HAL
open science

Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel

Matthias de Lozzo, Amandine Marrel

To cite this version:

Matthias de Lozzo, Amandine Marrel. Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel. SIAM/ASA Journal on Uncertainty Quantification, 2016, 4, pp.708-738. hal-01164215v2

HAL Id: hal-01164215

https://hal.science/hal-01164215v2

Submitted on 4 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimation of the derivative-based global sensitivity measures using a Gaussian process metamodel

Matthias De Lozzo* Amandine Marrel†

CEA, DEN, DER, F-13108 Saint Paul Lez Durance, France

November 6, 2015

Abstract

Physical phenomena are often studied using numerical simulators. Such computer codes are function of uncertain input parameters and a global sensitivity analysis (GSA) can be performed to identify their impacts on the simulator outputs. Sobol' indices, based on output variance decomposition, are commonly used to perform quantitative GSA. For many years now, other tools have been studied, closer to physical practices such as the derivative-based global sensitivity measures (DGSM). However, numerical simulators rarely provide the output gradient and DGSM estimation is not directly possible. To address this limitation, we propose to estimate the DGSMs using a Gaussian process metamodel (GPM) which approximates the simulator.

Based on this GPM, we propose two DGSM estimators: a plug-in one defined by the DGSM of the GPM predictor and another one defined by the expectation of the DGSM associated to the full-GPM. The latter is equal to the first one completed by a variance term and can be accompanied by a credibility interval. For Gaussian kernel and uniform input laws, analytical formula are given for both DGSM estimators. For all other situations, Monte-Carlo methods for the expectation approximations are proposed: a propagative version of the Gibbs sampler and a chi-square approximation. Moreover, a significance test for the full-GPM based estimator is proposed for screening. The convergence of the two GPM-based DGSM estimators and the Monte-Carlo approaches are compared on analytical test cases. Finally, we apply our work to an environmental application.

Keywords: Sensitivity analysis, derivative-based global sensitivity measures (DGSM), Gaussian process model, significance test, screening.

1 Introduction

Computer models are widely used for the representation of physical phenomena in the industry. The main reasons are the permanent increase of computational power and numerical method efficiency. Such models take as input many numerical and physical explanatory variables. They can be used in computer experimentation to explore the relationship between the variables and increase the knowledge about the physical phenomenon, while an accurate physical experimentation is too costly [41]. However these numerical simulators often have a high number of uncertain input parameters, what can lead to an important uncertainty over the model output [8, 22]. Commonly, the input uncertainties are modeled by probability distributions and a quantification of their impacts on the output is performed. These steps define a sensitivity analysis (SA) of the considered computer code. Then, the SA results could be used to reduce the model output uncertainty by enhancing the characterization of the most influential input parameters and using nominal values for the non-significant ones.

This SA can be either local (LSA), when it focus on the output behavior associated to input variations around a specific input parameter value, or global (GSA), when it considers the output behavior for uncertainties based on the whole variation space of the input parameters [39]. GSA methods are widely used for industrial problems involving many variables [40]. They can distinguish the non-significant input parameters and the significant ones in a screening context. They can also quantify and order the influences of the significant parameters and of their interactions on the model output behavior. Recently, [17] propose a review on GSA in a methodological framework, presenting screening methods, measures of importance and tools for the "deep exploration" of computer code behaviors. Any GSA method has advantages and drawbacks: some approaches require a high number of computer experiments but cover a wide scope of model behaviors, while other ones are less greedy but rely on hypotheses such as a monotonic or linear behavior of the model output with respect to the input variables. The second ones are mainly used for qualitative

*matthias.delozzo@cea.fr
†amandine.marrel@cea.fr

purposes such as screening, while the first ones are used in a quantitative aim when an influence ranking of the input parameters is wished. In this second case, the most popular are the variance-based indices, usually represented by the Sobol' index which measures the proportion of the output variance explained by an input parameter [44], considering either its single contribution (first-order index), or its whole contribution including the interactions with the other input parameters (total index, e.g. [16]). Recently, [12] introduced the Goal Oriented Sensitivity Analysis, providing an unified framework for several sensitivity analyses based on the mean output value, on a specific output quantile for excess probability considerations, and so on. With the same idea to go further than the Sobol' indices, [6] proposes new sensitivity measures considering the whole distribution of the output respect to those of the input parameters, either comparing characteristic functions [48], or measuring the covariance between input and output parameters in some reproducing kernel Hilbert spaces [14]. Based on these measures, [7] study their meaning with respect to the Sobol' indices, and propose extensions to screening. Likewise, [33] propose sensitivity indices measuring the distance between the distribution of the model output conditioned by an input parameter with the unconditioned one.

From a more physical point of view, [42] introduced the derivative-based global sensitivity measure (DGSM) which represents the mean of the squared partial derivative of the model output over the input domain. This approach can be tied with the well-known Morris method [31] which considers finite differences instead of the local derivatives: in a way, DGSMs can be viewed as a generalization of the indices obtained with the Morris method [49]. The DGSMs often have the advantage of requiring much less model evaluations than the Sobol' indices [23], especially for high-dimensional problems (many tens of input parameters) or complex models (non linear and interaction effects). Moreover, based on output partial derivatives, they provide more local information which can be complementary to the one brought by the Sobol' indices for example.

Various applications have illustrated the use of DGSMs such as an aquatic prey-predator chain [18], a biological system model [21], a flood one [24] or a reservoir simulator [49]. For a screening purpose, it has been proved that the total Sobol' indices are upper bounded up to a constant by the DGSMs, firstly in the case of uniform or normal probability distributions [43], and then in the case of a wider variety of continuous distributions [24]. Recently, lower bounds depending on the DGSMs have been proposed for the total Sobol' indices, in presence of uniformly or normally distributed inputs [5]. These derivative-based sensitivity indices have also been extended to the interaction between two input parameters in [35]. The authors call such a measure a crossed DGSM and define it by the mean of the square partial derivative of the output model according to both input variables. They also provide an inequality link between the crossed-DGSMs and the total Sobol' indices.

For all these reasons, we focus here on the use of DGSMs, especially for a screening purpose. The computation of these sensitivity indices then requires that the model supplies the adjoint code to evaluate its output partial derivatives. When this code is unavailable, the gradient can be estimated by a finite-difference method. However, in presence of few numerical simulator evaluations, this alternative can lead to an important approximation error of the output gradient, which increases the DGSM one associated to the integral quadratures. Another alternative consists of replacing the time-expensive numerical simulator by a surrogate model [11] and using its gradient; this is the choice that we make in this paper. A surrogate model, also called metamodel, is a fast mathematical model built using some evaluations of the numerical simulator in order to approach its behavior. Polynomial chaos expansions [45], artificial neural networks [9] and Gaussian process metamodels [34] are the most popular ones. Recently, metamodeling techniques have been applied with polynomial chaos expansions to DGSMs [47] and Sobol' indices [46]. Gaussian process metamodels have also been used for the estimation of Sobol' indices [28]; in particular, these surrogate models can provide credibility intervals to quantify the approximation error of such sensitivity indices. More recently, Sobol' indices have been estimated combining the estimation method developed by [19] and the Gaussian process metamodel approach [28] in order to take into account numerical and integration and metamodel errors [26].

In this paper, following the same approach of [28], we investigate the DGSM approximation based on the replacement of the numerical simulator by a Gaussian process metamodel. Moreover, we propose a DGSM-based significance test for a screening purpose, in order to separate the significant inputs from the non-significant ones. We also deal with the meaning of the DGSM from local and global points of view.

Firstly, we introduce the definition of DGSM and Gaussian process metamodel (GPM). Then, we propose two estimators of the DGSMs based on a GPM: one using the GPM predictor and another one considering the whole GPM distribution. In this second section, we also propose a credibility interval for the DGSM, using the probability density function of the metamodel. Then in Section 3, we give explicit formulations of these estimators for a particular covariance function of the GPM and specific input parameter distributions. In all other cases, we propose to estimate the DGSM formulas by several Monte-Carlo methods. Moreover, we build a significance test based on DGSM to reject the non influential inputs for a screening purpose. Finally, the different DGSM estimations and associated computational methods are compared on analytical models in Section 4. An application to an industrial test is also proposed, for a screening purpose.

2 DGSM formulation using a Gaussian process metamodel

We consider the computer code

$$y = f(x_1, \ldots, x_d)$$

where $x_1, \ldots, x_d$ are d uncertain input parameters, resumed by the vector $x = (x_1, \ldots, x_d)$. The function f maps from $\mathcal{X} = \prod_{k=1}^{d} \mathcal{X}_k \subset \mathbb{R}^d$ to $\mathbb{R}$ and is assumed to be square integrable. We are interested in the variability of the output $f(x)$ with respect to the different input parameters over their definition domain $\mathcal{X}$. In the context of Global Sensitivity Analysis (GSA) [39], the $x_1, \ldots, x_d$ are considered as realizations of the independent random variables $X_1, \ldots, X_d$ whose probability density functions $\mu_{X_1}, \ldots, \mu_{X_d}$ are known, with $\mu_{X_k}(x) > 0$ over $\mathcal{X}_k$ for any $k \in \{1, \ldots, d\}$. We want to measure the dependence between the random input parameters and the random output $Y = f(X)$ using sensitivity indices.

Usually, a GSA is based on a M-sample $\mathcal{S} = \left(X^{(i)}, Y^{(i)}\right)_{1 \leq i \leq M}$ made of M independent and identically distributed (i.i.d.) computer code runs, where the output value $Y^{(i)} = f\left(X^{(i)}\right)$ is function of the ith input vector value $X^{(i)} = \left(X_1^{(i)}, \ldots, X_d^{(i)}\right)$. $\left(X^{(i)}, Y^{(i)}\right)$ is called a computer experiment, or a code simulation.

2.1 GSA based on Sobol' indices

A classical approach in GSA consists of computing the first-order and total Sobol' indices which are based on the output variance decomposition [44, 16]. If the variables $X_1, \ldots, X_d$ are independent and if $\mathbb{E}[f^2(X)] < +\infty$, where $\mathbb{E}$ is the expectation operator, we can apply the Hoeffding decomposition to the random variable $f(X)$ [10]:

$$\begin{aligned} f(X) &= f_\emptyset + \sum_{i=1}^{d} f_i(X_i) + \sum_{i=1}^{d} \sum_{i<j}^{d} f_{ij}(X_i, X_j) + \ldots + f_{1\ldots d}(X_1, \ldots, X_d) \\ &= \sum_{u \subset \{1,\ldots,d\}} f_u(X_u) \end{aligned} \tag{1}$$

where $f_\emptyset = \mathbb{E}[f(X)]$, $f_i(X_i) = \mathbb{E}[f(X)|X_i] - f_\emptyset$ and $f_u(X_u) = \mathbb{E}[f(X)|X_u] - \sum_{v \subset u} f_v(X_v)$, with $X_u = (X_i)_{i \in u}$, for all $u \subset \{1, \ldots, d\}$. All the 2^d summands in (1) have zero mean and are mutually uncorrelated with each other. This decomposition is unique and leads to the Sobol' indices. These are the elements of the $f(X)$ variance decomposition according to the different groups of input parameter interactions in (1). More precisely for each $u \subset \{1, \ldots, d\}$, the first-order and total Sobol sensitivity indices of X_u are defined by

$$S_u = \frac{\mathbb{V}\left[f_u(X_u)\right]}{\mathbb{V}\left[f(X)\right]} \text{ and } S_u^T = \sum_{v \supset u} S_v,$$

where $\mathbb{V}$ is the variance operator. Clearly $100 S_u$ is the percentage of the output variance explained by X_u, independently from the other inputs, and $100 S_u^T$ is the percentage of the output variance explained by X_u considered separately and in interaction with the other input parameters.

In practice, we are usually interested in the first-order sensitivity indices $S_1, \ldots, S_d$, the total ones $S_1^T, \ldots, S_d^T$ and sometimes in the second-order ones S_{ij}, $1 \leq i < j \leq d$, where S_i, S_{ij} and S_i^T measure the output sensitivities due to the main effect f_i of X_i, to the interaction f_{ij} between X_i and X_j and to all the X_i contributions $(f_u)_{u \ni i}$ respectively. The model f is devoid of interactions if $\sum_{i=1}^{d} S_i \approx 1$.

Sobol' indices are widely used in GSA because of their efficiency and facility of interpretation. Indeed, for a given input parameter, the Sobol' index measures its contribution to the mean squared deviation of the model output from its mean value. **However, from a physical point of view, engineers can be more interested in its contribution to the mean value of the output gradient.** Indeed, such a consideration can highlight some sharp output effects along an input direction, in addition to the contribution of this input to the output variance. The study of partial derivatives can help to measure the granularity of the numerical simulator output according to the different input parameters. It takes into account the important local variations of the numerical simulator and can be view as a localized sensitivity measure of the phenomenon. It brings also rough to fine information about the output gradient in a context of optimization.

2.2 Derivative-based global sensitivity measures

Introduced by [42], sensitivity indices based on the partial derivatives of the computer code f have recently been studied by [43] and [24], generalizing the importance measures introduced in the Morris method [31]. They are based

on the hypothesis that for a given input parameter x_k, $k \in \{1, \ldots, d\}$, an important partial derivative with respect to this variable, over the whole probabilized input parameter space, leads to an important variation of the model output $f(x)$. These sensitivity indices are called "derivative-based global sensitivity measures" (DGSMs) and the kth, k in $\{1, \ldots, d\}$, is equal to:

$$D_k = \mathbb{E}\left[\left(\frac{\partial f(X)}{\partial x_k}\right)^2\right]. \tag{2}$$

As mentioned by [49], the DGSMs are tied with the Morris method [31]. From a synthetic point of view, this method associates to each input parameter x_k the empirical mean $\widehat{M}_k = \frac{1}{R}\sum_{r=1}^{R} d_k^{[r]}$ and the standard deviation $\widehat{\Sigma}_k = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(d_k^{[r]} - \widehat{M}_k\right)^2}$ of the elementary effect $d_k^{[r]}$ defined as

$$d_k^{[r]} = \frac{f\left(X_1^{[r]}, \ldots, X_{k-1}^{[r]}, X_k^{[r-1]} + \delta_k^{[r]}, X_{k+1}^{[r-1]}, \ldots, X_d^{[r-1]}\right) - f\left(X_1^{[r]}, \ldots, X_{k-1}^{[r]}, X_k^{[r-1]}, \ldots, X_d^{[r-1]}\right)}{\delta_k^{[r]}}$$

where $X_k^{[r]} = X_k^{[r-1]} + \delta_k^{[r]}$ for any $k \in \{1, \ldots, d\}$, with $\delta_k^{[r]}$ the $((r-1)d+k)^{\text{th}}$ random step. The construction of the random design is studied in more details in [31], starting from an initial point $X^{[0]}$.

First, the DGSM of x_k can be rewritten as $D_k = M_k^2 + \Sigma_k$ where $M_k = \mathbb{E}\left[\frac{\partial f(X)}{\partial x_k}\right]$ and $\Sigma_k = \mathbb{V}\left[\frac{\partial f(X)}{\partial x_k}\right]$. Then, we note that M_k and Σ_k generalize $\widehat{M}_k$ and $\widehat{\Sigma}_k$ respectively: the first quantities consider the first centered moments of the kth partial derivative while the second ones consider their empirical versions replacing the partial derivative by a finite difference with step $\delta_k^{[r]}$. Consequently, if the step $\delta_{X_k}^{[r]}$ is greater than the characteristic dimension of the physical problem, the Morris method focus on the global variation of the model output whereas the DGSM approach focus on local changes averaged over the whole probabilized input parameter space.

Lastly, for some non-monotonic functions, the Morris index $\widehat{M}_k$ can be almost equal to zero even if x_k has an important influence on the model output; the DGSM D_k allows to detect this dependence, with the same idea than the sensitivity measure proposed by [2] which replaces the elementary effect by its absolute value in the $\hat{M}_k$ and $\widehat{\Sigma}_k$ formulations.

Otherwise, [43] and [24] compare DGSMs and Sobol' indices from a theoretical point of view and show that for a large category of continuous distributions, the total Sobol' index S_k^T is bounded by the DGSM index D_k weighted by a constant C_k, function of the probability law of X_k: $S_k^T \leq C_k D_k$. The constant C_k is equal to $4c_k^2/\mathbb{V}[Y]$ where c_k is the Cheeger constant if the probability density function of X_k is a Boltzmann probability measure (cf. Table 1). If X_k follows the uniform law $\mathcal{U}(a, b)$, the constant c_k is equal to $c_k = \frac{b-a}{2\pi}$.

Normal $\mathcal{N}(\mu, \sigma^2)$	Exponential $\mathcal{E}(\lambda)$	Gumbel $\mathcal{G}(\mu, \beta)$	Weibull $\mathcal{W}(k, \lambda)$
$\frac{\sigma}{2}$	$\frac{1}{\lambda}$	$\frac{\beta}{\log(2)}$	$\frac{\lambda(\log(2))^{(1-k)/k}}{k}$

Table 1: Cheeger constant for different probability distributions.

Similarly, in order to compare DGSMs, it is possible to weight these quantities by the input variances. These weighted DGSMs $(\Delta_k)_{1 \leq k \leq d}$ are defined by $\Delta_k = \sigma_k^2 D_k$ where $\sigma_k^2 = \mathbb{V}[X_k]$ and have the advantage of being dimensionless. This approach is similar to the one discussed by [38] in their introduction, considering a SA index equal to the output partial derivative weighted by the input standard deviation and normalized by the standard deviation of the output. In this way, a very important DGSM, due to a single strong local variation along the direction of x_k whose uncertainty is very small, can be transformed into a small weighted DGSM. Conversely, a very small DGSM, due to many medium local variations along the direction of x_k whose uncertainty is very important, can be transformed into a medium or high DGSM. Such behaviors are more in agreement with global sensitivity measure expectations: a high value when the input parameter strongly impacts the model output on average and a small one for the inverse situation. Moreover, the comparison of weighted DGSMs is more reasonable than the comparison of DGSMs. Indeed, while the first sensitivity measures are dimensionless, the second indices have different dimensions, each of them corresponding to the associated input parameter one. As an intuitive justification, the choice of the variance term σ_k^2 to weight the DGSM D_k makes sense in an output variance decomposition, using a linear expansion of the function $f(x)$ around x^*: $f(x) \approx f^*(x) = f(x^*) + \sum_{k=1}^{d} \left.\frac{\partial f(x)}{\partial x_k}\right|_{x^*} (x_k - x_k^*)$. In this case, we obtain for independent input parameters the mean variance decomposition: $\mathbb{E}\left[\mathbb{V}\left[f^*(X)\right]|_{x^*=X^*}\right] = \sum_{k=1}^{d} \Delta_k$.

Moreover, [23] show that the estimation of a DGSM index often requires much less model evaluations than the estimation of a Sobol' one. From this consideration, the weighted DGSMs could be used in screening to exclude the

non-significant input parameters from the model specification, in a more economical way. This selection of influential variables should be almost the same as the one obtained with the Sobol' indices, because of the inequality linking these sensitivity measures. Nevertheless, these studies mention that ranking the input parameters according to the upper bound values does not lead necessarily to the same order than ranking these parameters according to the Sobol' indices. These differences can be even more important that the numerical simulator output is rough. For example, the sinus function $f(x) = \sin(\omega x)$ governed by an uniform random variable leads to an upper bound increasing to infinity with the frequency while the total Sobol' index remains equal to 1.

Last but not least, the computer code does not always provide the adjoint code for the gradient evaluation. Moreover, when this gradient can be get, which is not the case for the problems that we have to deal with, the number of available computer experiments is often too low and the integrals are badly estimated in Expression (2). Consequently in this paper, we propose to replace the computer code f by a surrogate model $\hat{f}$ with a tractable gradient which is less accurate but faster in execution [11], so allowing to have a much higher number M of Monte-Carlo runs for the expectation estimation. More particularly, we focus on the Gaussian process metamodel [34], in the same way as [3] and [28] for the estimation of the Sobol' indices. This surrogate model is often powerful for industrial studies and has a stochastic formulation which can usefully provide credibility intervals for the DGSM estimation. Consequently, the GSA is realized using M independent and identically distributed evaluations of the surrogate model according to the probabilistic law of the input variables $X_1, \ldots, X_d$.

2.3 Gaussian process metamodel

Let Ω be a sample space. We suppose that the observed computer code f can be modeled as an instance of the Gaussian process (GP) $Z : (\mathcal{X}, \Omega) \to \mathbb{R}$ indexed by $\mathbb{R}^d$ and defined for any ω in Ω by

$$Z(\cdot;\omega) = Z_1(\cdot;\omega) + Z_2(\cdot;\omega) \tag{3}$$

where $Z_1(\cdot;\omega)$ and $Z_2(\cdot;\omega)$ are independent GPs with means $\mu_{Z_1} : x \mapsto= h(x)^T\beta$ and $\mu_{Z_2} : x \mapsto 0$, and covariance functions $c_{Z_1} : (x, x') \mapsto \sigma^2 r(x, x')$ and $c_{Z_2} : (x, x') \mapsto \sigma^2\tau^2\delta_x(x')$. The random variable $Z(\cdot;\omega)$ is defined by its mean $\mu_Z : x \mapsto h(x)^T\beta$ and its covariance structure $C_Z : (x, x') \mapsto \sigma^2 \left(r(x, x') + \tau^2\delta_x(x')\right)$, β being a regression parameter vector, (σ^2, τ^2) variance parameters and $\delta.$ being the Dirac delta function. The vector $h(\cdot) = (h_1(\cdot), \ldots, h_p(\cdot))^T$ is made of p basis functions, which are commonly monomials, e.g. $\forall x \in \mathcal{X}$, $h(x) = (1\ x_1\ \ldots\ x_d)^T \in \mathbb{R}^{d+1}$ with $p = d + 1$. r is a kernel function parameterized by an hyperparameter vector $\theta \in \Theta$. The random field mentioned in (3) can be decomposed into the sum of a deterministic term $h(\cdot)^T\beta$ and a stochastic one $Z_0(\cdot;\omega)$, where $Z_0(\cdot;\omega)$ is a centered GP with same covariance.

From the GP formulation, we associate to a sample $\mathcal{A} = \left(x^{(i)}, y^{(i)}\right)_{1\le i\le n}$ the matrices $H \in \mathcal{M}_{n,p}(\mathbb{R})$ and $R \in \mathcal{M}_n(\mathbb{R})$ defined by $H = \left(h_j\left(x^{(i)}\right)\right)_{\substack{1\le i\le n\\ 1\le j\le p}}$ and $R = \left(r\left(x^{(i)}, x^{(j)}\right) + \tau^2\delta_{ij}\right)_{1\le i,j\le n}$, where δ_{ij} is the Kronecker delta. We also define the n-dimensional vectors $\mathbf{y} = \left(y^{(1)} \ldots y^{(n)}\right)^T$ and $\rho(\cdot) = \left(r\left(\cdot, x^{(1)}\right) \ldots r\left(\cdot, x^{(n)}\right)\right)^T$, where $y^{(i)}$ is supposed to be a realization of Z at point $x^{(i)}$ for some $\omega^* \in \Omega$. Note that R and ρ depend on the hyperparameters θ.

Then we note Z_C the GP Z_1 conditioned by this sample $\mathcal{A}$ and the parameters $(\beta, \sigma^2, \tau^2, \theta)$. Z_C is the new Gaussian process from which the computer code f is supposed to be an instance:

$$Z_{\mathsf{C}}(\cdot;\omega) = \left[Z_1(\cdot;\omega)|\mathcal{A}, \beta, \sigma^2, \tau^2, \theta\right] \tag{4}$$

with mean:

$$\hat{f} : x \mapsto h(x)^T\beta + \rho(x)^T R^{-1}(\mathbf{y} - H\beta) = h(x)^T\beta + \rho(x)^T\gamma \tag{5}$$

and covariance function:

$$s^2 : (x, x') \mapsto \sigma^2 \left(r(x, x') - \rho(x)^T R^{-1}\rho(x')\right). \tag{6}$$

where $\gamma = R^{-1}(\mathbf{y} - H\beta)$.

Predictor. As an instance of this posterior random field $Z_{\mathsf{C}}(\cdot;\omega)$, the response of the computer code f at a given location x is predicted by the conditional mean $\hat{f} = \mathbb{E}\left[Z_{\mathsf{C}}(\cdot;\omega)\right]$. Moreover, for any x in $\mathcal{X}$, the conditional covariance s^2 gives the quadratic risk $s^2(x) := s^2(x, x)$ of $\hat{f}(x)$.

Estimation. Formula (5) and (6) can not be used directly because the parameters $(\beta, \sigma^2, \theta)$ have to be estimated, classically by cross-validation or likelihood maximization procedures. In this paper, we consider the second method; precisely, we iterate explicit computation steps, $\hat{\beta} = \left(H^T R^{-1} H\right)^{-1} H^T R^{-1}\mathbf{y}$ and $\widehat{\sigma^2} = n^{-1}(\mathbf{y} - H\hat{\beta})^T R^{-1}(\mathbf{y} - H\hat{\beta})$, and a numerical minimization one, $\hat{\theta} \in \operatorname{argmin}_{\theta\in\Theta} \left\{\widehat{\sigma^2}\sqrt[n]{\det(R)}\right\}$. Then, the parameter set $(\beta, \theta, \sigma^2)$ in Equations

(5) and (6) is replaced by $(\hat{\beta}, \hat{\theta}, \widehat{\sigma^2})$. Hereafter, we always consider the estimated parameters but write $(\beta, \theta, \sigma^2)$ in order to lighten up the equations. For a same purpose, we do not consider a full Bayesian framework and not not take into account the variability of the estimators in Expressions (5) and (6).

Nugget effect. When the parameter τ^2 is forced to zero, the predictor $\hat{f}$ interpolates the learning sample $\mathcal{A}$ [34]. This situation can be too restrictive for the predictor shape, e.g. an important value of n can provide an ill-conditioned correlation matrix R and a sharp predictor. On the contrary, allowing a non-null value for τ^2 often leads to a smoother predictor $\hat{f}$ and a higher generalization ability; this change is called a nugget effect. The value of τ^2 can be optimized by maximum likelihood, jointly with the other hyperparameters.

In our situation, we deal with sparse problems characterized by a weak density of observations in the input parameter space. Consequently, the nugget effect plays a part in the good conditioning of the matrix R but it is usefulness for the prediction step, due to the regularity of the computer codes to substitute. This is the reason why the random process Z_C is defined by the Gaussian process Z_1 conditioned by the learning sample, rather by the Gaussian process Z conditioned by this sample.

Covariance choice. Finally, we limit ourselves to the case where the kernel r is a tensorized product of monodimensional stationary kernels: $\forall x, x' \in \mathcal{X},\ r(x, x') = \prod_{i=1}^{d} r_i(x_i - x'_i)$. This common hypothesis allows the analytical developments presented in this paper. Furthermore, we suppose that these monodimensional kernels come from a same family of covariance functions that the user has to choose: exponential, Gaussian, Matérn, ... Complementary informations about Gaussian process metamodels can be found in the book *Gaussian Processes for Machine Learning* of [34].

2.4 GPM-based estimation for DGSMs

In this paper, we propose two approaches for the estimation of the DGSM D_k, $k \in \{1, \ldots, d\}$, using a GPM in the same way as [28]. The first one uses no more than the estimator $\hat{f}$ of f defined by (5) and builds the plug-in estimator:

$$\hat{D}_k^{(1)} = \mathbb{E}\left[\left(\frac{\partial \hat{f}(X)}{\partial x_k}\right)^2\right].$$

The second approach uses the law of the stochastic DGSM $\widehat{RD}_k^{(2)}(\omega)$ defined by the random variable:

$$\widehat{RD}_k^{(2)}(\omega) = \mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |\omega\right]. \tag{7}$$

From this, the DGSM D_k is estimated by the full-GPM estimator which is the $\widehat{RD}_k^{(2)}(\omega)$ expectation:

$$\hat{D}_k^{(2)} = \mathbb{E}\left[\widehat{RD}_k^{(2)}(\omega)\right].$$

Similarly to the use of $s^2(x) = \mathbb{V}\left[Z_{\mathsf{C}}(x;\omega)\right]$ as an indicator of the $\hat{f}(x)$ error, we could quantify the $\hat{D}_k^{(2)}$ estimation error using the $\widehat{RD}_k^{(2)}(\omega)$ variance:

$$\hat{s}_k^2 = \mathbb{V}\left[\widehat{RD}_k^{(2)}(\omega)\right]. \tag{8}$$

This quantity can be estimated using intensive Monte-Carlo sampling of the random variable $\widehat{RD}_k^{(2)}(\omega)$. Similarly, credibility intervals associated to $\hat{D}_k^{(2)}$ are developed in Section 3.

2.4.1 Plug-in estimation of a DGSM

The plug-in estimation of the DGSM consists of the computation of $\hat{D}_k^{(1)}$, which corresponds to the DGSM directly estimated from the GPM predictor $\hat{f}$ given in Equation (5). Knowing that $\hat{f}(x) = h(x)^T\beta + \rho(x)^T\gamma$, the kth partial derivative of $\hat{f}(x)$ is:

$$\frac{\partial \hat{f}(x)}{\partial x_k} = \frac{\partial h(x)^T}{\partial x_k}\beta + \frac{\partial \rho(x)^T}{\partial x_k}\gamma$$

where $\frac{\partial h(x)}{\partial x_k} = \left(\frac{\partial h_1(x)}{\partial x_k}, \ldots, \frac{\partial h_p(x)}{\partial x_k}\right)^T$ and $\frac{\partial \rho(x)}{\partial x_k} = \left(\frac{\partial r\left(x, x^{(1)}\right)}{\partial x_k}, \ldots, \frac{\partial r\left(x, x^{(n)}\right)}{\partial x_k}\right)^T$.

Under the hypothesis of a tensorized product of monodimensional kernels for the covariance (see Section 2.3), we obtain for any $i \in \{1, \dots, n\}$:

$$\frac{\partial r\left(x, x^{(i)}\right)}{\partial x_k} = \frac{\partial r_k\left(x_k, x_k^{(i)}\right)}{\partial x_k} \prod_{\substack{j=1 \\ j \neq k}}^{d} r_j\left(x_j, x_j^{(i)}\right).$$

Then, considering element-wise mean for matrices, i.e. $(\mathbb{E}[A])_{ij} = \mathbb{E}[A_{ij}]$ for any real matrix A, we obtain a first estimator of the sensitivity measure D_k:

$$\hat{D}_k^{(1)} = \beta^T \mathbb{E}\left[\frac{\partial h(X)}{\partial x_k}\frac{\partial h(X)^T}{\partial x_k}\right]\beta + \gamma^T \mathbb{E}\left[\frac{\partial \rho(X)}{\partial x_k}\frac{\partial \rho(X)^T}{\partial x_k}\right]\gamma + 2\beta^T \mathbb{E}\left[\frac{\partial h(X)}{\partial x_k}\frac{\partial \rho(X)^T}{\partial x_k}\right]\gamma \tag{9}$$

whose integrals have to be analytically or numerically computed.

In this paper, we present the most common situation where the prior mean is constant. Consequently, we have $h(x) = 1$ and Equation (9) becomes:

$$\hat{D}_k^{(1)} = \gamma^T \mathbb{E}\left[\frac{\partial \rho(X)}{\partial x_k}\frac{\partial \rho(X)^T}{\partial x_k}\right]\gamma.$$

Remark 1. *Extensions of the following results to a more complex prior mean, such as a polynomial, are straightforward from Equation (9).*

Then, thanks to the tensorized form of the kernel functions and of the independence hypothesis of the input parameters $X_1, \dots, X_d$, the matrix $A^{[k]} = \mathbb{E}\left[\frac{\partial \rho(X)}{\partial x_k}\frac{\partial \rho(X)^T}{\partial x_k}\right]$ can be better rewritten

$$A^{[k]} = \underbrace{\mathbb{E}\left[\frac{\partial \rho_k(X_k)}{\partial x_k}\frac{\partial \rho_k(X_k)^T}{\partial x_k}\right]}_{B^{[k]}} \bigodot_{\substack{l=1 \\ l \neq k}}^{d} \underbrace{\mathbb{E}\left[\rho_l(X_l)\rho_l(X_l)^T\right]}_{C^{[l]}} \tag{10}$$

where $\rho_l(x) = \left(r_l\left(x_l, x_l^{(1)}\right) \dots r_l\left(x_l, x_l^{(n)}\right)\right)^T$, for any $l \in \{1, \dots, d\}$ and $\bigodot$ is the element-wise multiplication operator. The computation of $A^{[1]}, \dots, A^{[d]}$ requires $n(n+1)d$ monodimensional integral evaluations, because these matrices are function of the symmetric matrices $B^{[1]}, C^{[1]}, \dots, B^{[d]}, C^{[d]}$, each of which having $\frac{n(n+1)}{2}$ different elements of the form:

$$C_{ij}^{[k]} = \mathbb{E}\left[r_k\left(X_k, x_k^{(i)}\right) r_k\left(X_k, x_k^{(j)}\right)\right] \tag{11}$$

or

$$B_{ij}^{[k]} = \mathbb{E}\left[\frac{\partial r_k\left(X_k, x_k^{(i)}\right)}{\partial x_k}\frac{\partial r_k\left(X_k, x_k^{(j)}\right)}{\partial x_k}\right]. \tag{12}$$

For particular cases of covariance functions and input parameter laws, analytical simplifications can be applied to Equation (10). More generally, these integrals can be quickly approached by Monte-Carlo sampling.

2.4.2 Full-GPM estimation of a DGSM

Another approach consists of the estimation of the sensitivity measure D_k based on the full Gaussian process $Z_{\mathsf{C}}(x;\omega)$, rather than on its mean only. More precisely, we consider the full-GPM estimator $\hat{D}_k^{(2)}$ which can be easily developed as

$$\hat{D}_k^{(2)} = \hat{D}_k^{(1)} + \mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right]. \tag{13}$$

In this way, the estimator $\hat{D}_k^{(2)}$ is equal to the plug-in one, $\hat{D}_k^{(1)}$, completed by an additive positive term associated to the variance of the GPM; consequently, we have $\hat{D}_k^{(2)} \geq \hat{D}_k^{(1)}$. We explicit this new part in the following manner:

$$\mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right] = \sigma^2\left\{\mathbb{E}\left[\left.\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\right|_{(U,V):=(X,X)}\right] - \mathbf{1}^T\left(R^{-1} \bigodot A^{[k]}\right)\mathbf{1}\right\} \tag{14}$$

where $\mathbf{1}$ is the all-ones element of $\mathbb{R}^n$. A new term appears in this expression:

$$\mathbb{E}\left[\left.\frac{\partial^2 r(U,V;\theta)}{\partial u_k \partial v_k}\right|_{(U,V):=(X,X)}\right] \tag{15}$$

which is easily estimable by Monte-Carlo sampling. Derivations of (13) and (14) are given in Appendix.

In addition, the full-GPM estimator (13) can be completed by an approximation of its error using the variance (8) or a credibility interval computed by simulations.

The next part of this paper deals with the computation of terms (11), (12) and (15) required for the approximation of $\hat{D}^{(1)}$ and $\hat{D}^{(2)}$ and for the construction of credibility intervals. Significance tests are also proposed.

3 Numerical implementation of the GPM-based DGSM estimators

In this part, we propose different numerical implementations of the DGSM estimators derived from GPM and defined in Section 2. Precisely, we compute the matrices (11), (12) and (15) used in $\hat{D}_k^{(1)}$ and $\hat{D}_k^{(2)}$, analytically or with a Monte-Carlo method.

3.1 Analytical results for uniform laws and Gaussian kernels

First, we consider a specific covariance function for the GP and a certain type of input distribution, in order to get analytical results for the computation of the estimator $\hat{D}_k^{(1)}$ of the sensitivity index D_k. More precisely, we consider classical Gaussian kernel functions

$$r_k(x_k, x_k') = \exp\left(-\frac{(x_k - x_k')^2}{2\theta_k^2}\right),\ k \in \{1, \dots, d\},$$

and independent input parameters $X_1, \dots, X_d$ following uniform laws:

$$X_k \sim \mathcal{U}\left([m_k, M_k]\right),\ k \in \{1, \dots, d\}.$$

First of all, under these considerations and using Lemma 1 (see Appendix), we rewrite the term (11)

$$C_{ij}^{[k]} = \frac{\sqrt{\pi}\theta_k \mathbb{P}\left[W_{i,j}^{[k]} \in [m_k, M_k]\right] e^{-\frac{\left(x_k^{(i)} - x_k^{(j)}\right)^2}{4\theta_k^2}}}{M_k - m_k} \tag{16}$$

where $W_{i,j}^{[k]} \sim \mathcal{N}\left(\frac{x_k^{(i)} + x_k^{(j)}}{2}, \frac{\theta_k^2}{2}\right)$.

Furthermore, using Lemma 2 (see Appendix), we can show that the term (12) is equal to

$$\begin{aligned} B_{ij}^{[k]} &= -\frac{(x_k^{(i)} - x_k^{(j)})^2}{4\theta_l^4} C_{ij}^{[k]} \\ &+ \frac{\sqrt{\pi}}{2\theta_k(M_k - m_k)} \left\{ a_{i,j}^{[k]} \varphi\left(a_{i,j}^{[k]}\right) - b_{i,j}^{[k]} \varphi\left(b_{i,j}^{[k]}\right) + \mathbb{P}\left[Z_{i,j}^{[k]} \in \left[a_{i,j}^{[k]}, b_{i,j}^{[k]}\right]\right] \right\} e^{-\frac{\left(x_k^{(i)} - x_k^{(j)}\right)^2}{4\theta_k^2}} \end{aligned} \tag{17}$$

where $Z_{i,j}^{[k]} \sim \mathcal{N}(0,1)$, $a_{i,j}^{[k]} = \frac{2m_k - x_k^{(i)} - x_k^{(j)}}{\sqrt{2}\theta_k}$, $b_{i,j}^{[k]} = \frac{2M_k - x_k^{(i)} - x_k^{(j)}}{\sqrt{2}\theta_k}$ and $\varphi(.)$ is the probability density function of the standard normal law.

Finally, we have:

$$\mathbb{E}\left[\left.\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\right|_{(U,V):=(X,X)}\right] = \frac{1}{\theta_k^2}.$$

Expressions (16) and (17) only use evaluations of probability density and distribution functions. Consequently, the DGSM estimators $\hat{D}_k^{(1)}$ and $\hat{D}_k^{(2)}$ do not require integral quadratures to approach (11), (12) and (15) in presence of Gaussian kernel functions and uniform distributions for the input parameters. Under these assumptions, Section 4 illustrates the computational superiority of this method in comparison to Monte-Carlo approaches.

3.2 Monte-Carlo approximation for general case

In spite of the exact formulations of the GPM-based DGSM estimators, kernel functions are not always Gaussian and the input distributions are not always uniform in many industrial applications. Consequently, in such cases, numerical methods are required for the approximation of both DGSM estimators defined by (9) and (13).

The first estimator given in (9) is a plug-in one, which represents the mean of the squared estimator of a f partial derivative. This estimator having a deterministic and analytical formulation, the mean is approached by crude Monte-Carlo according to the specified input distributions. The computational cost is not high because the method only requires to estimate monodimensional integrals. A similar sampling technique can approach the variance (8) expressed in terms of simple and double integrals.

The second estimator given in (13) is more complex because it represents the stochastic mean of the squared Gaussian process Z_C averaged over the input parameter space $\mathcal{X}$:

$$\hat{D}_k^{(2)} = \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |\omega\right]\right].$$

In order to get the second GPM-based estimator $\hat{D}_k^{(2)}$, we decide to approach numerically the distribution of the stochastic DGSM $\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |\omega\right]$ using N instances:

$$\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega_1)}{\partial x_k}\right)^2\right], \ldots, \mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega_N)}{\partial x_k}\right)^2\right],$$

or more precisely using N instances:

$$M^{-1}\sum_{i=1}^{M}\left(\frac{\partial Z_{\mathsf{C}}\left(X^{(i)};\omega_1\right)}{\partial x_k}\right)^2, \ldots, M^{-1}\sum_{i=1}^{M}\left(\frac{\partial Z_{\mathsf{C}}\left(X^{(i)};\omega_N\right)}{\partial x_k}\right)^2$$

where $\mathbf{X} = \left(X^{(1)}, \ldots, X^{(M)}\right)$ are M i.i.d. instances of X. In this case, we have:

$$\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k} := \begin{pmatrix} \frac{\partial Z_{\mathsf{C}}\left(X^{(1)};\omega\right)}{\partial x_k} \\ \vdots \\ \frac{\partial Z_{\mathsf{C}}\left(X^{(M)};\omega\right)}{\partial x_k} \end{pmatrix} \sim \mathcal{N}\left(b_M, C_M\right)$$

with

$$\left(b_M\right)_i = \frac{\partial h(X^{(i)})^T}{\partial x_k}\beta + \frac{\partial \rho(X^{(i)})^T}{\partial x_k}\gamma, \ i \in \{1, \ldots, M\}$$

and

$$\left(C_M\right)_{ij} = \sigma^2\left(\left.\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\right|_{(U,V)=(X^{(i)},X^{(j)})} - \frac{\partial \rho(X^{(i)})^T}{\partial x_k} R^{-1} \frac{\partial \rho(X^{(j)})}{\partial x_k}\right), \ i,j \in \{1, \ldots, M\}.$$

In the following, we propose two options for the simulation of instances of the random variable

$$M^{-1}\sum_{i=1}^{M}\left(\frac{\partial Z_{\mathsf{C}}(X^{(i)};\omega)}{\partial x_k}\right)^2, \tag{18}$$

in order to estimate its mean and, in a second phase, an associated credibility interval. The first one is based on a propagative version of the Gibbs sampler which approximates the law of the Gaussian vector $\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k}$ and the second one is based on a chi-square approximation of the probabilistic law of the quadratic form (18). Both approaches directly use the vector b_M and the matrix C_M.

3.2.1 Use of a propagative version of the Gibbs sampler

Concerning the first option, we simulate realizations of the Gaussian vector $\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k}$ using the propagative version of the Gibbs sampler introduced by [25]. More precisely, we consider the blocking strategy of pivots presented in this paper and described in Algorithm 1. This propagative version is more efficient than the classical Gibbs sampler introduced by [13], because it does not require the inversion of the matrix C_M which is high-dimensional. The default value for the block length L is 5, which is used by [25].

With this method, it is possible to approach the distribution of the stochastic DGSM $\hat{D}_k^{(2)}(\omega)$. For this end, we define the ith instance of the estimation of $\widehat{RD}_k^{(2)}$ based on Algorithm 1 by

$$\hat{D}_{k,i}^{(2),M} = \frac{1}{M}\sum_{l=1}^{M}\left(\frac{\partial Z_{\mathsf{C}}\left(X^{(l)};\omega_i\right)}{\partial x_k}\right)^2 \tag{19}$$

for all $i \in \{1, \dots, N\}$. Then, we propose to estimate $\hat{D}_k^{(2)}$ by:

$$\hat{D}_k^{(2),N,M} = \frac{1}{N} \sum_{i=1}^{N} \hat{D}_{k,i}^{(2),M} \tag{20}$$

and its credibility interval of level α by:

$$\mathsf{CI}_\alpha\left(D_k\right) = \left[q_{k,\alpha/2}^{(2),M,N}, q_{k,1-\alpha/2}^{(2),M,N}\right] \tag{21}$$

where $q_{k,\alpha/2}^{(2),M,N}$ and $q_{k,1-\alpha/2}^{(2),M,N}$ are the $\alpha/2$ and $1-\alpha/2$ empirical quantiles of the sample $\left(\hat{D}_{k,i}^{(2),M}\right)_{1 \leq i \leq N}$. In this paper, we compute the quantities using the `quantile` function of `R` with the default parametrization.

Data: the mean vector b_M, the covariance matrix C_M, the block length L and the number of instances N;
Result: N realizations of the random vector $\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k}$;
Extract $s = \left(\frac{1}{(C_m)_{ii}}\right)_{1 \leq i \leq M}$;
Update $C_M := C_M \bigodot \left(ss^T\right)$ where $\odot$ is the element-wise multiplication operator;
Let $A^{(0)} := 0_{\mathbb{R}^M}$;
for $i = 1, \dots, N$ **do**
1. Take a sample $S_i = \{i_1, \dots, i_L\}$ of L integers in $\{1, \dots, M\}$ without replacement;
2. $A^{(i)} = A^{(i-1)} + C_{M,i} B_i^{-1} \left(U_i - A^{(i-1)}[S_i]\right)$, where :
   - $U_i \sim \mathcal{N}\left(0_{\mathbb{R}^L}, I_L\right)$,
   - B_i is the upper triangular matrix of the $C_M[S_i, S_i]$ Cholesky decomposition,
   - $C_{M,i} := C_M[:, S_i]$.

end
for $i = 1, \dots, N$ **do**
$\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega^{(i)})}{\partial x_k} := b_M + A^{(i)} \odot s$.
end

Algorithm 1: Propagative version of the Gibbs sampler with the blocking strategy of pivots.

Remark 2. *This method based on the Gibbs sampler can be easily extended to the crossed DGSM introduced by [35]:*

$$\hat{D}_{ij} = \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial^2 Z_{\mathsf{C}}(X;\omega)}{\partial x_i \partial x_j}\right)^2 |\omega\right]\right].$$

Such a sensitivity measure allows to quantify the influence of an interaction between two input parameters on the output. [35] recommend their use for the detection of additive structure in the model; indeed, the authors prove that a crossed DGSM D_{ij} equal to zero implies that the input parameters x_i and x_j do not interact together in the model.

3.2.2 Use of a chi-square approximation

Concerning the second option, we consider a chi-square approximation of the stochastic DGSM $\widehat{RD}_k^{(2)}(\omega)$:

$$\begin{aligned} \widehat{RD}_k^{(2)}(\omega) \approx \hat{D}_k^{(2),M}(\omega) &= \frac{1}{M} \sum_{i=1}^{M} \left(\frac{\partial Z_{\mathsf{C}}(X^{(i)};\omega)}{\partial x_k}\right)^2 \\ &= \left(\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k}\right)^T A \frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k} =: Q_k(\omega;\mathbf{X}) \end{aligned} \tag{22}$$

where $A = MI_M \in \mathcal{M}_M(\mathbb{R})$. Recently, [27] propose to approach the distribution of the quadratic form $Q_k(\omega;\mathbf{X})$ using a noncentral chi-squared distribution $\chi_l^2(\delta)$ with $(l,\delta) \in \mathbb{N}^* \times \mathbb{R}_+$. This method does not require any matrix inversion or spectral decomposition, which is an advantage when the dimension of the covariance matrix C_M is important. More precisely, the authors rewrite $Q_k(\omega;\mathbf{X})$ as a weighted sum of non-central chi-squared variables whose weights are the eigenvalues of $C_M^{1/2} A C_M^{1/2}$. Then, they determine the hyperparameters of the $\chi_l^2(\delta)$ distribution so

that its skewness is equal to the $Q_k(\omega;\mathbf{X})$ one and the distance between the kurtoses of these random variables is minimal.

From this chi-square approximation of the $Q_k(\omega;\mathbf{X})$ distribution, it is possible to approach quickly the mean of $\widehat{RD}_k^{(2)}(\omega)$ and its quantiles, in order to obtain an estimator $\hat{D}_k^{(2)}$ of D_k with a credibility interval. This method is described in Algorithm 2.

Data: the mean vector b_M, the covariance matrix C_M and the number of instances N;
Result: N realizations of the random variable $\hat{D}_k^{(2),M}(\omega)$;
$c_1 = n^{-1}\operatorname{Tr}(C_M) + n^{-1}b_M^T b_M$;
$c_2 = n^{-2}\operatorname{Tr}(C_M^2) + 2n^{-2}b_M^T C_M b_M$;
$c_3 = n^{-3}\mathbf{1}^T\left(C_M^2 \odot C_M\right)\mathbf{1} + 3n^{-3}\left(C_M b_M\right)^T C_M b_M$;
$c_4 = n^{-4}\mathbf{1}^T\left(C_M^2 \odot C_M^2\right)\mathbf{1} + 4n^{-4}\left(\left(C_M^T\right)^T b_M\right)^T C_M b_M$;
$s_1 = c_3 c_2^{-1.5}$, $s_2 = c_4 c_2^{-2}$, $\mu_1 = c_1$ and $\sigma_1 = \sqrt{2c_2}$;
if $s_1^2 > s_2$ **then**
  $a = (s_1 - \sqrt{s_1^2 - s_2})^{-1}$;
  $\delta = s_1 a^{-3} - a^{-2}$;
  $l = a^{-2} - 2\delta$;
else
  $a = s_1^{-1}$;
  $\delta = 0$;
  $l = c_2^3 c_3^{-2}$;
end
$\mu_2 = l + \delta$;
$\sigma_2 = \sqrt{2}a$;
for $i = 1, \ldots, N$ **do**
  $\hat{D}_{k,i}^{(2),M} = (u_i - \mu_2)\sigma_2^{-1}\sigma_1 + \mu_1$ where $u_i \sim \chi_l^2(\delta)$;
end

Algorithm 2: Stochastic DGSM sampling using a noncentral chi-squared approximation.

At the end of Algorithm 2, we can compute the estimator $\hat{D}_k^{(2),N,M}$ and the credibility interval $\mathrm{CI}_\alpha\left(D_k^{(2)}\right)$ given in Equations (20) and (21).

In Section 4, the different Monte-Carlo approaches are compared from accuracy and CPU time points of view.

3.3 Significance tests for screening

Finally, we would like to use the GPM-based DGSM estimators for a screening purpose, in order to distinguish the non-significant input parameters and the significant ones. To this end, we propose to test the nullity of the DGSM D_k using a statistical significance test. In this case, the null hypothesis is "$\mathcal{H}_0$: $D_k = 0$" and the alternative one is "$\mathcal{H}_1$: $D_k \neq 0$". These hypotheses can be rewritten "$\mathcal{H}_0$: $\forall x \in \mathcal{X},\ \frac{\partial f(x)}{\partial x_k} = 0$" and "$\mathcal{H}_1$: $\exists \mathcal{X}_0 \subset \mathcal{X},\ \forall x \in \mathcal{X}_0,\ \frac{\partial f(x)}{\partial x_k} \neq 0$".

Under $\mathcal{H}_0$, the mean m_M of the Gaussian vector $\frac{\partial Z_{\mathsf{C}}(\mathbf{X})}{\partial x_k}$ is the null vector of $\mathbb{R}^M$. Consequently, the p-value associated to this statistical test and to the estimator $\hat{D}_k^{(2)}$ is

$$p_{\mathsf{val},k} = \mathbb{P}\left[Q_k(\omega;\mathbf{X}) > \hat{D}_k^{(2)} | \mathcal{H}_0\right],$$

and we use the chi-square approximation or the Gibbs sampler propagative version in order to approximate this probability under $\mathcal{H}_0$. In this way, we can sort the input parameters according to their p-values; this approach seems more robust for us than sorting according to the values of the DGSM estimators because this method considers the number of observations, the estimator variances and the sensitivity index sizes.

4 Numerical experiments

In this section, we propose some numerical experiments to compare the different GPM-based DGSM estimators previously developed. In a first part, we study the convergence of the plug-in and full-GPM DGSM estimators and compare, for the second one, the different Monte-Carlo approximations proposed in Section 3.2. In a second part, we apply the DGSMs to an industrial application and compare the results with those obtained with the classical Sobol' indices; we also apply the significance test proposed in Section 3.3 for a screening purpose.

4.1 Comparison of the different estimators and numerical methods for the DGSM computation

In this first part, we are interested in the convergence of the different DGSM estimators (plug-in and full-DGSM) according to the learning sample size n and, for the full-GPM, in the comparison of the different Monte-Carlo approximation methods. We consider the classical Ishigami function defined by

$$f(X) = \sin(X_1) + 7\sin^2(X_2) + 0.1X_3^4 \sin(X_1)$$

where X_1, X_2, X_3 are i.i.d. $\mathcal{U}([-\pi, \pi])$. Its partial derivatives are:

$$\frac{\partial f(X)}{\partial x_1} = \cos(X_1)\left(1 + 0.1X_3^4\right), \quad \frac{\partial f(X)}{\partial x_2} = 14\cos(X_2)\sin(X_2) \quad \text{and} \quad \frac{\partial f(X)}{\partial x_3} = 0.4X_3^3 \sin(X_1).$$

Table 2 contains the theoretical DGSMs and Sobol' indices. The conclusions obtained with the total Sobol' indices and the DGSMs are qualitatively and quantitatively different. According to the first ones, the more influential input is X_1 (45% of the explained output variance) while X_2 is the more significant according to the second ones (57% of the DGSM sum). Concerning X_3, results are roughly the same with both types of sensitivity indices.

	X_1	X_2	X_3
1st order Sobol	0.31	0.44	0
Total Sobol	0.56 (45%)	0.44 (36%)	0.24 (19%)
DGSM	7.7 (18%)	24.5 (57%)	11.0 (25%)

Table 2: Theoretical DGSMs and Sobol' indices for the Ishigami function.

Procedure

In the following, the DGSM estimators are built from a GPM with a constant mean and a stationary tensorized covariance based on Gaussian kernels (see Section 2.3). First, we consider a sample $\mathcal{A} = \left(x^{(i)}, f(x^{(i)})\right)_{1 \le i \le n}$ where $\left(x^{(i)}\right)_{1 \le i \le n}$ is an optimized Latin Hypercube Sample [30] of size n in $[-\pi, \pi]^3$. This design of experiments is built using the `maximinSA_LHS` function of the `R` package `DiceDesign v 1.2`, producing a low-discrepancy LHS based on the centered L^2-discrepancy criteria [20] and on the Enhanced Stochastic Evolutionary [37]. Then, the GPM is obtained using this learning sample and the `R` package `DiceKriging v 1.5.3` [36], the parameters being estimated by likelihood maximization with a BFGS algorithm [1].

From this GPM, we want to:

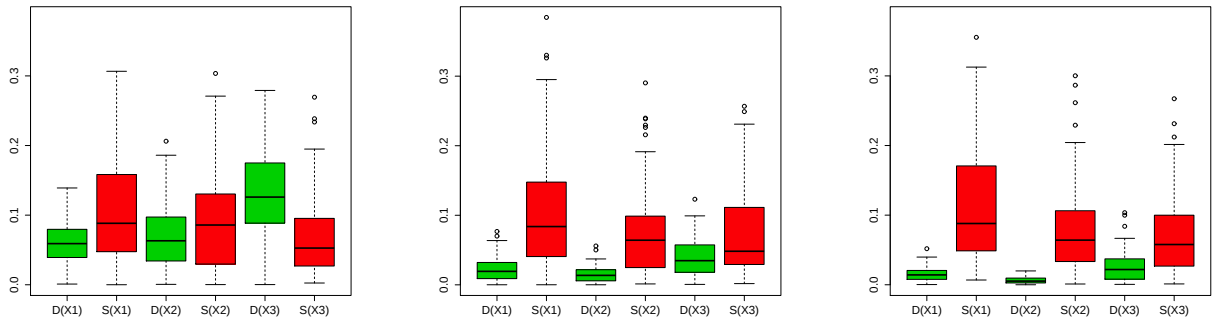
1. compare the plug-in and full-GPM DGSM estimators using the analytical results presented in Section 3.1;
2. compare the Monte-Carlo approximation methods from accuracy and CPU time points of view.

But right before, we can look at the CPU time associated to the computation of the Sobol' indices and DGSMs from a Gaussian process model. Precisely, we consider a GPM built from $n = 200$ observations, with a constant mean and a Gaussian covariance function. Total Sobol' indices are approximated using the `sobolGP` function of the R package `sensitivity`, with the option `sobol2002`, and DGSMs are approached using the chi-square approximation. This calculation is repeated for 100 different designs of experiments and for both kind of sensitivity measures, the number of X samplings is equal to 1000. Results show that the computation of the Sobol' indices lasts 7.51 (+/- 1.96) seconds on average, while the DGSM one lasts 0.20 (+/-0.20) second, which demonstrates the computational superiority of these indices. We also compare the distribution of the absolute error of estimation, normalized by the maximum of the true sensitivity indices. We justify this choice of normalization by the desire to scale all the sensitivity measures into the interval $[0, 1]$ in order to compare quantities having the same variation range. Figure 1 represents the results associated to this normalized error for the previous GPM. We can see that the errors of the estimated DGSMs is significantly lower than the estimated Sobol' index one. Moreover, Sobol' index estimators have a greater variability than the DGSM ones. This difference is all the more important that the number of observations n is high (e.g. for $n = 300$), and conversely (e.g. for $n = 100$).

4.1.1 Comparison of plug-in and full-GPM DGSM estimators

Now, we run both estimators for different learning sample sizes $n \in \{50, 100, 150, 200\}$ and, for each learning sample, we measure the accuracy of the surrogate model using the prediction coefficient

$$Q^2 = 1 - \frac{\sum_{i=1}^{n_{\text{test}}} \left(y^{\text{test},(i)} - \hat{y}^{\text{test},(i)}\right)^2}{\sum_{i=1}^{n_{\text{test}}} \left(y^{\text{test},(i)} - \bar{y}^{\text{test}}\right)^2}$$



(a) $n = 100$ (b) $n = 200$ (c) $n = 300$

Figure 1: Comparison of the normalized error distributions for DGSM (D) and total Sobol' index (S) estimations from a GPM.

where $(x^{\text{test},(i)}, y^{\text{test},(i)})_{1 \leq i \leq n_{\text{test}}}$ is a test sample built from an uniform grid of $n_{\text{test}} = 10000$ nodes and $\bar{y}^{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} y^{\text{test},(i)}$. Note that this accuracy criterion is also computed for the partial derivatives of the Ishigami function. Finally, all these steps are repeated 1000 times and we look at the mean, the 0.025 and the 0.975 quantiles of the Monte-Carlo distributions of both estimators.

Figure 2 represents the evolution of the plug-in and full-GPM DGSM estimators in function of the sample size n. It shows also the evolution of the plug-in ones in function of the accuracy criterion Q^2 associated to the corresponding f partial derivatives. In this case, each star represents a plug-in DGSM estimation associated to one sample size and one Monte-Carlo run. First of all, we can arrange the input parameters according to their influences almost surely, when the number of observations is greater than 170. Moreover, we can conclude that the second variable is the most influential from 50 observations. Thirdly, looking at the dashed lines which represent 0.025 and the 0.975 quantiles of the Monte-Carlo estimator distribution, the full-GPM DGSM estimators lead to possibly much higher values than the plug-in ones. This phenomenon can be explained by the difference between both estimators, which is positive and equal to the mean variance of the stochastic squared partial derivative of the conditioned Gaussian process (see Equation (13)). Indeed, this quantity can be important for small designs of experiments and tends to zero when the number of observations increases. The same phenomenon has been observed with a Matérn kernel.

Then, Figure 3 shows that the differences of convergence rate between the explanatory variable DGSMs is due to the fitting quality of the different partial derivatives which differs from an input parameter to another. Particularly, the performance of the GPM to predict the partial derivatives is weaker for the input parameter X_3 than for the other ones. This may be due to the presence of this variable only in an interaction term, with a small coefficient and under the form of a non-linear effect almost equal to zero over the half of the variation domain of X_3. In this way, its influence could be hidden by the other input parameters for weak sample size.

Finally, both estimators converge to the same value which is the theoretical DGSM and for small sample sizes, the full-GPM one can make some important excesses (here with the input X_2) or get closer to the right value (here with the input X_1). In absolute terms, **the full-GPM estimator is more justified because it takes up the hypothesis underlying the GPM: "the model $f(x)$ is an instance $Z_{\mathsf{C}}(\cdot;\omega^*)$ of the Gaussian process (4)".** Consequently, **under this assumption, the theoretical DGSM D_k is the DGSM of $Z_{\mathsf{C}}(\cdot;\omega^*)$, with an unknown $\omega^* \in \Omega$, and the best estimator in a mean-square sense is the mean of $\hat{D}_k^{(2)}(\omega)$ over Ω.**

4.1.2 Comparison of Monte-Carlo approximations for the stochastic DGSM

Now, we focus on the distribution of the random variable $\widehat{RD}^{(2)}(\omega)$ and compare the three following Monte-Carlo methods:

1. sampling according to the multivariate Gaussian law of $\frac{\partial Z_{\mathsf{C}}(\mathbf{X};\omega)}{\partial x_k}$, using the `mvrnorm` function of the `R` package `MASS`;

2. sampling using the propagative version of the Gibbs sampler based on a blocking strategy of pivots, proposed in Section 3.2.1;

3. sampling using the chi-square approximation proposed in Section 3.2.2.

The first method, based on a eigendecomposition of the covariance matrix, is used for reference. Indeed, from an accuracy point of view, this Monte-Carlo approach does not use any probability law approximation, contrary to the

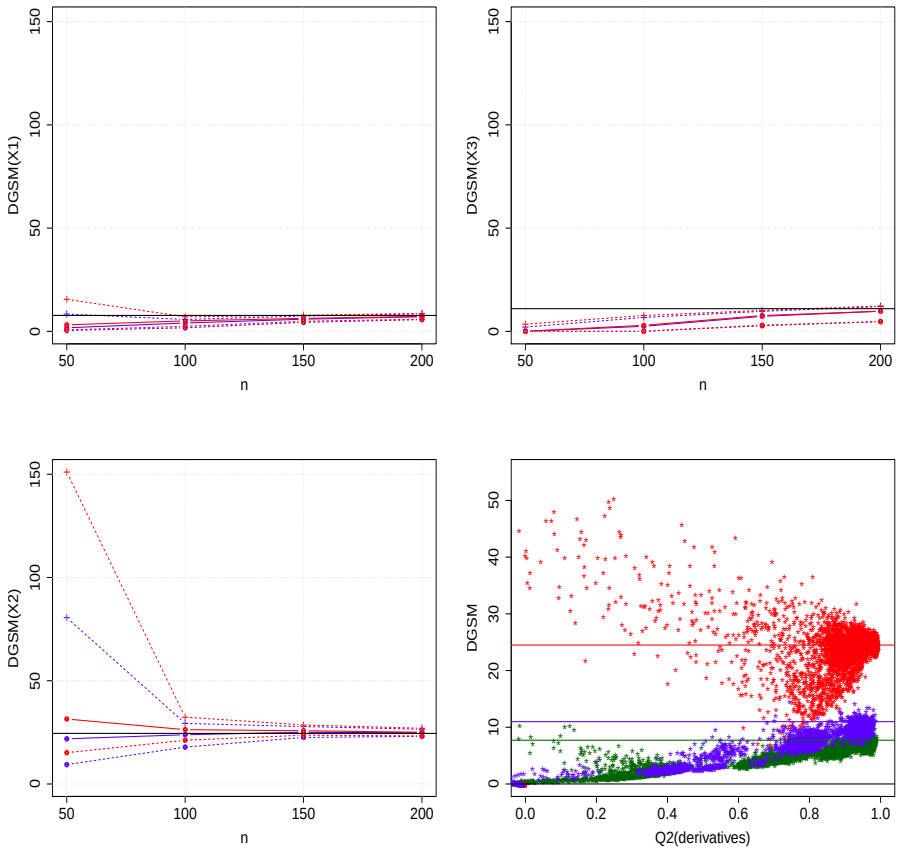


Figure 2: Evolution of the plug-in (blue) and full-GPM (red) DGSM estimators associated to the Ishigami function, in function of the number of observations n (three first figures). The solid lines represent the mean while the dashed ones represent the 2.5% and 97.5% quantiles of the Monte-Carlo distribution. The last picture represents the plug-in DGSM estimations associated to X_1 (green), X_2 (red) and X_3 (blue) in function of the accuracy criterion Q^2 of their associated partial derivatives.

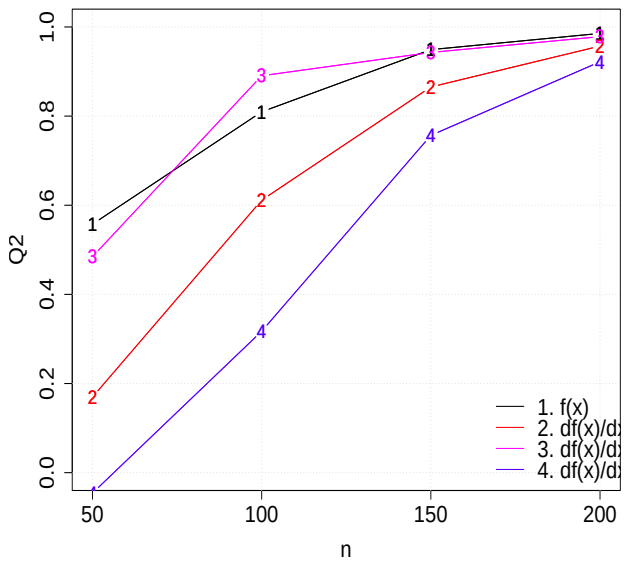


Figure 3: Evolution of the fitting accuracy of the Ishigami function and its partial derivatives according to the number of observations.

other ones. But this method is much more expensive from a CPU time point of view and would not be used in practice afterwards.

In this study, for different sample sizes n, we compute N instances of the stochastic DGSM $\hat{D}_3^{(2)}(\omega)$ associated to X_3 according to the three Monte-Carlo methods and using $M = 10000$ simulations in the $\left(\hat{D}_{3,i}^{(2),M}\right)_{1\leq i\leq N}$ computation (see Equation (19)), with $n \in \{50, 100, 200\}$ and $N \in \{500, 1000, 5000, 10000\}$. From these instances, we obtain the full-GPM estimator $\hat{D}_3^{(2)}$ and the 2.5% and 97.5% quantiles of the $\hat{D}_3^{(2)}(\omega)$ law.

	$N = 500$				$N = 1000$				$N = 5000$				$N = 10000$			
Method	*0*	1	2	3	*0*	1	2	3	*0*	1	2	3	*0*	1	2	3
$n = 50$	*0.0123*	5.9	**0.9**	2.1	*0.0124*	6.8	**1.3**	2.0	*0.0118*	14.4	4.6	**1.9**	*0.0121*	24.1	8.7	**1.9**
$n = 100$	*0.0361*	6.1	**1.1**	2.2	*0.0339*	6.6	**1.4**	2.0	*0.0339*	14.5	4.7	**2.0**	*0.0344*	24.2	8.7	**2.0**
$n = 200$	*0.1343*	6.3	**1.3**	2.5	*0.1280*	6.8	**1.7**	2.3	*0.1284*	14.8	5.0	**2.3**	*0.130*	24.5	9.0	**2.3**

Table 3: Comparison of the mean computational time (in seconds) for the GPM-based Monte-Carlo estimators using the `mvrnorm` function (1), the Gibbs sampler (2) or the chi-square approximation (3). The method indexed by 0 is the analytical one for uniform input laws and Gaussian kernels.

After $R = 100$ repetitions of the previous steps, Table 3 presents the mean computational time associated to the three Monte-Carlo methods, for different sample sizes n and different Monte-Carlo lengths N. **Firstly, the CPU time is almost insensitive to n. Secondly, when N is small with respect to M, the propagative version of the Gibbs sampler is faster than the chi-square approximation. To the contrary, when N is of the same order as M, this second approach is quicker than the first one. More generally, the chi-square approximation is all the more rapid than M is small with respect to N.** In the specific case where the estimator is the only quantity of interest, that is to say when standard deviation or credibility interval are not required, the use of analytical formula is very advised for uniform input laws and Gaussian kernels, with a CPU time between 10 and 100 times lower than the best one obtained with Monte-Carlo methods.

From an accuracy point of view, the differences between the different estimator errors are negligible with respect to the surrogate model error and the choice of a particular Monte-Carlo method has to be made according to CPU time considerations rather than accuracy ones.

To conclude, the propagative version of the Gibbs sampler and the chi-square approximations presented in Sections 3.2.1 and 3.2.2 are accurate approximations of the reference method using `mvrnorm`. From a CPU time point of view, we advise the use of the the Gibbs sampler when N is small with respect to M, and otherwise the use of the chi-square approximation.

4.1.3 Comparison for a high-dimensional test case

We also compare both DGSM estimators in a high-dimensional context. For this, we consider the following function:

$$f(X) = \alpha \sum_{i=1}^{d} \left(X_i + \beta \sum_{j>i}^{d} \left(\frac{d+1-i}{d} \right)^2 X_i X_j \right)$$

where $\alpha = \sqrt{12} - 6\sqrt{0.1(d-1)}$ and $\beta = 12\sqrt{0.1(d-1)}$. This function is related to the one proposed by [32], with a decay term $\left(\frac{d+1-i}{d}\right)^2$ for the interaction between the input variables X_i and X_j. In our case, we consider $d = 20$ independent random inputs, identically distributed according to the uniform distribution on the interval $[0, 1]$. For the inputs $X_1, \ldots, X_9$, the associated DGSMs are respectively equal to 15%, 12%, 10%, 8%, 7%, 6%, 5%, 4% and 4% of the mean squared gradient component sum; for the other inputs, the contribution is lower than 3%. Then, we compare the plug-in and full-GPM estimators for different sample sizes $n \in \{50, 100, 200, 300\}$. More precisely, for a given input, we consider the errors of the scaled DGSM estimators $\left(\frac{\hat{D}_k - D_k}{\max_j D_j}\right)$ in Figure 4 and the errors of the DGSM estimator ratios $\frac{\hat{D}_k}{\sum_{j=1}^{d} \hat{D}_j} - \frac{D_k}{\sum_{j=1}^{d} D_j}$ in Figure 5. Firstly, we can see that the full-GPM estimators lead to better results in terms of the DGSM approximation (see Figure 4), especially for the most significant input parameters. It is interesting to notice that the full-GPM DGSM estimators are greater than the true DGSM values for the less significant variables while their are lower for the more significant ones, while the plug-in DGSM estimators underestimate the DGSMs whatever the level of significativity. Then, Figure 5 compares the estimated and theoretical ratios of the DGSMs and conclusions are clear: the mean plug-in estimators lead to the better results for all the input parameters, while the full-GPM estimators are biased, with a behavior similar to the one observed in Figure 4 when

the significativity level decreases. Conversely, the full-GPM estimator quality is less dependent to the sample, as we can see it through the boxplot widths. **To conclude, full-GPM and plug-in DGSMs seem to be complementary in terms of scaled estimation errors for the individual DGSMs and errors of their individual contribution to the mean squared gradient component sum.**

Remark 3. *Note that the partial derivatives are complicated to estimate from a GPM with a constant mean, because the model f is linear while the GPM predictor is a kind of radial basis function. Consequently, even if the GPM is robust for the approximation of f, with a Q^2 criterion close to 1 when the sample size is important, the function gradient can be not sufficiently well estimated. This can explain the 10% error occurring for the first input variables, even for a sample size n equal to 300.*

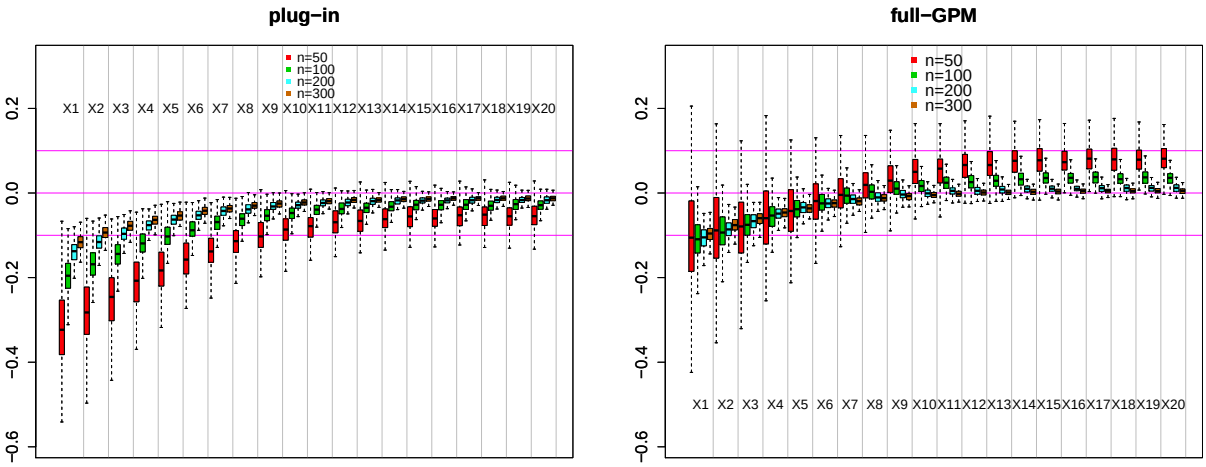


Figure 4: Errors of the DGSM estimators for different sample sizes.

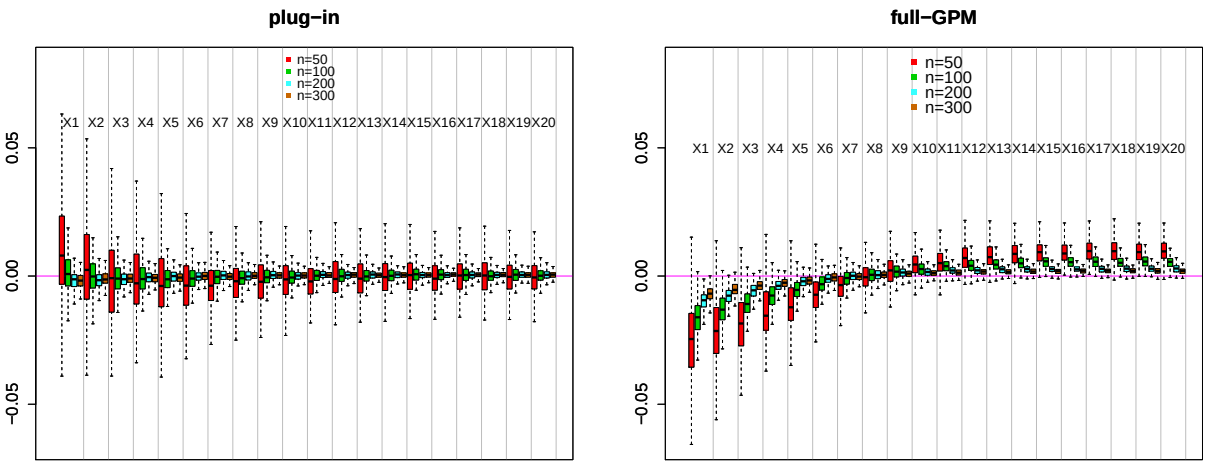


Figure 5: Errors of the DGSM estimator ratios for different sample sizes.

4.2 Application to an environmental problem

Finally, we apply the DGSMs defined by Expression (2) to an environmental problem developed by CEA (France) and Kurchatov Institute (Russia) and dealing with a radioactive waste temporary storage site close to Moscow. The main objective is to predict the transport of strontium 90 between 2002 and 2010, in order to determine the aquifer contamination. This transport in saturated porous media is modeled by the MARTHE computer code [50]. In

particular, this model returns in output the strontium 90 concentration predicted for year 2010 in many piezometers located on the waster repository site. In this study, we focus on this concentration for a piezometer of interest. This output is function of 20 uncertain input parameters representing the permeability of different geological layers, longitudinal dispersivity coefficients, transverse dispersivity coefficients, sorption coefficients, porosity and meteoric water infiltration intensities. [28] ran a global sensitivity analysis from the more influential input variables, estimating the Sobol' indices with a Gaussian process metamodel. In the same way, we estimate the DGSMs from a GPM and from the tools developed in this paper, considering the whole set of input parameters.

Procedure

Concretely, we build a GPM from the 300 observations used by [28] with a constant mean and a stationary tensorized covariance based on Gaussian kernels. Its accuracy is measured with the prediction coefficient Q^2 estimated by leave-one-out cross-validation [15]. Because of the high number of input parameters, we build this surrogate model iteratively using the efficient methodology of [29].

For the global sensitivity analysis, 17 out of 20 input parameters follow uniform distributions and the 3 remaining ones follow Weibull distributions, as mentioned in Table 4. So, we can not use the analytical formula of the DGSM estimators dedicated to uniform laws. Instead, we estimate them using the chi-square approximation approach with $M = 1000$.

Remark 4. *Note that in the initial test case [50], some factors are dependent: d1-dt1, d2-dt2, d2-dt3 and i1-i2-i3; this is not a problem for the DGSMs because the definition of these sensitivity indices is not based on a independence assumption, contrarily to the Sobol' indices for example. Furthermore, the DGSM estimation using the chi-square approximation method can deal with dependent input parameters, contrarily to the DGSM estimators $\hat{D}^{(1)}$ and $\hat{D}^{(2)}$. However, the interpretation of the results is more complicated in presence of dependent input variables and is beyond the remit of this study. Consequently, we replace (dt1, dt2, dt3) by the ratios (dt1/d1, dt2/d2, dt3/d3), which are independent of the other input parameters, and we consider the variations of infiltration coefficients $(i_1, \delta i_2, \delta i_3)$ between consecutive layers rather than their values. These modifications are mentioned by a bold italic index in Table 4.*

Results

The GPM built with the learning sample has a $Q^2 = 0.97$ using a leave-one-out cross-validation. Then we compute $N = 10000$ instances of the weighted stochastic DGSM $\widehat{R\Delta}_k^{(2)}(\omega)$ to obtain the mean $\hat{\Delta}_k^{(2)}$, which is the estimator of Δ_k, the 2.5% and 97.5% quantiles and the p-value associated to the significance test described in Section 3.3. These quantities are gathered in Table 5. Firstly, the DGSM estimations lead to the conclusion that the variable `kd1` explains 88% of the global output uncertainty while the δ`i3` contribution is equal to 7% and `per1` and `perz2` to around 2%. Secondly, the 95% intervals support these conclusion. Finally, we compute statistical tests on the nullity of these DGSMs and sort the input parameters according to their p-values. With this approach, `kd1`, δ`i3`, `per1`, `perz2`, `per3` and `dt1/d1` are the most significant variables while `kd2` and `kd3` can be fixed to nominal values if we consider a level $\alpha = 5\%$. In the case of a level $\alpha = 1\%$, the input parameters `dt3/d3`, `poros`, `dt2/d2`, `d2`, `perz4` and `i1` can also be fixed to nominal values, what remove nine uncertainty sources in the Marthe model and consequently reduce its complexity.

5 Conclusion

This paper deals with the estimation of the derivative-based global sensitivity measures (DGSMs) associated to a numerical simulator in global sensitivity analysis (GSA) [43]. This work falls within the scenario where this model does not provide the adjoint code to compute its output gradient. Under this consideration, we replaced this simulator by a surrogate model, more precisely by a Gaussian process metamodel (GPM) [34]. This idea has already been investigated with other surrogate models or sensitivity measures [46, 28, 47].

We proposed two kinds of GPM-based DGSM estimators. A first-one consists of substituting the numerical model by the GPM predictor; we called it "plug-in estimator". A second one uses the full GPM and we showed that this DGSM estimator is equal to the plug-in one, completed by an additive variance term; we called it "full-GPM estimator". This second estimator is the expectation of a stochastic DGSM where the numerical model is replaced by the whole GPM, not only its mean contrarily to the plug-in estimator; in this way, it takes into account the metamodel error, contrarily to the plug-in one. Consequently, we can also approach the distribution of this stochastic DGSM, build many statistics and more particularly provide a credibility interval for the DGSM estimator. Using this full-GPM estimator, we also proposed a significance test for eliminating the non-influential variables in a screening

	Input parameters	Notation	Distribution type	Distribution parameters[1]
1	Hydraulic conductivity layer 1	per1	Uniform	1, 15
2	Hydraulic conductivity layer 2	per2	Uniform	5, 0
3	Hydraulic conductivity layer 3	per3	Uniform	1, 15
4	Hydraulic conductivity zone 1	perz1	Uniform	1, 15
5	Hydraulic conductivity zone 2	perz2	Uniform	1, 15
6	Hydraulic conductivity zone 3	perz3	Uniform	1, 15
7	Hydraulic conductivity zone 4	perz4	Uniform	1, 15
8	Longitudinal dispersivity layer 1	d1	Uniform	0.05, 2
9	Longitudinal dispersivity layer 2	d2	Uniform	0.05, 2
10	Longitudinal dispersivity layer 3	d3	Uniform	0.05, 2
	Transversal dispersivity layer 1	dt1	Uniform	0.01× d1, 0.1× d1
11	Ratio of transversal to longitudinal dispersivity layer 1	dt1/d1	Uniform	0.01, 0.1
	Transversal dispersivity layer 2	dt2	Uniform	0.01× d2, 0.1× d2
12	Ratio of transversal to longitudinal dispersivity layer 2	dt2/d2	Uniform	0.01, 0.1
	Transversal dispersivity layer 3	dt3	Uniform	0.01× d3, 0.1× d3
13	Ratio of transversal to longitudinal dispersivity layer 3	dt3/d3	Uniform	0.01, 0.1
14	Volumetric distribution coefficient I.1	kd1	Weibull	1.1597, 19.9875
15	Volumetric distribution coefficient I.2	kd2	Weibull	0.891597, 24.4455
16	Volumetric distribution coefficient I.3	kd3	Weibull	1.27363, 22.4986
17	Porosity	poros	Uniform	0.3, 0.37
18	Infiltration type 1	i1	Uniform	0, 0.0001
	Infiltration type 2	i2	Uniform	i1, 0.01
19	Infiltration type 2 - type 1	δi2	Uniform	0, 0.0092
	Infiltration type 3	i3	Uniform	i2, 0.1
20	Infiltration type 3 - type 2	δi3	Uniform	0, 0.096

Table 4: Input parameter description for the Marthe test case.

X_k	Weighted stochastic DGSM $\widehat{RC}_k^{(2)}(\omega)$							Sobol Total
	Mean $\hat{\Delta}_k^{(2)} = C_k \hat{D}_k^{(2)}$		Standard deviation	Quantiles 2.5%	Quantiles 97.5%	Screening using $p_{\text{val},k}$		
kd1	6.5e-2	(88.0%)	2.4e-3	6.0e-2	6.9e-2	0.000	***	**0.75**
δi3	4.7e-3	(6.4%)	2.5e-4	4.2e-3	5.2e-3	0.000	***	**0.18**
per1	1.1e-3	(1.6%)	1.1e-4	9.4e-4	1.4e-3	0.000	***	**0.03**
perz2	1.0e-0	(1.4%)	1.3e-4	8.0e-4	1.3e-4	0.000	***	0.02
per3	2.1e-4	(0.3%)	3.4e-5	1.5e-4	2.8e-4	0.000	***	0.00
dt1/d1	1.4e-4	(0.2%)	2.3e-5	1.0e-4	1.9e-4	0.000	***	0.00
perz1	1.4e-4	(0.2%)	2.3e-5	1.0e-4	1.9e-4	0.001	**	0.00
perz3	1.4e-4	(0.2%)	2.1e-5	1.0e-4	1.9e-4	0.001	**	0.00
d3	1.4e-4	(0.2%)	2.1e-5	1.0e-4	1.8e-4	0.001	**	0.00
d1	1.3e-4	(0.2%)	2.0e-5	9.4e-5	1.7e-4	0.001	**	0.00
per2	1.3e-4	(0.2%)	2.0e-5	9.1e-5	1.6e-4	0.005	**	0.00
dt3/d3	1.2e-4	(0.2%)	1.8e-5	9.1e-5	1.6e-4	0.018	*	0.00
poros	1.2e-4	(0.2%)	1.8e-5	8.9e-5	1.6e-4	0.026	*	0.00
dt2/d2	1.2e-4	(0.2%)	1.7e-5	9.1e-5	1.6e-4	0.017	*	0.00
d2	1.2e-04	(0.2%)	1.7e-5	9.0e-5	1.6e-4	0.027	*	0.00
perz4	1.2e-4	(0.2%)	1.8e-5	8.8e-5	1.6e-4	0.018	*	0.00
i1	1.2-4	(0.2%)	1.6e-5	8.9e-5	1.5e-4	0.019	*	0.00
δi2	1.1e-4	(0.2%)	1.7e-5	8.3e-5	1.5e-4	0.001	**	0.00
kd2	5.6e-5	(0.1%)	1.0e-5	4.1e-5	8.1e-5	0.140		0.00
kd3	4.8e-5	(0.1%)	8.2e-6	3.5e-5	6.7e-5	0.180		0.00

Table 5: DGSM statistics for the MARTHE test case using the weighted DGSM formulation.

context. For all these reasons, the full-GPM is more relevant than the plug-in one, especially in presence of the small learning sample.

These estimators require many integral evaluations which can be computed by Monte-Carlo sampling. To this end, we proposed efficient methods based either on a propagative version of the Gibbs sampler, or on a chi-square approximation. We advise the first approach when the number of Monte-Carlo replications is small with respect to the number of GPM simulations required to the approximation of a stochastic DGSM instance, and conversely, and vice versa. Both methods can be used for the significance test computation. Furthermore, we showed that these plug-in and full-GPM estimators are analytically tractable when the input parameters are uniformly distributed and when the kernel functions of the GPM are Gaussian, considerably reducing the computational time.

Finally, we ran numerical studies with the Ishigami function and a high-dimensional one, and consider an industrial application with a complex hydrogeological computer code. For the first point, the convergence rates of the DGSM estimators differ from an input parameter to another, which can be explained by the accuracy differences between the GPM partial derivatives. This is due to the fact that the GPM maximizes the likelihood on a set of computer code output observations, not on the corresponding partial derivative values. In other words, the metamodel is built without any idea of the behavior of the model gradient. However, we showed that, from a few observations, the estimators using the full-GPM can be used in a screening context to rank the input parameters according to their level of influence on the model output, and provide an idea of their relative significance. The results are also convenient for the high-dimensional function approximated by a GPM built from a few simulations. For the industrial application, the results obtained with DGSMs weighted by the input variances give the same input parameter order as the one obtained with Sobol' indices [28] and the significant test lead to the same conclusion about the influential variables. This reinforces the pertinence of these DGSM estimators for screening purposes.

To conclude, the GPM-based DGSM estimators developed in this paper are encouraging for global sensitivity analysis and for screening from a few evaluations of the numerical simulator. A criticism of this approach is that these estimators consider the gradient of the GPM which has been built using observations of the simulator output, not of its gradient. Consequently, nothing guarantees that the partial derivatives of the GPM are accurate approximations of the associated partial derivative of the simulator. For future research, adaptive sampling dedicated to the gradient approximation enhancement could be investigated: a strategy could be to add, as a new code simulation, the input parameter vector maximizing the trace of the GPM gradient covariance or the reduction of this trace. Moreover, the GPM-based DGSM estimations should be improved if some observations of the model output partial derivatives were available or if the prior mean of the Gaussian process was function of the input parameters. In the case where the adjoint code is present, developments could be done about the integration of some model gradient simulations completing the output evaluations. Finally, it would be relevant to connect the DGSMs with the active subspace methodology [4], an emerging domain of computer experiments dealing with the identification of the main directions in the input parameter space. These directions are the first eigenvectors of the eigenvalue decomposition applied to the matrix composed of the pairwise products of the numerical simulator partial derivatives.

Acknowledgements

We are grateful to Béatrice Laurent for helpful discussions.

References

[1] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Scientific Computing, 16 (1995), pp. 1190–1208.

[2] F. Campolongo, J. Cariboni, and A. Saltelli, *An effective screening design for sensitivity analysis of large models*, Environmental Modelling & Software, 22 (2007), pp. 1509 – 1518. Modelling, computer-assisted simulations, and mapping of dangerous phenomena for hazard assessment.

[3] W. Chen, R. Jin, and A. Sudjianto, *Analytical metamodel-based global sensitivity analysis and uncertainty propagation for robust design*, Journal of Mechanical Design, (2005), pp. 875–886.

[4] P. G. Constantine, *Active subspaces: emerging ides for dimension recution in parameter studies*, Spotlights, SIAM, 2015.

[5] R Cools and D Nuyens, eds., *Derivative-based global sensitivity measures and their link with Sobol' sensitivity indices*, vol. 5805 of Proceedings of the Eleventh International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC 2014), Leuven, Belgium, 2015, Springer-Verlag.

[6] S. Da Veiga, *Global sensitivity analysis with dependence measures*, Journal of Statistical Computation and Simulation, 85 (2015), pp. 1283–1305.

[7] M. De Lozzo and A. Marrel, *New improvements in the use of dependence measures for sensitivity analysis and screening*. Submitted, hal-01090475, 2015.

[8] E. de Rocquigny, N. Devictor, and S. Tarantola, *Uncertainty in industrial practice*, Wiley, 2008.

[9] G. Dreyfus, *Neural networks - methodology and applications.*, Springer, 2005.

[10] B. Efron and C. Stein, *The jackknife estimate of variance*, The Annals of Statistics, 9 (1981), pp. 586–596.

[11] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, 1 ed., Sept. 2008.

[12] J.-C. Fort, T. Klein, and N. Rachdi, *New sensitivity analysis subordinated to a contrast*, Communications in Statistics - Theory and Methods, (2014). In press.

[13] S. Geman and D. Geman, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-6 (1984), pp. 721–741.

[14] G. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, *Measuring statistical dependence with hilbert-schmidt norms*, in Proceedings Algorithmic Learning Theory, Springer-Verlag, 2005, pp. 63–77.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

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

[17] B. Iooss and P. Lemaître, *A review on global sensitivity analysis methods*, in Uncertainty management in Simulation-Optimization of Complex Systems: Algorithms and Applications, C. Meloni and G. Dellino, eds., Springer, 2015.

[18] B. Iooss, A-L. Popelin, G. Blatman, C. Ciric, F. Gamboa, S. Lacaze, and M. lamboni, *Some new insights in derivative-based global sensitivity measures*, in Proceedings of SPAM11 & ESREL 2012 Conference, ESRA, 2012.

[19] A. Janon, M. Nodet, and C. Prieur, *Uncertainties assessment in global sensitivity indices estimation from metamodels*, International Journal for Uncertainty Quantification, 4 (2014).

[20] F. Kaitai, L. Runze, and S. Agus, *Design and modeling for computer experiments*, Computer science and data analysis series, Chapman & Hall/CRC, 2006.

[21] A. Kiparissides, S. S. Kucherenko, A. Mantalaris, and E. N. Pistikopoulos, *Global Sensitivity Analysis Challenges in Biological Systems Modeling*, Industrial & Engineering Chemistry Research, 48 (2009), pp. 7168–7180.

[22] J. P.C. Kleijnen, *Sensitivity analysis and related analyses: A review of some statistical techniques*, Journal of Statistical Computation and Simulation, 57 (1997), pp. 111–142.

[23] S. Kucherenko, M. Rodriguez-Fernandez, C. Pantelides, and N. Shah, *Monte carlo evaluation of derivative-based global sensitivity measures*, Reliability Engineering & System Safety, 94 (2009), pp. 1135 – 1148. Special Issue on Sensitivity Analysis.

[24] M. Lamboni, B. Iooss, A.-L. Popelin, and F. Gamboa, *Derivative-based global sensitivity measures: General links with sobol' indices and numerical tests*, Mathematics and Computers in Simulation, 87 (2013), pp. 45–54.

[25] L. Lantuéjoul and N. Desassis, *Simulation of a gaussian random vector: A propagative version of the gibbs sampler*, in Ninth International Geostatistics Congress, June 11. - 15., 2012, Oslo, Norway, 2012.

[26] L. Le Gratiet, C. Cannamela, and B. Iooss, *A bayesian approach for global sensitivity analysis of (multifidelity) computer codes*, SIAM/ASA Journal on Uncertainty Quantification, 2 (2014), pp. 336–363.

[27] H. Liu, Tangn Y., and H. H. Zhang, *A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables*, Computational Statistics & Data Analysis, 53 (2009), pp. 853 – 856.

[28] A. Marrel, B. Iooss, B. Laurent, and O. Roustant, *Calculations of sobol indices for the gaussian process metamodel*, Reliability Engineering & System Safety, 94 (2009), pp. 742–751.

[29] A. Marrel, B. Iooss, F. Van Dorpe, and E. Volkova, *An efficient methodology for modeling complex computer codes with Gaussian processes*, Computational Statistics & Data Analysis, 52 (2008), pp. 4731–4744.

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

[31] M. D. Morris, *Factorial sampling plans for preliminary computational experiments*, Technometrics, 33 (1991), pp. 161–174.

[32] M. D. Morris, L. M. Moore, and M. D. McKay, *Sampling plans based on balanced incomplete block designs for evaluating the importance of computer model inputs*, Journal of Statistical Planning and Inference, 136 (2006), pp. 3203–3220.

[33] E. Plischke, E. Borgonovo, and C. L. Smith, *Global sensitivity measures from given data*, European Journal of Operational Research, 226 (2013), pp. 536 – 550.

[34] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[35] O. Roustant, J. Fruth, B. Iooss, and S. Kuhnt, *Crossed-derivative based sensitivity measures for interaction screening*, Mathematics and Computers in Simulation, 105 (2014), pp. 105 – 118.

[36] R. Roustant, D. Ginsbourger, and Y. Deville, *DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization*, Journal of Statistical Software, 51 (2012), pp. 1–55.

[37] J. Ruichen, C. Wei, and A. Agus, *An efficient algorithm for constructing optimal design of computer experiments*, Journal of Statistical Planning and Inference, 134 (2005), pp. 268–287.

[38] A Saltelli, K Chan, and E M Scott, *Sensitivity analysis*, Wiley, New York, NY, 2000.

[39] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*, John Wiley, 2008.

[40] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity analysis in practice: a guide to assessing scientific models*, John Wiley & Sons, 2004.

[41] T. J. Santner, B. J. Williams, and W. Notz, *The design and analysis of computer experiments*, Springer series in statistics, Springer, New York, 2003.

[42] I. Sobol and A. Gersham, *On an alternative global sensitivity estimator*, in Proceedings of SAMO 1995, Belgirate, 1995, pp. 40–42.

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

[44] I. M. Sobol, *Sensitivity estimates for nonlinear mathematical models*, MMCE, 1 (1993), pp. 407–414.

[45] C. Soize and R. Ghanem, *Physical systems with random uncertainties: Chaos representations with arbitrary probability measure*, SIAM J. Scientific Computing, 26 (2004), pp. 395–410.

[46] B. Sudret, *Global sensitivity analysis using polynomial chaos expansions*, Reliability Engineering & System Safety, 93 (2008), pp. 964 – 979. Bayesian Networks in Dependability.

[47] B. Sudret and C. V. Mai, *Computing derivative-based global sensitivity measures using polynomial chaos expansions*, Reliability Engineering & System Safety, 134 (2015), pp. 241–250.

[48] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, *Measuring and testing dependence by correlation of distances*, The Annals of Statistics, 35 (2007), pp. 2769–2794.

[49] S. Touzani and D. Busby, *Screening method using the derivative-based global sensitivity indices with application to reservoir simulator*, Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles, (2014).

[50] E. Volkova, B. Iooss, and F. Van Dorpe, *Global sensitivity analysis for a numerical model of radionuclide migration from the rrc kurchatov institute radwaste disposal site*, Stochastic Environmental Research and Risk Assessment, 22 (2008), pp. 17–31.

Appendix: Mathematical details

Computation of $\hat{D}_k^{(2)}$

From the Gaussian process metamodel and the definition $\hat{D}_k^{(2)} = \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |\omega\right]\right]$, we obtain the relation (13):

$$\begin{aligned}
\hat{D}_k^{(2)} &= \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |\omega\right]\right] = \mathbb{E}\left[\mathbb{E}\left[\left(\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right)^2 |X\right]\right] \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|\omega\right]\right)^2 + \mathbb{V}_\omega\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}\right]\right] \\
&= \mathbb{E}\left[\left(\frac{\partial \hat{f}(X)}{\partial x_k}\right)^2\right] + \mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right] \\
&= \hat{D}_k^{(1)} + \mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right].
\end{aligned}$$

Then we develop the second term $\mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right]$ and get the result (14):

$$\begin{aligned}
\mathbb{E}\left[\mathbb{V}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right] &= \mathbb{E}\left[\mathrm{Cov}\left[\frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}, \frac{\partial Z_{\mathsf{C}}(X;\omega)}{\partial x_k}|X\right]\right] \\
&= \mathbb{E}\left[\frac{\partial^2}{\partial u_k \partial v_k}\mathrm{Cov}_\omega\left[Z_{\mathsf{C}}(U;\omega), Z_{\mathsf{C}}(V;\omega)\right]\Big|_{(U,V):=(X,X)}\right] \\
&= \mathbb{E}\left[\frac{\partial^2}{\partial u_k \partial v_k}\left(\sigma^2 r(U,V) - \sigma^2 k(U)^T R^{-1} k(V)\right)\Big|_{(U,V):=(X,X)}\right] \\
&= \sigma^2 \mathbb{E}\left[\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\Big|_{(U,V):=(X,X)}\right] - \sigma^2 \mathbb{E}\left[\frac{\partial k(U)^T}{\partial u_k} R^{-1} \frac{\partial k(V)}{\partial v_k}\Big|_{(U,V):=(X,X)}\right] \\
&= \sigma^2 \mathbb{E}\left[\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\Big|_{(U,V):=(X,X)}\right] \\
&- \sigma^2 \sum_{i,j=1}^{n} \left(R^{-1}\right)_{ij} \mathbb{E}_X\left[\frac{\partial r(X, X^{(i)})}{\partial x_k}\frac{\partial r(X, X^{(j)})}{\partial x_k}\right] \\
&= \sigma^2 \mathbb{E}\left[\frac{\partial^2 r(U,V)}{\partial u_k \partial v_k}\Big|_{(U,V):=(X,X)}\right] \\
&- \sigma^2 \sum_{i,j=1}^{n} \left(R^{-1}\right)_{ij} \mathbb{E}\left[\frac{\partial r_k(X_k, X_k^{(i)})}{\partial x_k}\frac{\partial r_k(X_k, X_k^{(j)})}{\partial x_k}\right] \\
&\times \prod_{\substack{l=1 \\ l\neq k}}^{d} \mathbb{E}\left[r_l(X_l, X_l^{(i)}) r_l(X_l, X_l^{(j)})\right].
\end{aligned}$$

Gaussian kernel and uniform laws

Lemma 1. *Let $X \sim \mathcal{U}([m,M])$ and $r(x,x') = \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$. Then*

$$\mathbb{E}\left[r(X,y)r(X,z)\right] = \sqrt{\pi}\theta(M-m)^{-1}\exp\left(-\frac{(y-z)^2}{4\theta^2}\right)\mathbb{P}\left[Z \in [m,M]\right]$$

where $Z \sim \mathcal{N}\left(\frac{y+z}{2}; \frac{\theta^2}{2}\right)$.

Proof.

$$\begin{aligned}
\mathbb{E}\left[r(X,y)r(X,z)\right] &= \mathbb{E}\left[\exp\left(-\frac{2X^2-2X(y+z)+y^2+z^2}{2\theta^2}\right)\right] \\
&= \mathbb{E}\left[\exp\left(-\frac{(X-\frac{y+z}{2})^2+\frac{1}{4}(y-z)^2}{2(\frac{\theta}{\sqrt{2}})^2}\right)\right] \\
&= \sqrt{\pi}\theta\exp\left(-\frac{(y-z)^2}{4\theta^2}\right)\mathbb{E}\left[\frac{1}{\sqrt{2\pi\frac{\theta^2}{2}}}\exp\left(-\frac{(X-\frac{y+z}{2})^2}{2(\frac{\theta}{\sqrt{2}})^2}\right)\right] \\
&= \sqrt{\pi}\theta\exp\left(-\frac{(y-z)^2}{4\theta^2}\right)(M-m)^{-1}\int_m^M \varphi_{\mathcal{N}\left(\frac{y+z}{2};\frac{\theta^2}{2}\right)}(x)dx
\end{aligned}$$

where $\varphi_{\mathcal{N}(m;\sigma^2)}(x)$ is the probability density function of the normal law with mean m and variance σ^2. □

Lemma 2. *Let $X \sim \mathcal{U}([m,M])$ and $r(x,x') = \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$. Then*

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial r(X,y)}{\partial x}\frac{\partial r(X,z)}{\partial x}\right] &= -\frac{(y-z)^2}{4\theta^4}\mathbb{E}\left[r(X,y)r(X,z)\right] \\
&+ \frac{\sqrt{\pi}}{2\theta(M-m)}\exp\left(-\frac{(y-z)^2}{4\theta^2}\right)\left\{a\varphi(a)-b\varphi(b)-\mathbb{P}\left[Z\in[a,b]\right]\right\}
\end{aligned}$$

where $Z \sim \mathcal{N}(0,1)$, $a = \frac{2m-y-z}{\sqrt{2}\theta}$, $b = \frac{2M-y-z}{\sqrt{2}\theta}$ and where $\varphi(.)$ is the probability density function of the standard normal law.

Proof.

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial r(X,y)}{\partial x}\frac{\partial r(X,z)}{\partial x}\right] &= \mathbb{E}\left[\frac{(X-y)(X-z)}{\theta^4}r(X,y)r(X,z)\right] \\
&= \mathbb{E}\left[\left(\frac{(X-\frac{y+z}{2})^2}{\theta^4}-\frac{1}{4}\frac{(y-z)^2}{\theta^4}\right)r(X,y)r(X,z)\right] \\
&= \mathbb{E}\left[\frac{(X-\frac{y+z}{2})^2}{\theta^4}r(X,y)r(X,z)\right]-\frac{1}{4}\frac{(y-z)^2}{\theta^4}\mathbb{E}\left[r(X,y)r(X,z)\right]
\end{aligned}$$

where

$$\mathbb{E}\left[\frac{(X-\frac{y+z}{2})^2}{\theta^4}r(X,y)r(X,z)\right] = \sqrt{\pi}\theta\exp\left(-\frac{(y-z)^2}{4\theta^2}\right)\underbrace{\mathbb{E}\left[\frac{(X-\frac{y+z}{2})^2}{\theta^4}\varphi_{\mathcal{N}(\frac{y+z}{2},\frac{\theta^2}{2})}(X)\right]}_{\Delta}$$

with, using $w = (x-\frac{y+z}{2})/(\theta/\sqrt{2})$,

$$\begin{aligned}
\Delta &= (M-m)^{-1}\int_m^M \frac{(x-\frac{y+z}{2})^2}{\theta^4}\frac{1}{\sqrt{2\pi}\frac{\theta}{\sqrt{2}}}\exp\left(-\frac{(X-\frac{y+z}{2})^2}{2(\frac{\theta}{\sqrt{2}})^2}\right)dx \\
&= (M-m)^{-1}\int_a^b \frac{w^2}{2\theta^2}\varphi(w)dw \quad \text{with } a = \frac{2m-y-z}{\sqrt{2}\theta} \text{ and } b = \frac{2M-y-z}{\sqrt{2}\theta} \\
&= \frac{1}{2\theta^2(M-m)}\int_b^a w\varphi'(w)dw \\
&= \frac{1}{2\theta^2(M-m)}\left(a\varphi(a)-b\varphi(b)-\int_b^a \varphi(w)dw\right).
\end{aligned}$$

□