi-random sequences such as the Sobol' sequence. The size of the experimental design is of crucial importance: it must be larger than the number of unknowns $\text{card}(\mathcal{A})$ for the problem to be well posed. However, in practice most of the problems lead eventually to sparse expansions, i.e., PCE in which most of the coefficients are zero or negligible.

Sobol' indices ad PC expansions

The truncated PC expansion

$$\hat{Y} = G^{PC}(\mathbf{X}) = \sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha \Psi_\alpha(\mathbf{X})$$

contains all the information about the statistical properties of the random output $Y = G(\mathbf{X})$. Due to the orthogonality of the PC basis, mean and standard deviation of \hat{Y} may be computed directly from the coefficients \hat{y} . Indeed, since $\Psi_0 = 1$ and the functions $\{\Psi_\alpha\}_{\alpha}$ are orthogonal, we get

$$\mathbb{E}[\Psi_\alpha(\mathbf{X})] = 0 \quad \forall \alpha \neq 0.$$

Thus, the mean value of \hat{Y} is the first term of the series:

$$\mathbb{E}[\hat{Y}] = \mathbb{E}\left[\sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha \Psi_\alpha(\mathbf{X})\right] = \hat{y}_0.$$

Similarly, the variance of \hat{Y} may be cast as

$$\sigma_{\hat{Y}}^2 = \text{Var}[\hat{Y}] = \mathbb{E}[(\hat{Y} - \hat{y}_0)^2] = \sum_{\alpha \in \mathcal{A}} \hat{y}_\alpha^2 - \hat{y}_0^2.$$

In other words, the mean and variance of the random response may be obtained by a mere combination of the PCE coefficients once the latter have been computed.

Taking advantage of similar properties, it is possible to derive analytic expressions for Sobol' indices based on a PC expansion. The key observation is the following one:

Both polynomial chaos expansions and Sobol' decomposition (in the HDMR format) are sums of orthogonal functions. Taking advantage of this property, it is possible to derive analytic expressions for Sobol' indices based on a PC expansion.

Recall that the Sobol' index attached to each subset of variables $\Omega = \{i_1, \dots, i_s\} \subset \{1, \dots, k\}$ is given by

$$S_\Omega = \frac{V_\Omega}{\text{Var}[Y]} = \frac{\text{Var}[G_\Omega(\mathbf{X}_\Omega)]}{\text{Var}[Y]} = \frac{\mathbb{E}[G_\Omega^2(\mathbf{X}_\Omega)]}{\text{Var}[Y]}.$$

By setting $\Omega = \{i\}$ we obtain the first-order Sobol' indices; total effect Sobol' indices can be computed by the sum of the Sobol' indices of any order that contain X_i , that is,

$$S_i^{tot} = \sum_{\Omega \ni i} S_\Omega.$$

If we now consider the set of multivariate polynomials Ψ_{α} which depend *only* on a subset of variables $\Omega = \{i_1, \dots, i_s\} \subset \{1, \dots, k\}$:

$$\mathcal{A}_{\Omega} = \{\alpha \in \mathcal{A} : \alpha_k \neq 0 \text{ if and only if } k \in \Omega\}.$$

The union of all these sets is by construction equal to Ω . Thus we can reorder the terms of the truncated PC expansion so as to exhibit the Sobol' decomposition:

$$G^{PC}(\mathbf{x}) = y_0 + \sum_{\Omega \subset \{1, \dots, k\}} G_{\Omega}^{PC}(\mathbf{x}_{\Omega}), \quad G_{\Omega}^{PC}(\mathbf{x}_{\Omega}) = \sum_{\alpha \in \Omega} y_{\alpha} \Psi_{\alpha}(\mathbf{x}).$$

Due to the *orthogonality* of the PC basis, the partial variance V_{Ω} then reduces to

$$V_{\Omega} = \text{Var}[G_{\Omega}^{PC}(\mathbf{x}_{\Omega})] = \sum_{\alpha \in \mathcal{A}_{\Omega}} y_{\alpha}^2.$$

In other words, from a given PC expansion, the Sobol' indices at any order may be obtained by a mere combination of the squares of the coefficients. More specifically (see, e.g., [17, Chapter 38])

The PC-based estimator of the first-order Sobol' indices reads:

$$\hat{S}_i = \frac{\sum_{\alpha \in \mathcal{A}_i} \hat{y}_{\alpha}^2}{\sum_{\alpha \in \mathcal{A}} \hat{y}_{\alpha}^2} \quad \text{where} \quad \mathcal{A}_i = \{\alpha \in \mathcal{A} : \alpha_i > 0, \alpha_j \neq 0\}.$$

The total PC-based Sobol' indices read instead as

$$\hat{S}_i^{tot} = \frac{\sum_{\alpha \in \mathcal{A}_i^{tot}} \hat{y}_{\alpha}^2}{\sum_{\alpha \in \mathcal{A}} \hat{y}_{\alpha}^2} \quad \text{where} \quad \mathcal{A}_i^{tot} = \{\alpha \in \mathcal{A} : \alpha_i > 0\}.$$

- **Example 7.7.3.** We want to compare the Sobol' indices computed with the procedure based on MC sampling (see Section 7.5.2) with the ones obtained replacing the model with a gPC surrogate, involving terms up to order 5. The model we consider is the so-called Borehole model,

$$Y = \frac{2\pi T_u (H_u - H_l)}{\log(r/r_w) \cdot \left(1 + \frac{2LT_u}{\log(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}\right)}$$

which returns the value of the water flow Y through a borehole, described by the following $k = 8$ input parameters:

- $r_w \sim N(0.10, 0.0161812^2)$ - radius of borehole (m)
- r such that $\log r \sim N(7.71, 1.0056^2)$ - radius of influence (m)
- $T_u \sim \mathcal{U}(63070, 115600)$ - transmissivity of upper aquifer (m^2/yr)
- $H_u \sim \mathcal{U}(990, 1110)$ - potentiometric head of upper aquifer (m)
- $T_l \sim \mathcal{U}(63.1, 116)$ - transmissivity of lower aquifer (m^2/yr)
- $H_l \sim \mathcal{U}(700, 820)$ - potentiometric head of lower aquifer (m)
- $L \sim \mathcal{U}(1120, 1680)$ - length of borehole (m)
- $K_w \sim \mathcal{U}(9855, 12045)$ - hydraulic conductivity of borehole (m/yr)

Its simplicity and quick evaluation makes it a commonly used function for testing a wide variety of methods in computer experiments.

Here gPC coefficients are computed using the least-squares method relying on $N = 200$ model evaluation. Note instead that, taking $N = 10^5$ as sample size for the MC simulation, the total cost of computation is $(k + 2) \times N = (8 + 2) \times 10^5 = 10^6$ evaluations of the full computational model. Results are reported in Fig. 7.15 and are clearly very close to each other, showing the reliability of the use of gPC expansions in sensitivity analysis. Results have been obtained with the Matlab library Uqlab.

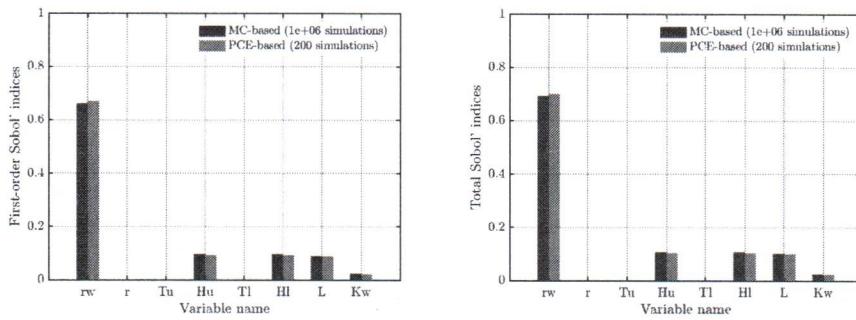


Figure 7.15: First order (left) and total (right) Sobol' indices for the Borehole model computed through MC sampling and using a gPC surrogate.

7.7.2 Gaussian Process Regression

Another possible, frequently used option to build surrogate models is to rely on Gaussian process regression, also known as *kriging*. Gaussian processes have been introduced in Section 2.6, which we refer to for the general setting and notation. The basic recipe is extremely simple, relying on conditional multivariate Gaussian distributions (see Section 2.5.5).

The principle of Gaussian process regression is to consider that the prior knowledge about the computational model $G(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^k$, can be modeled by a Gaussian process $Z(\mathbf{x})$ with a mean $\mu_Z(\mathbf{x}) = E[Z(\mathbf{x})]$ and a covariance function (or kernel) $C_Z(\mathbf{x}, \mathbf{x}')$.

Roughly speaking, we consider that the true response is a realization of $Z(\mathbf{x})$. Starting from this prior distribution, the goal is to determine the predictive distribution (for any possible \mathbf{x}) given a set of model responses $\mathcal{Y} = G(\mathcal{X})$ obtained for a given set of training points $\mathcal{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})$. We first consider the case of a single input variable ($k = 1$) and a scalar output, then we extend our discussion to the more general case of multiple inputs ($k > 1$).

Single input - single output

Let us first consider the case $k = 1$ of a single input and of a scalar output. In this case a GP is fully specified by its mean function $\mu_Z(x)$ and covariance function⁵ $C_Z(x, x')$, and results in a

⁵Recall that the covariance (or kernel) function is usually parametrized in terms of hyper-parameters; one of the most common choices is the squared-exponential kernel,

$$k(x_j, x_l) = \sigma^2 \exp\left(-\frac{|x_j - x_l|^2}{2\theta^2}\right)$$

distribution over functions; we can denote $Z \sim \mathcal{GP}(\mu_Z, C_Z)$. As a result, a realization of a GP at locations x_1, \dots, x_n will be a vector $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where

$$\mu_j = \mu_Z(x_j), \quad \Sigma_{jl} = C_Z(x_j, x_l), \quad j, l = 1, \dots, n.$$

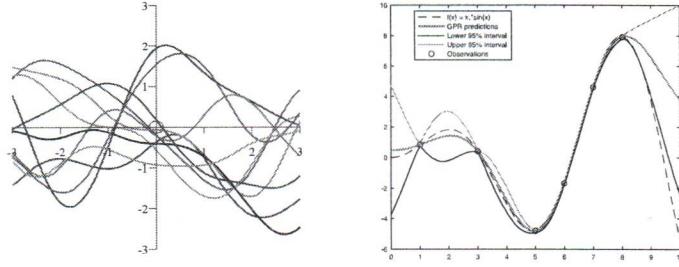


Figure 7.16: Left: samples from a Gaussian process. Right: Gaussian process regression (data coming from the function $x \sin(x)$, prediction, 95% confidence band).

Assume now that the prior knowledge about the computational model $G(x)$, $x \in \mathbb{R}$, can be modeled by a Gaussian process $Z(x)$ with mean $\mu_Z(x)$ and covariance kernel $C_Z(x, x')$.

Let us now denote by $\mathcal{X} = (x^{(1)}, \dots, x^{(n)}) \in \mathbb{R}^n$ a set of *training samples* (or *experimental design*) and, correspondingly, by $\mathcal{Y} = G(\mathcal{X})$ the corresponding model responses, that is, $\mathcal{Y} = (y^{(1)}, \dots, y^{(N)}) \in \mathbb{R}^N$ where

$$y^{(j)} = G(x^{(j)}), \quad j = 1, \dots, N.$$

Our goal is now to perform out-of-sample prediction, that is, to determine $Z(\mathcal{X}_{new})$ at a new set of locations (or *test samples*) $\mathcal{X}_{new} = (x_{new}^{(1)}, \dots, x_{new}^{(m)}) \in \mathbb{R}^m$, given that $Z(\mathcal{X}) = \mathcal{Y}$, updating the prior distribution $Z \sim \mathcal{GP}(\boldsymbol{\mu}_Z, \boldsymbol{C}_Z)$ by incorporating the knowledge provided by the fact that $Z(\mathcal{X}) = \mathcal{Y}$ – that is, the knowledge of the model evaluations on the training points. The resulting *posterior* distribution is then used to make predictions for (unseen) test samples $x_{new}^{(1)}, \dots, x_{new}^{(m)}$. Updating the prior in the light of the training data hinges upon the basic formulas for conditioning a joint Gaussian distribution, see Section 2.5.5.

Indeed, let us denote by $\mathcal{X}_{new} \in \mathbb{R}^m$ a set of test samples and, correspondingly, by $Z(\mathcal{X}_{new}) = (Z(x_{new}^{(1)}), \dots, Z(x_{new}^{(m)})) \in \mathbb{R}^m$ the set of predictions on the test samples. The joint distribution of the training data and the GP predictions $(\mathcal{Y}, Z(\mathcal{X}_{new}))$, according to the prior, is then expressed as

$$\begin{pmatrix} \mathcal{Y} \\ Z(\mathcal{X}_{new}) \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \mu_Z(\mathcal{X}) \\ \mu_Z(\mathcal{X}_{new}) \end{pmatrix}, \begin{pmatrix} C_Z(\mathcal{X}, \mathcal{X}) & C_Z(\mathcal{X}, \mathcal{X}_{new}) \\ C_Z(\mathcal{X}_{new}, \mathcal{X}) & C_Z(\mathcal{X}_{new}, \mathcal{X}_{new}) \end{pmatrix} \right)$$

If there are n training points and m test points, $C_Z(\mathcal{X}, \mathcal{X}_{new})$ denotes the $n \times m$ matrix of the covariances evaluated at all pairs of training and test points, and similarly for the other entries $C_Z(\mathcal{X}, \mathcal{X}_{new})$ and $C_Z(\mathcal{X}_{new}, \mathcal{X}_{new})$.

for which the closer are the two points x_j, x_l , the higher their correlation.

The conditional distribution of $[Z(\mathcal{X}_{new})|Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \theta]$, also referred to as *predictive distribution* of $G(\mathcal{X}_{new})$, at any test sample \mathcal{X}_{new} , is then

$$Z(\mathcal{X}_{new}) | Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \theta \sim \mathcal{N}(\mu_{Z|\mathcal{Y}}(\mathcal{X}_{new}), \Sigma_{Z|\mathcal{Y}}(\mathcal{X}_{new}, \mathcal{X}_{new}))$$

where

$$\mu_{Z|\mathcal{Y}}(\mathcal{X}_{new}) = \mu_Z(\mathcal{X}_{new}) + k(\mathcal{X}_{new}, \mathcal{X})(C_Z(\mathcal{X}, \mathcal{X}))^{-1}(\mathcal{Y} - \mu_Z(\mathcal{X})) \quad (7.18)$$

and

$$\Sigma_{Z|\mathcal{Y}}(\mathcal{X}_{new}, \mathcal{X}_{new}) = C_Z(\mathcal{X}_{new}, \mathcal{X}_{new}) - C_Z(\mathcal{X}_{new}, \mathcal{X})(C_Z(\mathcal{X}, \mathcal{X}))^{-1}C_Z(\mathcal{X}_{new}, \mathcal{X})^\top. \quad (7.19)$$

this is the best part
because we do not only
have a prediction on
new points, for every
new point we have the
distribution of the
outputs (outcomes)

The mean prediction $\mu_{Z|\mathcal{Y}}$ is expressed as a linear combination of observations \mathcal{Y} , plus the prior mean (that is, it is a *linear predictor*); the covariance matrix prediction $\Sigma_{Z|\mathcal{Y}}$ does not depend on the observed targets, but only on the inputs. The GP regression metamodel is then provided by the function values $Z(\mathcal{X}_{new})$ (corresponding to the test sample \mathcal{X}_{new}) sampled from the posterior distribution, by evaluating the mean vector and covariance matrix from (7.18)-(7.19).

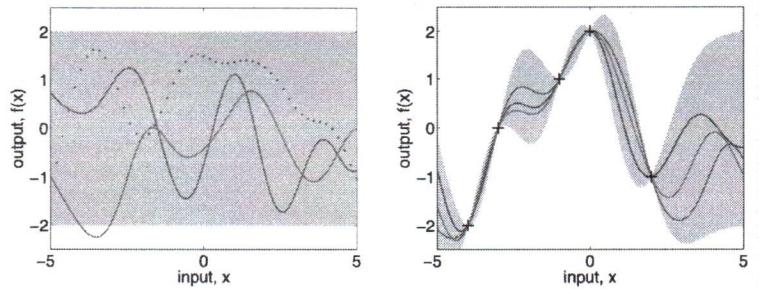


Figure 7.17: GP regression. Left: realizations of the prior GP. Right: realizations of the posterior GP. For any given \bar{x} , different realizations yield a sample from a Gaussian distribution with mean $m_n(\bar{x})$ and variance $k_n(\bar{x}, \bar{x})$.

- **Remark 7.7.3.** In the case model responses are affected by (experimental) noise, so that $\mathcal{Y}_{obs} = \mathcal{Y} + \varepsilon = G(\mathcal{X}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_p^2)$ is additive, independent identically distributed Gaussian noise. In this case, the covariance matrix $C_Z(\mathcal{X}, \mathcal{X})$ has to be replaced by $C_Z(\mathcal{X}, \mathcal{X}) + \sigma_p^2 I$.

7.7.3 Multiple inputs - single output

In the case of multiple inputs, let us consider that the prior knowledge about the computational model $G(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^k$ can be modeled by a Gaussian process $Z(\mathbf{x})$ with mean $\mu_Z(\mathbf{x})$ and a covariance kernel denoted by $C_Z(\mathbf{x}, \mathbf{x}')$. Usually, the mean and the covariance are parametrized as follows:

$$\mu_Z(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta}, \quad C_Z(\mathbf{x}, \mathbf{x}') = \sigma^2 r(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}),$$

where $\mathbf{f}(\mathbf{x})^\top$ is a vector of p prescribed functions, and $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$ have to be estimated. The mean function $\mu_Z(\mathbf{x})$ describes the trend and the covariance kernel $C_Z(\mathbf{x}, \mathbf{x}')$ describes the regularity and characteristic length scale of the model.

Considering an experimental design $\mathcal{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})$, $\mathbf{x}^{(i)} \in \mathbb{R}^k$, and the corresponding model responses $\mathcal{Y} = G(\mathcal{X})$. In this case the predictive distribution of $G(\mathbf{x}_{new})$, at a *single, test* point $\mathbf{x}_{new} \in \mathbb{R}^k$ in the input space, is given by

$$[Z(\mathbf{x}_{new})|Z(\mathcal{X}) = \mathcal{Y}, \sigma^2, \boldsymbol{\theta}] \sim \mathcal{N}(\mu_{Z|\mathcal{Y}}(\mathbf{x}_{new}), C_{Z|\mathcal{Y}}(\mathbf{x}_{new}, \mathbf{x}_{new}))$$

where

$$\mu_{Z|\mathcal{Y}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x})^\top \mathbf{R}^{-1}(\mathcal{Y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \quad (7.20)$$

and

$$C_{Z|\mathcal{Y}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 - (\mathbf{f}^\top(\mathbf{x}) \mathbf{r}^\top(\mathbf{x})) \begin{pmatrix} 0 & \mathbf{F}^\top \\ \mathbf{F} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \mathbf{f}(\mathbf{x}') \\ \mathbf{r}(\mathbf{x}') \end{pmatrix} \right), \quad (7.21)$$

being

$$(\mathbf{R})_{ij} = r(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta}), \quad i, j = 1, \dots, N,$$

$$(\mathbf{r}(\mathbf{x}))_i = r(\mathbf{x}, \mathbf{x}^{(i)}; \boldsymbol{\theta}), \quad i = 1, \dots, N,$$

$$(\mathbf{F})_i = \mathbf{f}(\mathbf{x}^{(i)})^\top, \quad i = 1, \dots, N$$

and $\hat{\boldsymbol{\beta}}$ is such that

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^\top \mathbf{R}^{-1} \mathcal{Y}.$$

This latter expression is obtained by maximum likelihood estimation⁶. The hyper-parameters σ^2 and $\boldsymbol{\theta}$ are not known in practice and shall be estimated with the maximum likelihood method or a cross-validation strategy.

Hence, to build up a GP model, the user has to make several choices. Indeed, the vector of functions $\mathbf{f}(\mathbf{x})$ and the class of the correlation kernel $r(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ need to be set. These choices and the relevance of the model are tested a posteriori with a validation procedure. If the number n of observations is large, an external validation may be performed on a test set. Otherwise, a cross-validation procedure may be used.

- **Example 7.7.4.** We consider a simple analytical function defined by:

$$Y = x \sin(x), \quad x \in [0, 15],$$

as computational model, and a single uniform random variable as probabilistic input model:

$$X \sim \mathcal{U}(0, 15).$$

We generate an experimental design \mathcal{X} of size $N = 8$ using latin hypercube sampling; then, we build a surrogate model using GP regression choosing $\mu_Z(\mathbf{x}) = 0$ and three different covariance functions:

- the Matérn $\nu = 5/2$ correlation function (see Section 3.2.3)

$$r(x, y; \theta, \nu = 5/2) = \left(1 + \sqrt{5} \frac{|x - y|}{\theta} + \frac{5}{3} \left(\frac{|x - y|}{\theta} \right)^2 \right) \exp \left(-\sqrt{5} \frac{|x - y|}{\theta} \right)$$

where θ is the correlation function scale parameter, Γ is the Euler's Gamma function and K_ν is the modified Bessel function of the second kind;

- the linear correlation function

$$r(x, y; \theta) = \max \left(0, 1 - \frac{|x - y|}{\theta} \right);$$

⁶Maximum likelihood estimation involves finding those parameters that maximize the likelihood function. The likelihood function describes the probability of observing the training data for a particular set of parameters. Under the Gaussian process framework, the likelihood function is based on the multivariate normal distribution. For computational reasons, the problem is typically formulated as a minimization of the negative log of the likelihood function. Stay tuned...

- the exponential correlation function

$$r(x, y; \theta) = \exp\left(-\frac{|x - y|}{\theta}\right).$$

Results have been obtained with the Matlab library UQlab and are reported in Figure 7.18.

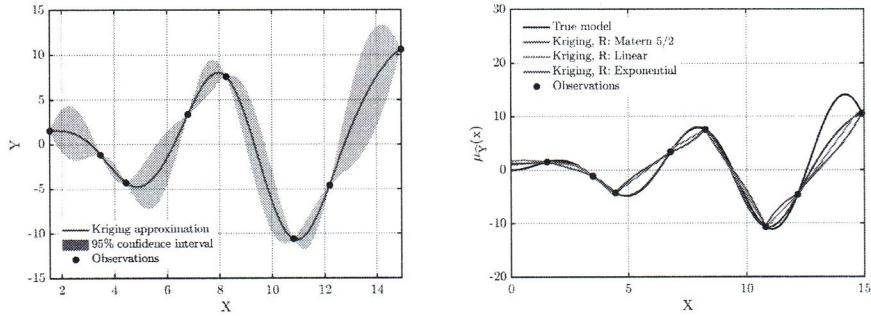


Figure 7.18: Left: GP regression of the function $Y = x \sin(x)$ based on a LHS design \mathcal{X} of size $N = 8$. Right: comparison between the models obtained with different covariance functions.

Then, we compare the results obtained with a Matérn $\nu = 3/2$ correlation function,

$$R(x, y; \theta, \nu = 3/2) = \left(1 + \sqrt{3} \frac{|x - y|}{\theta}\right) \exp\left(-\sqrt{3} \frac{|x - y|}{\theta}\right)$$

and three different trend functions $\mu_Z(x)$: a constant (yielding the so-called ordinary kriging), a third order polynomial, and the function

$$f(x) = x^2 + \sqrt{|x|}$$

Results are reported in Figure 7.19.

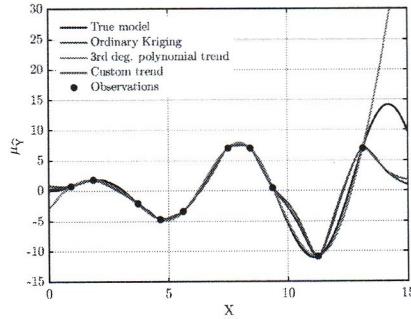


Figure 7.19: GP regression of the function $Y = x \sin(x)$ based on a LHS design \mathcal{X} of size $N = 10$. Comparison between the models obtained with different mean functions (or trends) $\mu_Z(x)$.

In general, possible choices for the GP mean (or trend function) are:

- simple Kriging

$$\mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} = \sum_{j=1}^P f_j(\mathbf{x})$$

where f_j 's are arbitrary but fully specified functions. Note that no estimation on $\boldsymbol{\beta}$ is taken place and the coefficients $\boldsymbol{\beta}$ are all 1's;

- ordinary Kriging

$$\mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} = \beta_0 f_0(\mathbf{x}) = \beta_0$$

that is, the trend has a constant yet unknown value; by convention $f_0(\mathbf{x}) = 1$.

- universal kriging, the most general and flexible formulation, which assumes that the trend is a linear combination of prescribed arbitrary functions (e.g., polynomials)

$$\mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} = \sum_{j=1}^P \beta_j f_j(\mathbf{x}).$$

7.7.4 Sobol' indices and GP regression

Once a GP regression emulator has been constructed, to perform a sensitivity analysis from such a GP model, two approaches are possible. The first one consists in substituting the true model $G(\mathbf{x})$ with the mean of the conditional Gaussian process $\mu_{Z|Y}(\mathbf{x})$, and performing a sensitivity analysis on the mean $\mu_{Z|Y}(\mathbf{x})$. Although straightforward, this approach may provide biased sensitivity index estimates; furthermore it does not allow one to quantify the error on the sensitivity indices due to the surrogate approximation.

The second one consists in substituting $G(\mathbf{x})$ by a Gaussian process $Z_N(\mathbf{x})$ having the predictive distribution $[Z(\mathbf{X})|Z(\mathcal{X}) = \mathbf{y}, \sigma^2, \boldsymbol{\theta}]$. This approach makes it possible to quantify the uncertainty due to the metamodel approximation and allows for building unbiased index estimates. In particular, the same Monte Carlo estimates of the Sobol' indices (7.11)–(7.12) can be used, where the model is replaced by a Gaussian process that has to be sampled N times for input values realized from a Sobol' quasi-random sequence.

- **Remark 7.7.4.** Gaussian process regression makes it possible to perform sensitivity analysis on complex computational models using a limited number of model evaluations. It also provides a powerful tool to visualize the main effect of a group of variables and the uncertainty of its estimate. Furthermore, it can also be used for multi-fidelity computer codes, i.e., codes which can be run at multiple levels of accuracy [26].

7.8 Further readings, code and material

Several papers and books deal with sensitivity analysis. The research activity by Andrea Saltelli (check his webpage...) has been almost completely devoted to sensitivity analysis [43, 42]. An interesting review on recent advances by Emanuele Borgonovo (Univ. Bocconi) can be found in [8]. Chapters from 31 to 39 of a recent Handbook of Uncertainty Quantification [17] are devoted to Sensitivity Analysis. Applications to environmental models can be found, e.g., in [34]. Recent contributions dealing with the use of surrogate models or reduced order models for sensitivity analysis can be found, e.g., in [21, 38].

Matlab implementations of global sensitivity analysis tools such as the Sobol' method, the EE screening method and the PAWN method can be found in the **SAFE Toolbox** (available in **matlab**, **R**, and **Python**), see [35] and <https://www.safetoolbox.info>.

Alternative as well as at <https://www.mathworks.com/matlabcentral/fileexchange/> at the following pages:

- [40759-global-sensitivity-analysis-toolbox](#)
- [47758-global-sensitivity-and-uncertainty-analysis-gsua-of-dynamical-systems-using-variance-based-methods](#)

An implementation of Monte Carlo estimation of the moment-independent importance measure alternative to the one included in SAFE can be found, e.g., at <https://github.com/mia2mia/Global-Sensitivity-Code-Borgonovo>.

An extremely general, well-documented and user-friendly framework for UQ in Matlab is **UQLab** [27], which can be found at <https://www.uqlab.com/>.

On the other hand, a Sensitivity Analysis Library in Python (NumPy) containing, among others, Sobol and Morris methods, is **SALib**, which can be found at <http://SALib.github.io/SALib/>.