SIAM/ASA J. Uncertainty Quantification
Vol. 2, pp. 336–363

A Bayesian Approach for Global Sensitivity Analysis of (Multifidelity) Computer Codes*

Loic Le Gratiet†, Claire Cannamela‡, and Bertrand Iooss§

Abstract. Complex computer codes are widely used in science and engineering to model physical phenomena. Global sensitivity analysis aims to identify the input parameters which have the most important impact on the code output. Sobol indices are a popular tool for performing such analysis. However, their estimates require an important number of simulations and often cannot be processed under reasonable time constraint. To handle this problem, a Gaussian process regression model is built to approximate the computer code output and the Sobol indices are estimated through it. The aim of this paper is to provide a methodology for estimating the Sobol indices through a surrogate model taking into account both the estimation errors and the surrogate model errors. In particular, it allows us to derive nonasymptotic confidence intervals for the Sobol index estimates. Furthermore, we extend the suggested strategy to the case of multifidelity computer codes which can be run at different levels of accuracy. For such simulators, we use an extension of Gaussian process regression models for multivariate outputs.

Key words. sensitivity analysis, Gaussian process regression, multifidelity model, complex computer codes, Sobol index, Bayesian analysis

AMS subject classifications. 62G99, 60G25

DOI. 10.1137/130926869

1. Introduction.

In complex computer codes analysis, we commonly want to measure the importance of input parameters on the model response. In this work, we focus on the Sobol indices (see [32], [30], and [33]), which are variance-based importance measures coming from the Hoeffding–Sobol decomposition [12]. We note that the presented sensitivity analysis holds when the input parameters are independent. For an analysis with dependent inputs, the reader is referred to the articles [20] and [4].

A widely used method for estimating the Sobol indices are the Monte Carlo–based methods. They allow for quantifying the errors due to numerical integration (with a bootstrap procedure in a nonasymptotic case (see [1] and [16]) or thanks to asymptotic normality properties in an asymptotic case [15]). However, the estimation of the Sobol indices by sampling methods requires a large number of simulations that are sometimes too costly and time-consuming. A popular method for overcoming this difficulty is to build a mathematical approximation of

*Received by the editors June 28, 2013; accepted for publication (in revised form) May 30, 2014; published electronically July 22, 2014.
http://www.siam.org/journals/juq/2/92686.html

†Université Paris Diderot, 75205 Paris Cedex 13, France, and CEA, DAM, DIF, F-91297 Arpajon, France (loic.legratiet@gmail.com).
‡CEA, DAM, DIF, F-91297 Arpajon, France (claire.cannamela@cea.fr).
§EDF R&D, 78401 Chatou, France, and Institut de Mathématiques de Toulouse, 31062 Toulouse, France (bertrand.iooss@edf.fr). The work of this author was funded by the French National Research Agency (ANR) through the COSINUS program (project COSTA BRAVA noANR-09-COSI-015) and by the CNRS NEEDS program through the ASINCRONE project.

the code input/output relation, also called a metamodel (see [24] and [13]).

In this paper we deal with the use of kriging and multifidelity cokriging models to approximate the computer code outputs. The reader is referred to the books [35], [31], and [28] for an overview of kriging methods for computer experiments. A pioneering article dealing with the kriging approach for performing global sensitivity analysis is that of Oakley and O'Hagan [25]. Their method is also investigated in [22]. The strength of the suggested approach is that it allows for inferring from the surrogate model uncertainty about the Sobol index estimates. However, it does not use Monte Carlo integration and it does not take into account the numerical errors due to the numerical integration. Another flaw of the method presented in [25] and [22] is that it does not consider the joint distribution of the numerator and the denominator of the Sobol indices. We note that a bootstrap procedure can also be used to evaluate the impact of the surrogate model uncertainty on the Sobol index estimates as presented in [36]. However, this approach focuses only on the parameter estimation errors.

On the other hand, a method giving confidence intervals for the Sobol index estimates and taking into account both the metamodel uncertainty and the numerical integration errors is suggested in [16]. They consider Monte Carlo integration to estimate the Sobol indices (see [32], [34], and [15]) instead of numerical integration, and they infer from the sampling errors thanks to a bootstrap procedure. Furthermore, to deal with the metamodel error, they consider an upper bound on it. In the kriging case they use the kriging variance up to a multiplicative constant as the upper bound. Nevertheless, this is a rough upper bound which considers the worst error on a test sample. Furthermore, this method does not allow for inferring from the metamodel uncertainty about the Sobol index estimates.

In this paper we propose a method combining the approaches presented in [25] and [16]. As in [25] we consider the code as a realization of a Gaussian process. Furthermore, we use the method suggested in [16] to estimate the Sobol indices with Monte Carlo integration. Therefore, we can use the bootstrap method presented in [1] to infer from the sampling error on the Sobol index estimates. Furthermore, contrary to [25] and [22] we deal with the exact definition of Sobol indices. Consequently, we introduce nonasymptotic certified Sobol index estimates, i.e., with confidence intervals which take into account both the surrogate model error and the numerical integration errors. We note that the efficiency of the suggested approach is monitored by the prediction capability of the Gaussian process regression models. Performing a Gaussian process regression in high dimension could be an issue (see [26] and [23]), and this is not our focus in this paper. Therefore, we avoid it by considering applications in a dimension not relatively high (e.g., around 10).

Finally, the suggested approach is extended to a multifidelity cokriging model. It allows for approximating a computer code using fast and coarse versions of it. The suggested multifidelity model is derived from the original one proposed in [17]. We note that the use of cokriging models for dealing with multifidelity codes has largely been investigated during the past decade (see [18], [11], [29], and [27]). Furthermore, the efficiency of the method has been intensively highlighted on numerous applications (see, e.g., [9], [21], [3], and [10]). A definition of Sobol indices for multifidelity computer codes is presented in [14]. However, their approach is based on tabulated biases between fine and coarse codes and does not allow for inferring from the metamodel uncertainty. The cokriging model fixes this weakness since it allows for considering general forms for the biases and for inferring from the surrogate model error.

This paper is organized as follows. First, we introduce in section 2 the so-called Sobol indices. Then, we present in section 3 the kriging-based sensitivity analysis suggested by [25]. Our approach is developed in section 4. In particular, we give an important result allowing for effectively sampling with respect to the kriging predictive distribution in subsection 4.3. Finally, we extend in section 5.1 the presented approaches to multifidelity cokriging models. We highlight that we present in subsection 5.2 a method for sampling with respect to the multifidelity predictive distribution. In this case the predictive distribution is no longer Gaussian. Numerical tests are performed in section 6, and an industrial example is considered in section 7. A conclusion synthesizes this work in the last section.

2. Global sensitivity analysis: The method of Sobol.

We present in this section the method of Sobol for global sensitivity analysis [32]. It is inspired by the book [30] giving an overview of classical sensitivity analysis methods.

2.1. Sobol variance-based sensitivity analysis.

Let us consider the input parameter space $Q \subset \mathbb{R}^d$ such that $(Q, \mathcal{B}(Q))$ is a measurable product space of the form

$$(Q, \mathcal{B}(Q)) = (Q_1 \times \cdots \times Q_d, \mathcal{B}(Q_1 \times \cdots \times Q_d)),$$

where $\mathcal{B}$ is the Borelian σ-algebra and $Q_i \subset \mathbb{R}$ is a nonempty open set for $i = 1, \ldots, d$. Furthermore, we consider a probability measure μ on $(Q, \mathcal{B}(Q))$, with values in $\mathbb{R}$ and of the form

$$\mu(x) = \mu_1(x^1) \otimes \cdots \otimes \mu_d(x^d).$$

The Hoeffding–Sobol decomposition (see [12]) states that any function $z(x) \in L^2_\mu(\mathbb{R}^d)$ can be decomposed into summands of increasing dimensionality in such a way that ($L^2_\mu(\mathbb{R}^d)$ is the space of squared integrable functions with respect to the measure μ on $\mathbb{R}^d$):

$$z(x) = z_0 + \sum_{i=1}^{d} z_i(x^i) + \sum_{1 \le i < j \le k} z_{ij}(x^i, x^j) + \cdots + z_{1,2,\ldots,d}(x^1, \ldots, x^d) = \sum_{u \in \mathcal{P}} z_u(x^u), \tag{2.1}$$

where $\mathcal{P}$ is the collection of all subsets of $\{1, \ldots, d\}$ and x^u is a group of variables such that $x^u = (x^i)_{i \in u}$. Furthermore, the decomposition is unique if we consider the following property:

$$\int z_u(x^u)\, d\mu_i(x^i) = 0 \quad \forall i \in u,\ \forall u \in \mathcal{P}. \tag{2.2}$$

Now, let us suppose that the inputs are a random vector $X = (X^1, \ldots, X^d)$ defined on the probability space $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ and with measure μ. Sobol [32] showed that the decomposition (2.1) can be interpreted as conditional expectations as follows:

$$\begin{aligned}
z_0 &= \mathbb{E}_X\left[z(X)\right], \\
z_i(X^i) &= \mathbb{E}_X\left[z(X)|X^i\right] - z_0, \\
z_{ij}(X^i, X^j) &= \mathbb{E}_X\left[z(X)|X^i, X^j\right] - z_i(X^i) - z_j(X^j) - z_0, \\
&\vdots \\
z_u(X^u) &= \mathbb{E}_X\left[z(X)|X^u\right] - \sum_{v \subset u} z_v(X^v),
\end{aligned}$$

with $u \in \mathcal{P}$ and v a subset of u. From this scheme, we can naturally develop the variance-based sensitivity indices of Sobol. First, let us consider the total variance D of $z(x)$:

$$D = \mathrm{var}_X\left(z(X)\right). \tag{2.3}$$

By squaring and integrating the decomposition (2.1), we obtain [32]

$$D = \sum_{i=1}^{d} D_i + \sum_{1\le i<j\le d} D_{ij} + \cdots + D_{1,2,\ldots,d} = \sum_{u\in\mathcal{P}} D_u, \tag{2.4}$$

with $D_u = \mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^u\right]\right) - \sum_{v\subset u}\mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^v\right]\right)$. Finally, the Sobol sensitivity indices are given by

$$S_u = \frac{D_u}{D}, \tag{2.5}$$

where $u \in \mathcal{P}$. We note that we have the following useful equality, which allows for easily interpreting S_u as the part of variance of $z(x)$ due to x^u and not explained by x^v with $v \subset u$:

$$1 = \sum_{i=1}^{d} S_i + \sum_{1\le i<j\le d} S_{ij} + \cdots + S_{1,2,\ldots,d} = \sum_{u\in\mathcal{P}} S_u. \tag{2.6}$$

In particular, S_i is called the first-order sensitivity index for variable x^i. It measures the main effect of x^i on the output, i.e., the part of variance of $z(x)$ explained by the factor x^i. Furthermore, S_{ij} for $i \neq j$ is the second-order sensitivity index. It measures the part of variance of $z(x)$ due to x^i and x^j and not explained by the individual effects of x^i and x^j.

2.2. Monte Carlo–based estimation of Sobol indices. Now, let us denote $Q^{d_1} = Q_{i_1} \times \cdots \times Q_{i_{d_1}}$, $d_1 \le d$, $\{i_1, \ldots, i_{d_1}\} \in \mathcal{P}$, and $Q^{d_2} = Q_{j_1} \times \cdots \times Q_{j_{d_2}}$ such that $\{j_1, \ldots, j_{d_2}\} = \{1, \ldots, d\} \setminus \{i_1, \ldots, i_{d_1}\}$. Analogously, we use the notation $X^{d_1} = (X^i)_{i\in\{i_1,\ldots,i_{d_1}\}}$, $X^{d_2} = (X^j)_{j\in\{j_1,\ldots,j_{d_2}\}}$, $\mu^{d_1} = \left(\bigotimes_{i\in\{i_1,\ldots,i_{d_1}\}}\mu_i\right)$, and $\mu^{d_2} = \left(\bigotimes_{j\in\{j_1,\ldots,j_{d_2}\}}\mu_j\right)$, where μ^{d_1} and μ^{d_2} are probability measures on $(Q^{d_1}, \mathcal{B}(Q^{d_1}))$ and $(Q^{d_2}, \mathcal{B}(Q^{d_2}))$. Let us resort the input parameters such that $Q = Q^{d_1} \times Q^{d_2}$—e.g., for $d = 4$, let us consider $\{i_1, i_2\} = \{1, 4\}$ and $\{j_1, j_2\} = \{2, 3\}$; we have $Q^{d_1} = Q_1 \times Q_4$ and $Q^{d_2} = Q_2 \times Q_3$ and $Q = Q_1 \times Q_4 \times Q_2 \times Q_3$. Consequently, we have the equalities $\mu = \mu^{d_1} \otimes \mu^{d_2}$ and $X = (X^{d_1}, X^{d_2})$ with $d = d_1 + d_2$.

We are interested in evaluating the closed sensitivity index:

$$\mathcal{S}^{X^{d_1}} = \frac{V^{X^{d_1}}}{V} = \frac{\mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(z(X)\right)}. \tag{2.7}$$

A first method would be to use d-dimensional numerical integration to approximate the numerator and denominator of (2.7) as presented in [25] and [22]. Nonetheless, since d is large in general, this method leads to numerical issues and is computationally expensive. A second approach is to take advantage of the probabilistic interpretation of the Sobol indices and to use a Monte Carlo procedure to evaluate the different integrals as presented in the forthcoming developments.

Proposition 1 (see [32]). *Let us consider the random vectors* $(X, \tilde{X})$ *with* $X = (X^{d_1}, X^{d_2})$ *and* $\tilde{X} = (X^{d_1}, \tilde{X}^{d_2})$*, where* X^{d_1} *is a random vector with measure* μ^{d_1} *on* Q^{d_1}*, and* X^{d_2} *and* $\tilde{X}^{d_2}$ *are independent random vectors with measure* μ^{d_2} *on* Q^{d_2}*. We have the following equality:*

$$\operatorname{var}_X\left(\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right) = \operatorname{cov}_X\left(z(X), z(\tilde{X})\right). \tag{2.8}$$

$\mathcal{S}^{X^{d_1}}$ in (2.7) can thus be estimated by considering the random variables $(X_i, \tilde{X}_i)_{i=1,\dots,m}$, $m \in \mathbb{N}^*$, lying in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ such that $(X_i, \tilde{X}_i) \overset{\mathcal{L}}{=} (X, \tilde{X})$ ($\overset{\mathcal{L}}{=}$ stands for an equality in distribution) and $(X_i, \tilde{X}_i)$ are independent of $(X_j, \tilde{X}_j)$ for $i \neq j$ and by using an estimator for the covariance $\operatorname{cov}_X\big(z(X), z(\tilde{X})\big)$.

Following this principle, Sobol [32] suggests the following estimator for the ratio in (2.7):

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m z(X_i)\frac{1}{m}\sum_{i=1}^m z(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m z(X_i)\right)^2}. \tag{2.9}$$

This estimation is improved by Janon et al. [15], who propose the following estimator:

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \left(\frac{1}{2m}\sum_{i=1}^m z(X_i) + z(\tilde{X}_i)\right)^2}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{2m}\sum_{i=1}^m z(X_i) + z(\tilde{X}_i)\right)^2}. \tag{2.10}$$

In particular, they demonstrate that the asymptotic variance in (2.10) is better than that in (2.9) and they show that the estimator (2.10) is asymptotically efficient for the first-order indices. The main weakness of the estimators (2.9) and (2.10) is that they are sometimes not accurate for small values of $V^{X^{d_1}}/V$ in (2.7). To tackle this issue, Sobol et al. [34] propose the following estimator:

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{\tilde{X}}_i)}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m z(X_i)\right)^2}, \tag{2.11}$$

where $\tilde{\tilde{X}} = (\tilde{X}^{d_1}, \tilde{X}^{d_2})$, $\tilde{X}^{d_1} \overset{\mathcal{L}}{=} X^{d_1}$, $\tilde{X}^{d_1} \perp X^{d_1}$, and $(\tilde{\tilde{X}}_i)_{i=1,\dots,m}$ is such that $\tilde{\tilde{X}}_i \overset{\mathcal{L}}{=} \tilde{\tilde{X}}$ for all $i = 1, \dots, m$.

3. Kriging-based sensitivity analysis: A first approach.

We present in this section the approach suggested in [25] and [22] for performing global sensitivity analysis using kriging surrogate models. Then, we present an alternative method that allows for avoiding complex numerical integration. Nevertheless, we will see that this approach does not provide a correct representation of the Sobol indices. We handle this problem in the next section.

3.1. A short introduction to the kriging model.

The principle of the kriging model is to consider that our prior knowledge about the code $z(x)$ can be modeled by a Gaussian process $Z(x)$ with mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $\sigma^2 r(x, \tilde{x})$ (see, for example, [31]). Note that the correlation function $r(x, \tilde{x})$ usually depends on parameters. We do not make them explicit in order to clarify the notation. However, they are used for the application where we present

how to estimate them. Then, the code $z(x)$ is approximated by a Gaussian process $Z_n(x)$ having the predictive distribution $[Z(x)|Z(\mathbf{D}) = \mathbf{z}^n, \sigma^2]$, where $\mathbf{z}^n$ are the known values of $z(x)$ at points in the experimental design set $\mathbf{D} = \{x^1, \dots, x^n\}$, $x^i \in Q$, and σ^2 is the variance parameter. Therefore, we have

$$Z_n(x) \sim \mathrm{GP}\left(m_n(x), s_n^2(x, \tilde{x})\right), \tag{3.1}$$

where the mean $m_n(x)$ is given by

$$m_n(x) = \mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}}\right),$$

where $\mathbf{R} = [r(x_i, x_j)]_{i,j=1,\dots,n}$, $\mathbf{r}'(x) = [r(x, x_i)]_{i=1,\dots,n}$, $\mathbf{F} = [\mathbf{f}'(x_i)]_{i=1,\dots,n}$, and

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n.$$

The variance $s_n^2(x, \tilde{x})$ is given by

$$s_n^2(x, \tilde{x}) = \sigma^2\left(1 - \begin{pmatrix}\mathbf{f}'(x) & \mathbf{r}'(x)\end{pmatrix}\begin{pmatrix}0 & \mathbf{F}' \\ \mathbf{F} & \mathbf{R}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{f}(\tilde{x}) \\ \mathbf{r}(\tilde{x})\end{pmatrix}\right).$$

The variance parameter σ^2 can be estimated with a restricted maximum likelihood method, i.e., $\hat{\sigma}^2 = (\mathbf{z}^n - \hat{\boldsymbol{\beta}}\mathbf{F})'\mathbf{R}^{-1}(\mathbf{z}^n - \hat{\boldsymbol{\beta}}\mathbf{F})/(n-p)$, where p is the size of $\boldsymbol{\beta}$. For a Bayesian point of view, we integrate the posterior distribution of $\boldsymbol{\beta}$ obtained from the improper noninformative prior $p(\boldsymbol{\beta}) \propto 1$. Of course, informative prior distributions can be considered, but the predictive distribution will no longer be Gaussian (except for the case of Gaussian priors). Furthermore, for σ^2 we consider a plug-in approach; i.e., it is considered as known. These choices allow for having a Gaussian predictive distribution which can be easily sampled. We use this property in the forthcoming developments.

3.2. Kriging-based Sobol index. The idea suggested in [25] and [22] is to substitute $z(x)$ with $Z_n(x)$ in (2.7), which leads to

$$\mathcal{S}_n^{X^{d_1}} = \frac{V_n^{X^{d_1}}}{V_n} = \frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}. \tag{3.2}$$

Therefore, if we denote by $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ the probability space where the Gaussian process $Z(x)$ lies, then the estimator $\mathcal{S}_n^{X^{d_1}}$ lies in $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ (it is hence random). We note that $Z_n(X)$ is defined on the product probability space $(\Omega_X \times \Omega_Z, \sigma(\mathcal{F}_X \times \mathcal{F}_Z), \mathbb{P}_X \otimes \mathbb{P}_Z)$.

Nevertheless, the distribution of $\mathcal{S}_n^{X^{d_1}}$ is intractable and [25] and [22] focus on the mean and variance of $V_n^{X^{d_1}}$ and V_n. Following this idea, Marrel et al. [22] suggest the estimator

$$\tilde{\mathcal{S}}_n^{X^{d_1}} = \frac{\mathbb{E}_Z\left[\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right]}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]}, \tag{3.3}$$

where $\mathbb{E}_Z[.]$ stands for the expectation in the probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$. Furthermore, the uncertainty on $\tilde{\mathcal{S}}_n^{X^{d_1}}$ is evaluated with the following quantity:

$$\sigma^2(\tilde{\mathcal{S}}_n^{X^{d_1}}) = \frac{\mathrm{var}_Z\left(\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right)}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]^2}. \tag{3.4}$$

As shown in [25] and [22], (3.3) and (3.4) can be derived analytically through multidimensional integrals for the cases $d_1 = i$, $i = 1, \dots, d$, i.e., for the first-order indices. Furthermore, with some particular formulations of $\mathbf{f}(x)$, $\mu(x)$, and $r(x, \tilde{x})$, these multidimensional integrals can be written as the product of one-dimensional ones. We note that a method is suggested in [22] for generating samples of the numerator $\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)$ in (3.2). It allows for estimating the uncertainty of $\tilde{\mathcal{S}}_n^{X^{d_1}}$ in (3.3) without processing the complex numerical integration involved in (3.4).

Discussion. The method suggested in [25] and [22] provides an interesting tool for performing sensitivity analysis of complex models. Nevertheless, in our opinion it suffers from the following flaws:

1. For general choices of $\mathbf{f}(x)$, $\mu(x)$, and $r(x, \tilde{x})$, the numerical evaluations of (3.3) and (3.4) can be very complex since they require multidimensional integrals.
2. The method is derived for first-order sensitivity indices and cannot easily be extended to higher-order indices.
3. The method allows for inferring from the surrogate model uncertainty about the sensitivity indices but does not allow for taking into account the numerical errors due to the multidimensional integration.
4. The considered index expectation and deviation do not correspond to the real Sobol index ones since we obviously have

$$\frac{\mathbb{E}_Z\left[\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right]}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]} \neq \mathbb{E}_Z\left[\frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}\right]$$

and

$$\frac{\mathrm{var}_Z\left(\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right)}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]^2} \neq \mathrm{var}_Z\left(\frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}\right).$$

In fact, Oakley and O'Hagan [25] are interested in $V_n^{X^{d_1}}$ and V_n separately and do not consider the ratio $V_n^{X^{d_1}}/V_n$. Indeed, for very small V_n it would not be of interest to find which variables contribute to the variance.

We can deal with points 1, 2, and 3 by suggesting a Monte Carlo sampling method for evaluating (3.3) and (3.4) instead of quadrature integration. Nonetheless, we do not tackle the issue of point 4. To handle it, we suggest a strategy that allows for considering the joint distribution of $V_n^{X^{d_1}}$ and V_n. We present the method in section 4.

4. Kriging-based sensitivity analysis: A second approach.

We have highlighted at the end of subsection 3.2 that one of the main flaws of the method presented by [25] is that it does not care about the exact definition of Sobol indices. We present in subsection 4.1 another approach which deals with this issue. Then, in subsection 4.3 we present an efficient method for computing it.

4.1. Kriging-based Sobol index estimation.

First, in the previous section we have considered the variance of the main effects $V^{X^{d_1}}$ and the total variance V separately in (2.7). That is why the ratio of the expectations is considered as a sensitivity index in (3.3). In fact, in a Sobol index framework, we are interested in the ratio between $V^{X^{d_1}}$ and V. Therefore,

we suggest dealing directly with the following estimator (see (4.1)):

$$\mathcal{S}_{m,n}^{X^{d_1}} = \frac{V_{m,n}^{X^{d_1}}}{V_{m,n}} = \frac{\frac{1}{m}\sum_{i=1}^{m} Z_n(X_i)Z_n(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^{m} Z_n(X_i)\frac{1}{m}\sum_{i=1}^{m} Z_n(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^{m} Z_n(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^{m} Z_n(X_i)\right)^2}, \tag{4.1}$$

which corresponds to the ratio $V^{X^{d_1}}/V$ after substituting the code $z(x)$ by the Gaussian process $Z_n(x)$ and estimating the terms $\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)$ and $\mathrm{var}_X\left(Z_n(X)\right)$ with a Monte Carlo procedure as presented in [32]. We note that we can naturally adapt the presented estimator with those suggested by [34] and [15]. Nevertheless, we cannot obtain closed form expressions for the mean or the variance of this estimator. We thus have to numerically estimate them. We present in Algorithm 1 the suggested method for computing the distribution of $\mathcal{S}_{m,n}^{X^{d_1}}$.

Algorithm 1. Evaluation of the distribution of $\mathcal{S}_{m,n}^{X^{d_1}}$.

1: Build $Z_n(x)$ from the n observations $\mathbf{z}^n$ of $z(x)$ at points in $\mathbf{D}$ (see (3.1)).
2: Generate two samples $(x_i)_{i=1,\dots,m}$ and $(\tilde{x}_i)_{i=1,\dots,m}$ of the random vectors $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$ with respect to the probability measure μ (see Proposition 1).
3: Set N_Z as the number of samples for $Z_n(x)$ and B the number of bootstrap samples for evaluating the uncertainty due to Monte Carlo integration.
4: **for** $k = 1, \dots, N_Z$ **do**
5: Sample a realization $z_n(\mathbf{x})$ of $Z_n(\mathbf{x})$ with $\mathbf{x} = \{(x_i)_{i=1,\dots,m}, (\tilde{x}_i)_{i=1,\dots,m}\}$.
6: Compute $\hat{\mathcal{S}}_{m,n,k,1}^{X^{d_1}}$ thanks to (4.1) from $z_n(\mathbf{x})$.
7: **for** $l = 2, \dots, B$ **do**
8: Sample with replacements two samples $\mathbf{u}$ and $\tilde{\mathbf{u}}$ from $\{(x_i)_{i=1,\dots,m}\}$ and $\{(\tilde{x}_i)_{i=1,\dots,m}\}$.
9: Compute $\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}$ from $z_n(\mathbf{x}^B)$ with $\mathbf{x}^B = \{\mathbf{u}, \tilde{\mathbf{u}}\}$.
10: **end for**
11: **end for**
return $\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}\right)_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}}$.

The output $\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}\right)_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}}$ of Algorithm 1 is a sample of size $N_Z \times B$ of $\mathcal{S}_{m,n}^{X^{d_1}}$ defined on $(\Omega_X \times \Omega_Z, \sigma(\mathcal{F}_X \times \mathcal{F}_Z), \mathbb{P}_X \times \mathbb{P}_Z)$ (i.e., $\mathcal{S}_{m,n}^{X^{d_1}}$ takes into account both the uncertainty of the metamodel and that of the Monte Carlo integration). Then, we can deduce the following estimate $\bar{\mathcal{S}}_{m,n}^{X^{d_1}}$ for $\mathcal{S}_{m,n}^{X^{d_1}}$:

$$\bar{\mathcal{S}}_{m,n}^{X^{d_1}} = \frac{1}{N_Z B} \sum_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}} \hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}. \tag{4.2}$$

Furthermore, we can estimate the variance of $\mathcal{S}_{m,n}^{X^{d_1}}$ with

$$\hat{\sigma}^2(\mathcal{S}_{m,n}^{X^{d_1}}) = \frac{1}{N_Z B - 1} \sum_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}} \left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}} - \bar{\mathcal{S}}_{m,n}^{X^{d_1}}\right)^2. \tag{4.3}$$

We note that the computational limitation of the algorithm is the sampling of the Gaussian process $Z_n(x)$ on $\mathbf{x} = \{(x_i)_{i=1,\dots,m}, (\tilde{x}_i)_{i=1,\dots,m}\}$. For that reason, we use a bootstrap procedure to evaluate the uncertainty of the Monte Carlo integration instead of sampling different realizations of the random vectors $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$. Furthermore, the same bootstrap samples are used for the N_Z realizations of $Z_n(x)$.

Nevertheless, the number of Monte Carlo samples m is very large in general—it is often around $m = 5000d$—and it thus can be an issue to compute realizations of $Z_n(x)$ on $\mathbf{x}$. We present in subsection 4.3 an efficient method for dealing with this point for any choice of $\mu(x)$, $\mathbf{f}(x)$, and $r(x, \tilde{x})$ and any index order. The idea of carrying out an estimation of (4.1) from realizations of conditional Gaussian processes has already been suggested in [25] and [8]. The main contribution of this section is the procedure for balancing the Monte Carlo and the metamodel errors (see subsection 4.2).

4.2. Determining the minimal number of Monte Carlo samples m. We are interested here in quantifying the uncertainty of the considered estimator $\mathcal{S}^{X^{d_1}}_{m,n}$ (4.1). This estimator integrates two sources of uncertainty: the first one is related to the metamodel approximation, and the second one is related to the Monte Carlo integration. Therefore, we can decompose the variance of $\mathcal{S}^{X^{d_1}}_{m,n}$ as follows:

$$\mathrm{var}\left(\mathcal{S}^{X^{d_1}}_{m,n}\right) = \mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right]\right) + \mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right]\right),$$

where $\mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right]\right)$ is the contribution of the metamodel on the variability of $\mathcal{S}^{X^{d_1}}_{m,n}$ and $\mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right]\right)$ is that of the Monte Carlo integration. Let us consider the following two equalities obtained from the total variance formula:

$$\begin{cases} \mathrm{var}\left(\mathcal{S}^{X^{d_1}}_{m,n}\right) &= \mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right]\right) + \mathbb{E}_Z\left[\mathrm{var}_X\left(\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right)\right], \\ \mathrm{var}\left(\mathcal{S}^{X^{d_1}}_{m,n}\right) &= \mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right]\right) + \mathbb{E}_X\left[\mathrm{var}_Z\left(\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right)\right]. \end{cases}$$

We can deduce the following equalities:

$$\begin{cases} \mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right]\right) &= \mathbb{E}_X\left[\mathrm{var}_Z\left(\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right)\right], \\ \mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}^{X^{d_1}}_{m,n}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right]\right) &= \mathbb{E}_Z\left[\mathrm{var}_X\left(\mathcal{S}^{X^{d_1}}_{m,n}\middle|Z_n(x)\right)\right]. \end{cases}$$

Therefore, from the sample $\left(\hat{\mathcal{S}}^{X^{d_1}}_{m,n,k,l}\right)_{\substack{k=1,\dots,N_Z\\ l=1,\dots,B}}$ we can estimate the part of variance of the estimator $\mathcal{S}^{X^{d_1}}_{m,n}$ related to the metamodeling as follows:

$$\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) = \frac{1}{B}\sum_{l=1}^{B}\frac{1}{N_Z-1}\sum_{k=1}^{N_Z}\left(\hat{\mathcal{S}}^{X^{d_1}}_{m,n,k,l} - \bar{\hat{\mathcal{S}}}^{X^{d_1}}_{m,n,l}\right)^2, \tag{4.4}$$

where $\bar{\hat{\mathcal{S}}}^{X^{d_1}}_{m,n,l} = \left(\sum_{i=1}^{N_Z}\mathcal{S}^{X^{d_1}}_{m,n,i,l}\right)/N_Z$. Furthermore, we can evaluate the part of variance of $\mathcal{S}^{X^{d_1}}_{m,n}$ related to the Monte Carlo integration as follows:

$$\hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n}) = \frac{1}{N_Z}\sum_{i=1}^{N_Z}\frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\mathcal{S}}^{X^{d_1}}_{m,n,k,i} - \bar{\bar{\hat{\mathcal{S}}}}^{X^{d_1}}_{m,n,k}\right)^2, \tag{4.5}$$

where $\bar{\bar{\mathcal{S}}}^{X^{d_1}}_{m,n,k} = \left(\sum_{i=1}^{B} \mathcal{S}^{X^{d_1}}_{m,n,k,i}\right)/B$.

Therefore, we have three different cases:

1. $\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) \gg \hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n})$: the estimation error of $\mathcal{S}^{X^{d_1}}_{m,n}$ is essentially due to the metamodel error.
2. $\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) \ll \hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n})$: the estimation error of $\mathcal{S}^{X^{d_1}}_{m,n}$ is essentially due to the Monte Carlo error.
3. $\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) \approx \hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n})$: the metamodel and the Monte Carlo errors have the same contribution on the estimation error of $\mathcal{S}^{X^{d_1}}_{m,n}$.

Considering that the number of observations n is fixed, the minimal number of Monte Carlo samples m is that such that $\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) \approx \hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n})$. We call it "minimal" since it is the one from which the Monte Carlo error no longer dominates. Therefore, it should be the minimum number of required samples in practical applications. In practice, to determine it, we start with a small value of m and we increase it while the inequality $\hat{\sigma}^2_{Z_n}(\mathcal{S}^{X^{d_1}}_{m,n}) > \hat{\sigma}^2_{MC}(\mathcal{S}^{X^{d_1}}_{m,n})$ is true.

4.3. Sampling with respect to the kriging predictive distribution on large data sets.

We saw in the previous subsection in Algorithm 1 that in a kriging framework, we can assess the distribution of the Sobol index estimators from realizations of the conditional Gaussian process $Z_n(x)$ at points in $\mathbf{x}$. Nevertheless, the size of the corresponding random vector could be important since it equals twice the number of Monte Carlo samples m. Therefore, computing such realizations could lead to numerical issues such as an ill-conditioned matrix or huge computational cost, especially if we use a Cholesky decomposition. Indeed, Cholesky decomposition complexity is $\mathcal{O}((2m)^3)$, and it often leads to an ill-conditioned matrix since the predictive variance of $Z_n(x)$ is close to zero around the experimental design points.

Let us introduce the following unconditioned Gaussian process:

$$\tilde{Z}(x) \sim \mathrm{GP}(0, \sigma^2 r(x, \tilde{x})). \tag{4.6}$$

We have the following proposition.

Proposition 2 (sampling $Z_n(x)$ by kriging conditioning [5]). *Let us consider the following Gaussian process:*

$$\tilde{Z}_n(x) = m_n(x) - \tilde{m}_n(x) + \tilde{Z}(x), \tag{4.7}$$

where $m_n(x)$ is the predictive mean of $Z_n(x)$ (3.1),

$$\tilde{m}_n(x) = \mathbf{f}'(x)\tilde{\boldsymbol{\beta}} + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\tilde{Z}(\mathbf{D}) - \mathbf{F}\tilde{\boldsymbol{\beta}}\right), \tag{4.8}$$

and $\tilde{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\tilde{Z}(\mathbf{D})$. Then, we have

$$\tilde{Z}_n(x) \overset{\mathcal{L}}{=} Z_n(x),$$

where $Z_n(x)$ has the distribution of the Gaussian process $Z(x)$ of mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $\sigma^2 r(x, \tilde{x})$ conditioned by $\mathbf{z}^n$ at points in $\mathbf{D}$ (3.1). We note that we are in a universal kriging case; i.e., we infer from the parameter $\boldsymbol{\beta}$. In a simple kriging case, the proposition remains true by setting $\tilde{\boldsymbol{\beta}} = 0$.

The strength of Proposition 2 is that it allows for sampling with respect to the distribution of $Z_n(x)$ by sampling an unconditioned Gaussian process $\tilde{Z}(x)$. The first consequence is that the conditioning of the covariance matrix is better since the variance of $\tilde{Z}(x)$ is not close to zero around points in $\mathbf{D}$. The second important consequence is that it allows for using efficient algorithms to compute realizations of $\tilde{Z}(x)$. For example, if $r(x,\tilde{x})$ is a stationary kernel, one can use Bochner's theorem (see [35, p. 29]) and the Fourier representation of $\tilde{Z}(x)$ to compute realizations of $\tilde{Z}(x)$ as presented in [35]. Furthermore, another efficient method is to use Mercer's representation of $r(x,\tilde{x})$ (see [19] and [7]) and the Nyström procedure to approximate the Karhunen–Loève decomposition of $\tilde{Z}(x)$ as presented in [28, p. 98]. One of the main advantages of the Karhunen–Loève decomposition of $Z(x)$ is that it allows for sequentially adding new points to $\mathbf{x}$ without re-estimating the decomposition. Therefore, we can easily obtain the values of a given realization $z_n(x)$ of $Z_n(x)$ at new points not in $\mathbf{x}$. This interesting property will allow us to efficiently estimate the number m of Monte Carlo samples such that the metamodel error and the Monte Carlo estimation one are equivalent (see subsection 4.2).

5. Multifidelity cokriging-based sensitivity analysis.

Now let us suppose that we have s levels of code $(z_t(x))_{t=1,\dots,s}$ from the less accurate one $z_1(x)$ to the most accurate one $z_s(x)$ and that we want to perform a global sensitivity analysis for $z_s(x)$.

We present in subsection 5.1 the considered multifidelity cokriging model. Then, we present in subsection 5.2 the extension of our approach to perform cokriging-based multifidelity sensitivity analysis.

5.1. Multifidelity cokriging model.

We consider that, conditioning on the model parameters, $(z_t(x))_{t=1,\dots,s}$ are realizations of Gaussian processes $(Z_t(x))_{t=1,\dots,s}$. Furthermore, we consider the following multifidelity model $t = 2,\dots,s$:

$$\left\{\begin{array}{l} Z_t(x) = \rho_{t-1} Z^*_{t-1}(x) + \delta_t(x), \\ Z^*_{t-1}(x) \perp \delta_t(x), \\ Z^*_{t-1}(x) \sim [Z_{t-1}(x) | \mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\sigma}^2], \end{array}\right. \tag{5.1}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=1,\dots,s}$, $\boldsymbol{\rho} = (\rho_{t-1})_{t=2,\dots,s}$, $\boldsymbol{\sigma}^2 = (\sigma_t^2)_{t=1,\dots,s}$, $\mathbf{Z}^{(l-1)} = (Z_1(\mathbf{D}_1),\dots,Z_{t-1}(\mathbf{D}_{t-1}))$, $\mathbf{z}^{(t-1)} = (z_1(\mathbf{D}_1),\dots,z_{t-1}(\mathbf{D}_{t-1}))$, and $(\mathbf{D}_t)_{t=1,\dots,s}$ are the experimental design sets at level t with n_t points and such that $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$. Further, conditioning on $\boldsymbol{\beta}_t$ and σ_t^2, $\delta_t(x)$ is a Gaussian process of mean $\mathbf{f}_t'(x)\boldsymbol{\beta}_t$ and covariance $\sigma_t^2 r_t(x,\tilde{x})$ and we use the convention $Z_1(x) = \delta_1(x)$. This model is analogous to that presented in [17] except that in (5.1) the code at level $t-1$ is modeled by a Gaussian process having the conditional distribution $Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\sigma}^2]$. In [17], this code is modeled by the unconditioned Gaussian process $Z_{t-1}(x)$. The model (5.1) allows for obtaining closed form expressions for the mean and the variance of the predictive distribution (5.2).

We propose a Bayesian formulation of the model which allows us to consider noninformative prior distributions for the the regression parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=1,\dots,s}$ and the adjustment parameters $\boldsymbol{\rho} = (\rho_{t-1})_{t=2,\dots,s}$. This leads to the following predictive distribution, which integrates the posterior distributions of the parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=1,\dots,s}$ and $\boldsymbol{\rho} = (\rho_{t-1})_{t=2,\dots,s}$:

$$[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]. \tag{5.2}$$

The predictive distribution (5.2) is not Gaussian. Nevertheless, we can have closed form expressions for its mean $\mu^s_{n_s}(x)$ and covariance $k^s_{n_s}(x,\tilde{x})$:

$$\mu^s_{n_s}(x) = \hat{\rho}_{s-1}\mu^{s-1}_{n_{s-1}}(x) + \mu_{\delta_s}(x) \tag{5.3}$$

and

$$k^s_{n_s}(x,\tilde{x}) = \sigma^2_{\rho_{s-1}} k^{s-1}_{n_{s-1}}(x,\tilde{x}) + k_{\delta_s}(x,\tilde{x}), \tag{5.4}$$

where for $t = 1,\dots,s$, $\begin{pmatrix} \hat{\rho}_{t-1} \\ \hat{\boldsymbol{\beta}}_t \end{pmatrix} = (\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{z}_t$, $\mathbf{H}_t = [z_{t-1}(\mathbf{D}_t) \quad \mathbf{F}_t]$, $\mathbf{F}_t = \mathbf{f}_t'(\mathbf{D}_t)$, $\hat{\rho}_0 = 0$, $\mathbf{H}_1 = \mathbf{F}_1$, $\sigma^2_{\rho_{t-1}} = \hat{\rho}^2_{t-1} + \left[(\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\right]_{[1,1]}$, $\sigma^2_{\rho_0} = 0$,

$$\mu_{\delta_t}(x) = \mathbf{f}_t'(x)\hat{\boldsymbol{\beta}}_t + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{F}_t\hat{\boldsymbol{\beta}}_t - \hat{\rho}_{t-1}z_{t-1}(\mathbf{D}^t)), \tag{5.5}$$

and

$$k_{\delta_t}(x,\tilde{x}) = \sigma^2_t\left(r_t(x,\tilde{x}) - \begin{pmatrix}\mathbf{h}_t'(x) & \mathbf{r}_t'(x)\end{pmatrix}\begin{pmatrix}0 & \mathbf{H}_t' \\ \mathbf{H}_t & \mathbf{R}_t\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{h}_t(\tilde{x}) \\ \mathbf{r}_t(\tilde{x})\end{pmatrix}\right), \tag{5.6}$$

with $\mathbf{h}_t'(x) = [\mu^{t-1}_{n_{t-1}}(x) \quad \mathbf{f}_t'(x)]$ and $\mathbf{h}_1'(x) = \mathbf{f}_1'(x)$. We note that, in the mean of the predictive distribution, the regression and adjustment parameters have been replaced by their posterior means. Furthermore, the predictive variance integrates the uncertainty due to the regression and adjustment parameters. For a Bayesian point of view, we integrate the posterior distribution of the regression and adjustment parameters using the noninformative prior $p(\rho_{t-1},\boldsymbol{\beta}_t) \propto 1$.

We note that for each $t = 1,\dots,s$, the variance parameter σ^2_t is estimated with a restricted maximum likelihood method. Thus, its estimation is given by $\hat{\sigma}^2_t = (\mathbf{z}_t - \mathbf{H}_t\hat{\boldsymbol{\beta}}_t)'\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{H}_t\hat{\boldsymbol{\beta}}_t)/(n_t - p_t - 1)$, where p_t is the size of $\boldsymbol{\beta}_t$.

5.2. Extension of the second approach for multifidelity cokriging models. We present here the extension of the approach presented in section 4 to the multifidelity cokriging model. Therefore, we aim to sample with respect to the distribution of

$$\mathcal{S}^{X^{d_1}}_{m,s} = \frac{\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)Z_{n,s}(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)\frac{1}{m}\sum_{i=1}^m Z_{n,s}(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)\right)^2}, \tag{5.7}$$

which is the analogue of $\mathcal{S}^{X^{d_1}}_{m,n}$ (4.1) in a univariate case when we substitute $z(x)$ with $Z_{n,s}(x) \sim [Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)},\sigma^2]$. In fact, we can directly use Algorithm 1 by sampling realizations of $Z_{n,s}(x)$ instead of $Z_n(x)$. Moreover, the procedure presented in subsection 4.2 to determine the optimal number of Monte Carlo samples m is straightforward.

However, the distribution of $Z_{n,s}(x)$ is not Gaussian, and thus the method presented in subsection 4.3 cannot be used directly. In order to handle this problem, we consider the conditional distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)},\boldsymbol{\sigma}^2,\boldsymbol{\rho},\boldsymbol{\beta}]$, with $\boldsymbol{\sigma}^2 = (\sigma^2_t)_{t=1,\dots,s}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=2,\dots,s}$, and $\boldsymbol{\rho} = (\rho_{t-1})_{t=2,\dots,s}$, which is Gaussian (note that we infer from $\boldsymbol{\beta}_1$). It corresponds to the

distribution (5.2) conditioning by $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$. Furthermore, the Bayesian estimation of $(\rho_{t-1}, \boldsymbol{\beta}_t)$ gives us for all $t = 2, \ldots, s$

$$\begin{pmatrix} \rho_{t-1} \\ \boldsymbol{\beta}_t \end{pmatrix} \sim \mathcal{N}\left((\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{z}_t, \sigma_t^2(\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\right). \tag{5.8}$$

We note that the nested property $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$ implies that $\begin{pmatrix} \rho_{t-1} \\ \boldsymbol{\beta}_t \end{pmatrix}$ is independent of $\mathbf{z}_{t+1}$. From the recursive formulation given in (5.1), we can define the following Gaussian process having the desired distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2, \boldsymbol{\rho}, \boldsymbol{\beta}]$:

$$Z_{n,s,\rho,\boldsymbol{\beta}}(x) = \left(\prod_{j=1}^{s-1}\rho_j\right) Z_{n,1}(x) + \sum_{t=2}^{s-1}\left(\prod_{j=t}^{s-1}\rho_j\right)\delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) + \delta_{s,\rho_{s-1},\boldsymbol{\beta}_s}(x), \tag{5.9}$$

where (see (5.5) and (5.6))

$$Z_{n,1}(x) \sim \mathrm{GP}\left(\mu_{\delta_1}(x), k_{\delta_1}(x,\tilde{x})\right), \tag{5.10}$$

and for $t = 2, \ldots, s$

$$\delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) \sim \mathrm{GP}\left(\mu_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x), k_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x,\tilde{x})\right), \tag{5.11}$$

with $\mu_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) = \mathbf{r}_t'(x)\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{F}_t\boldsymbol{\beta}_t - \rho_{t-1}z_{t-1}(\mathbf{D}^t))$, $\left((\delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x))_{t=2,\ldots,s}, Z_{n,1}(x)\right)$ independent, and

$$k_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x,\tilde{x}) = \sigma_t^2\left(r_t(x,\tilde{x}) - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t(\tilde{x})\right).$$

Therefore, we can deduce the following algorithm to compute a realization $z_{n,s}(x)$ of $Z_{n,s}(x) \sim [Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]$.

Algorithm 2. Sampling with respect to the predictive distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2]$.

1: Generate a sample $z_{n,1}(x)$ with respect to (5.10) thanks to the method presented in Proposition 2 in the universal kriging case.
2: Set $z_{n,s}(x) = z_{n,1}(x)$.
3: **for** $t = 2, \ldots, s$ **do**
4: Generate a sample $\begin{pmatrix} \rho_{t-1}^* \\ \boldsymbol{\beta}_t^* \end{pmatrix}$ with respect to (5.8).
5: Conditionally to $\begin{pmatrix} \rho_{t-1}^* \\ \boldsymbol{\beta}_t^* \end{pmatrix}$, generate a sample $\delta^*_{t,\rho^*_{t-1},\boldsymbol{\beta}^*_t}(x)$ with respect to (5.11) thanks to the method presented in Proposition 2 in the simple kriging case.
6: Set $z_{n,s}(x) = \rho_{t-1}^* z_{n,s}(x) + \delta^*_{t,\rho^*_{t-1},\boldsymbol{\beta}^*_t}(x)$.
7: **end for**
return $z_{n,s}(x)$.

Algorithm 2 provides an efficient tool for sampling with respect to the distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2]$. Then, from each sample we can estimate the Sobol indices with a Monte Carlo procedure. Naturally, we can easily use a bootstrap procedure to take into account the uncertainty due to the Monte Carlo scheme. Furthermore, we see in Algorithm 2 that once a sample of $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2]$ is available, a sample for each distribution $[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \sigma^2]$, $t = 1, \ldots, s-1$, is also available. Therefore, we can directly quantify the difference between the Sobol indices at a level t and those at another level $\tilde{t}$.

6. Numerical illustrations on an academic example. We illustrate here the kriging-based sensitivity analysis suggested in section 4. We remind the reader that the aim of this approach is to perform a sensitivity index taking into account both the uncertainty related to the surrogate modeling and that related to the Monte Carlo integration. Let us consider the Ishigami function

$$z(x_1, x_2, x_3) = \sin(x_1) + 7\sin(\mathrm{x}_2)^2 + 0.1x_3^4\sin(x_1),$$

where μ_i is uniform on $[-\pi, \pi]$, $i = 1, 2, 3$. We are interested in the first-order sensitivity indices theoretically given by

$$(S_1, S_2, S_3) = (0.314, 0.442, 0).$$

This section is organized as follows. First, in subsection 6.1 we compare the Sobol index estimator $\tilde{\mathcal{S}}_n^{X^{d_1}}$ proposed in [25] (see subsection 3.2), the suggested one given by the mean of $\mathcal{S}_{m,n}^{X^{d_1}}$ (4.1), and the usual one which consists in substituting $z(x)$ by the predictive mean $m_n(x)$ (3.1) in (2.10). Then, in sections 6.3, 6.4, and 6.5 we deal with the approach presented in section 4. In particular, we show that this approach is relevant for performing an uncertainty quantification taking into account both the uncertainty of the metamodeling and that of the Monte Carlo integration. We note that the construction of the surrogate models used in sections 6.3, 6.4, and 6.5 is presented in section 6.2.

6.1. Comparison between the different methods. The aim of this subsection is to perform a numerical comparison between $\tilde{\mathcal{S}}_n^{X^{d_1}}$, the empirical mean of $\mathcal{S}_{m,n}^{X^{d_1}}$ given in (4.2), and the following estimator (see (2.10)):

$$\check{\mathcal{S}}_{m,n}^{X^{d_1}} = \frac{\frac{1}{m}\sum_{i=1}^m m_n(X_i)m_n(\tilde{X}_i) - \left(\frac{1}{2m}\sum_{i=1}^m m_n(X_i) + m_n(\tilde{X}_i)\right)^2}{\frac{1}{m}\sum_{i=1}^m m_n(X_i)^2 - \left(\frac{1}{2m}\sum_{i=1}^m m_n(X_i) + m_n(\tilde{X}_i)\right)^2}. \tag{6.1}$$

We note that the empirical mean $\bar{\mathcal{S}}_{m,n}^{X^{d_1}}$ of $\mathcal{S}_{m,n}^{X^{d_1}}$ is evaluated thanks to Algorithm 1, with $N_Z = 500$ and $B = 1$:

$$\bar{\mathcal{S}}_{m,n}^{X^{d_1}} = \frac{1}{N_Z}\sum_{k=1,\dots,N_Z} \hat{\mathcal{S}}_{m,n,k,1}^{X^{d_1}},$$

and for $\tilde{\mathcal{S}}_n^{X^{d_1}}$ and $\mathcal{S}_{m,n}^{X^{d_1}}$ we use the Monte Carlo estimator (2.10) suggested in [15] (it is that used in (6.1)). Then, for the comparison, different sizes of the learning sample are considered ($n = 40$, 50, 60, 70, 90, 120, 150, 200 observations) and we randomly build 100 Latin hypercube samples (LHSs) for each size of the learning sample. From these experimental design sets, we build kriging models with a constant trend β and a tensorized 5/2-Matérn kernel. Furthermore, the characteristic length scales $(\theta_i)_{i=1,2,3}$ are estimated with a maximum likelihood procedure for each design set [31]. The Nash–Sutcliffe model efficiency coefficient (sometimes called the predictivity coefficient Q^2),

$$\mathit{Eff}_n = 1 - \frac{\sum_{x\in T}(m_n(x) - z(x))^2}{\sum_{x\in T}(m_n(x) - \bar{z}(x))^2}, \quad \bar{z}(x) = \frac{1}{\#T}\sum_{x\in T} z(x),$$

of the different kriging models is evaluated on a test set T composed of 1,000 points uniformly spread on the input parameter space $[-\pi, \pi]^3$. The values of Eff_n are presented in Figure 1. The closer Eff is to 1, the more accurate is the model $m_n(x)$.

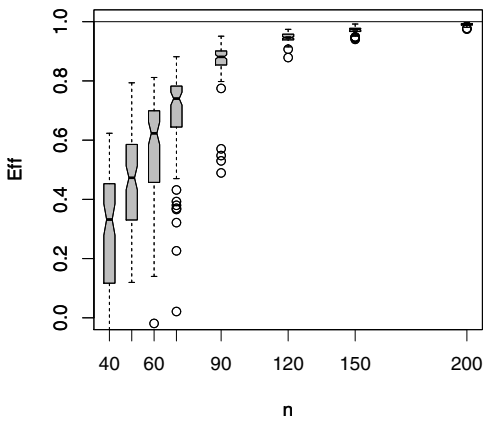

Figure 1. *Convergence of the model efficiency when the number n of observations increases.* 100 *LHSs are randomly sampled for each number of observations n.*

Figure 2 illustrates the Sobol index estimates obtained with the three methods. We see in Figure 2 that the suggested estimator $\bar{\mathcal{S}}^{X^{d_1}}_{m,n}$ performs as well as the usual estimator $\check{\mathcal{S}}^{X^{d_1}}_{m,n}$ (6.1). In fact, as we will see in the next subsections, the strength of the suggested estimator is to provide more relevant uncertainty quantification. Finally, we see in Figure 2 that the estimator $\tilde{\mathcal{S}}^{X^{d_1}}_{n}$ suggested in [25] seems to systematically underestimate the true value of the Sobol index for a nonnegligible index and when the model efficiency is low.

6.2. Model building and Monte Carlo–based estimator. For the numerical illustrations in sections 6.3 and 6.4, we use different kriging models built from different experimental design sets (optimized LHS with respect to the centered L_2-discrepancy criterion [6]) of sizes $n = 30, \ldots, 200$. Furthermore, for all kriging models, we consider a constant trend β and a tensorized 5/2-Matérn kernel (see [28]).

The characteristic length scales $(\theta_i)_{i=1,2,3}$ are estimated for each experimental design set by maximizing the marginal likelihood. Furthermore, the variance parameter σ^2 is estimated with a maximum likelihood method for each experimental design set too. Then, for each n, the Nash–Sutcliffe model efficiency is evaluated on a test set composed of 10,000 points uniformly spread on the input parameter space $[-\pi, \pi]^3$. Figure 3 illustrates the estimated values of Eff_n with respect to the number of observations n.

Then, for estimating the Sobol indices, we use the Monte Carlo–based estimator given by (2.10). It has the strength to be asymptotically efficient for the first-order indices (see [15]).

6.3. Sensitivity index estimates when n increases. Let us consider a fixed number of Monte Carlo samples $m = 10,000$. The aim of this subsection is to quantify the part of the index estimator uncertainty related to the Monte Carlo integration and that related to the surrogate modeling.

To perform such analysis we use the procedure presented in Algorithm 1 with $B = 300$

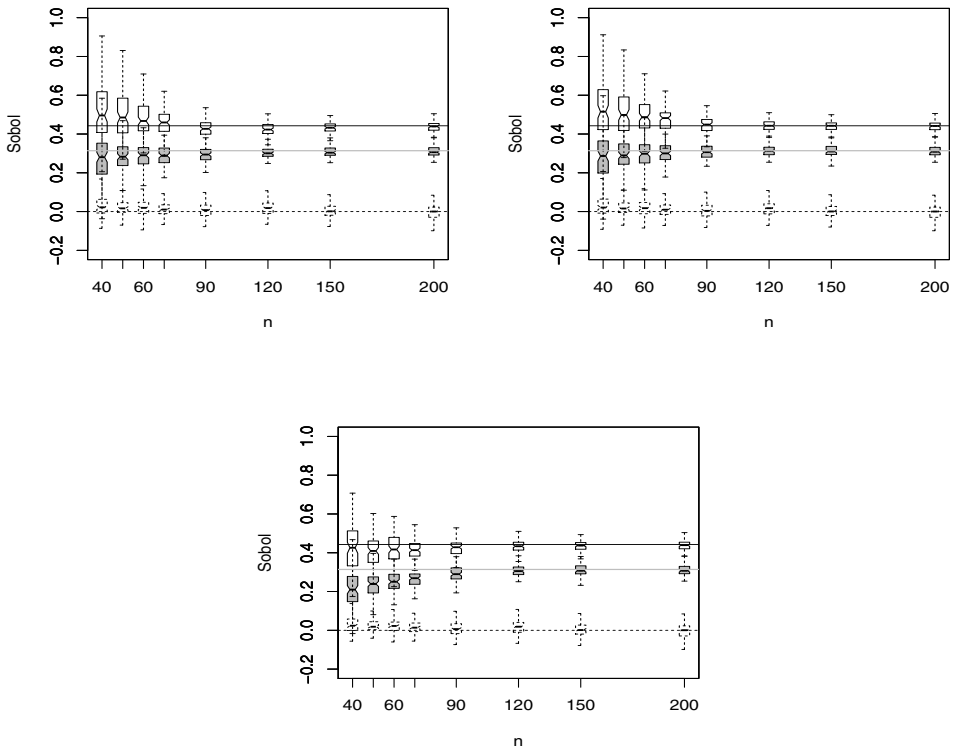

Figure 2. *Comparison between three Sobol index estimators. The comparisons are performed from* 100 *random LHSs for each number of observations n. The top left figure corresponds to the suggested Sobol estimator* $\bar{\mathcal{S}}_{m,n}^{X^{d_1}}$ *(see section* 4*), the top right figure corresponds to the usual (metamodel predictor only) estimator* $\check{\mathcal{S}}_{m,n}^{X^{d_1}}$ *(see* (6.1)*), and the bottom figure corresponds to the estimator* $\tilde{\mathcal{S}}_n^{X^{d_1}}$ *suggested in* [25]*. The horizontal lines represent the true values of the Sobol indices (solid gray line:* S_1*; solid black line:* S_2*; and dashed black line:* S_3*).*

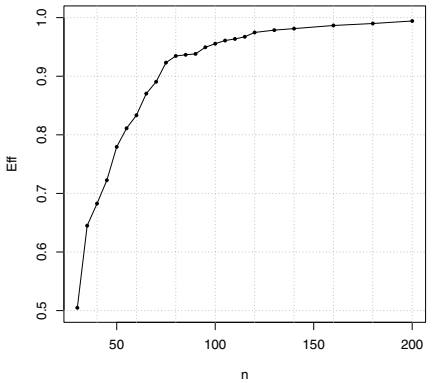

Figure 3. *Convergence of the model efficiency when the number n of observations increases. For each number of observations n, the experimental design set is an optimized LHS with respect to the centered* L_2*-discrepancy.*

bootstrap samples and $N_Z = 500$ realizations of $Z_n(x)$ (3.1). It results for each $i = 1, 2, 3$ in a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, with respect to the distribution of the estimator obtained by substituting $z(x)$ with $Z_n(x)$ in (2.10).

Then, we estimate the 0.05 and 0.95 quantiles of $\left(\hat{S}^i_{m,n,k,1}\right)$, $k = 1, \ldots, N_Z$, for each $i = 1, 2, 3$ with a bootstrap procedure. The resulting quantiles represent the uncertainty related to the surrogate modeling. Furthermore, we estimate the 2.50% and 97.50% quantiles of $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, with a bootstrap procedure too. These quantiles represent the total uncertainty of the index estimator. Figure 4 illustrates the result of this procedure for different numbers of observations n. We see in Figure 4 that for small values of n, the error related to the surrogate modeling dominates. Then, when n increases, this error decreases and it is that related to the Monte Carlo integration which is the largest. This emphasizes that it is worth adapting the number of Monte Carlo samples m to the number of observations n. Finally, we highlight that the equilibrium between the two types of uncertainty does not occur for the same n for the three indices. Indeed, it is around $n = 100$ for S_1, $n = 150$ for S_2, and $n = 75$ for S_3. We observe that the smaller the index is, the larger is its Monte Carlo estimation error.

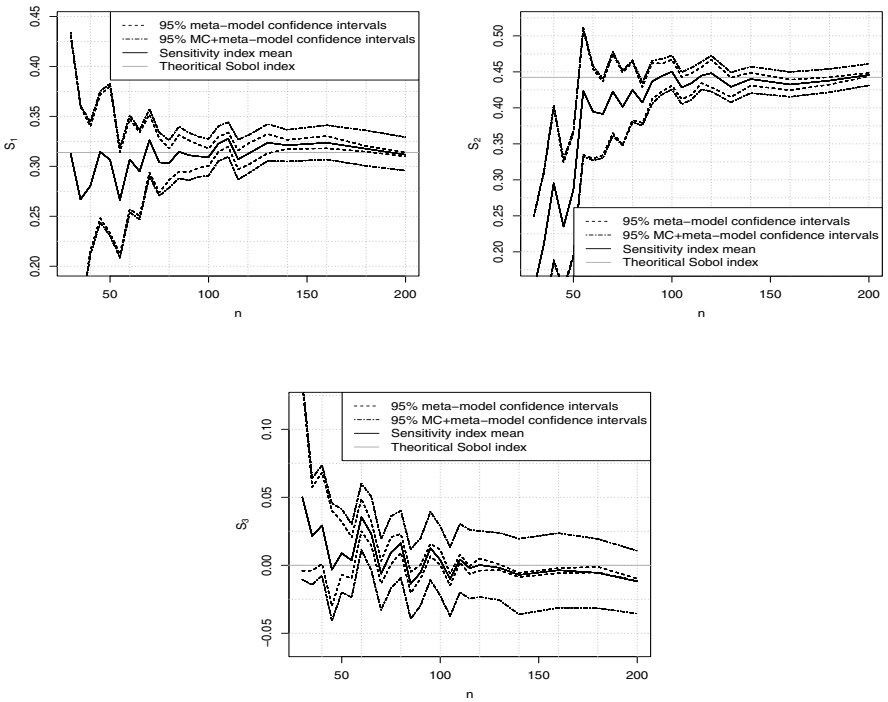

Figure 4. *Sensitivity index estimates when n increases. The solid lines represent the means of the sensitivity index estimators. The dashed lines represent the* 2.50% *and* 97.50% *confidence intervals taking into account only the uncertainty related to the surrogate modeling. The dashed-dotted lines represent the* 2.50% *and* 97.50% *confidence intervals taking into account both the uncertainty related to the surrogate modeling and that related to the Monte Carlo integration. The horizontal gray lines represent the true values of* S_1 *(on top left),* S_2 *(on top right), and* S_3 *(at bottom).*

6.4. Minimal Monte Carlo resource when n increases. We saw in the previous subsection that the equilibrium between the error related to the Monte Carlo integration and that related to the surrogate modeling depends on the considered sensitivity index. The purpose of this subsection is to determine this equilibrium for each index. To perform such analysis, we use the method presented in subsection 4.2.

Let us consider a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $n = 30, \ldots, 200$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, $i = 1, 2, 3$, generated with Algorithm 1 and using the Monte Carlo estimator presented in (2.10). For each pair (m, n) we can evaluate the variance $\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right)$, $i = 1, 2, 3$, related to the metamodeling with (4.4) and the variance $\hat{\sigma}^2_{MC}\left(S^i_{m,n}\right)$, $i = 1, 2, 3$, related to the Monte Carlo integration with (4.5). We state that the equilibrium between the two types of uncertainty corresponds to the case

$$\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right) = \hat{\sigma}^2_{MC}\left(S^i_{m,n}\right). \tag{6.2}$$

Actually, for practical applications, we choose m such that $\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right) \gg \hat{\sigma}^2_{MC}\left(S^i_{m,n}\right)$. We present in Figure 5 the pairs (m, n) such that the equality (6.2) is satisfied. We see that the smaller the sensitivity index is, the more important is the number of samples m required to have the equilibrium. Furthermore, we note that the curve increases extremely quickly for the index $S_3 = 0$. Therefore, it could be unrealistic to consider the equilibrium for this case, especially when n is important (i.e., $n > 100$).

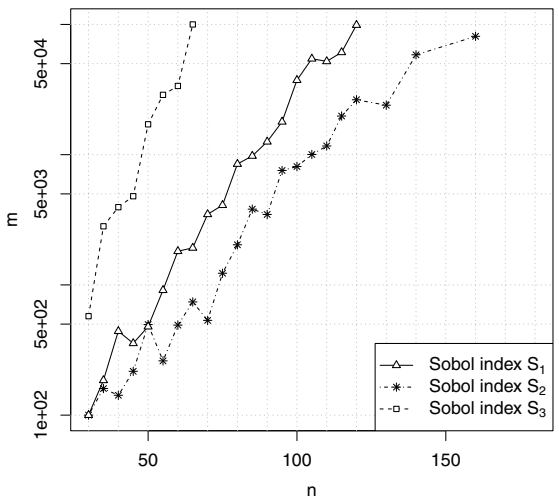

Figure 5. *Relation between the number of observations n and the number of Monte Carlo samples m such that the error related to the metamodeling and that related to the Monte Carlo scheme have the same order of magnitude.*

The presented analysis is of practical interest since it provides the minimal number of Monte Carlo samples m for the sensitivity index estimation in function of the number of observations n. Indeed, in the framework of computer experiments, the observations are often time-consuming and n cannot be large. Therefore, we look for a number of samples m such that the variance $\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right)$ related to the metamodeling is smaller than that of the Monte

Carlo integration $\hat{\sigma}^2_{MC}\left(S^i_{m,n}\right)$. However, we saw that it could be unfeasible for some values of the sensitivity index. In this case a compromise must necessarily be made.

6.5. Coverage rate of the suggested Sobol index estimator. Algorithm 1 in subsection 4.1 allows for obtaining a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, of the estimator of S_i for each $i = 1, 2, 3$. The purpose of this subsection is to verify the relevance of the confidence intervals provided by $\left(\hat{S}^i_{m,n,k,l}\right)$. To perform such analysis, we generate 200 random LHSs $(\mathbf{D}_{n,j})_{j=1,\ldots,200}$ for different numbers of observations n. For each $\mathbf{D}_{n,j}$, we build a kriging model with the procedure presented in subsection 6.2 and we generate a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, with $B = 200$ and $N_Z = 300$. The efficiency of the different kriging models with respect to the number of observation n is presented in Figure 6. From this sample, we evaluate the 2.50% and 97.50% quantiles with a bootstrap procedure and we check whether the true value of S_i is covered by these two quantiles. At the end of the procedure, the ratio between the number of confidence intervals covering the true value of S_i and the total number of confidence intervals (i.e., 200) has to be close to 95% for each n.

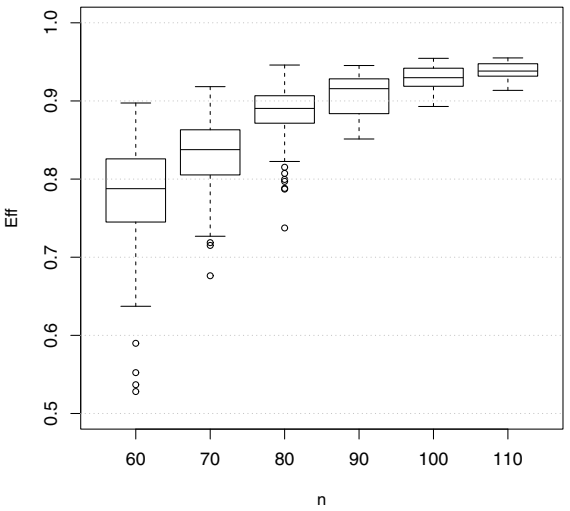

Figure 6. *Convergence of the model efficiency when the number n of observations increases. For each number of observations n,* 200 *LHSs are randomly sampled.*

Furthermore, to perform the analysis we use different values of m according to the procedure presented in subsection 4.2 for S_1 and S_2 (i.e., such that the variance related to the metamodeling has the same order of magnitude as that related to the Monte Carlo integration). For S_3, the number of Monte Carlo samples m increases too quickly with respect to n to use the method presented in subsection 4.2. Therefore, we fix m to the values presented in Table 1. We note that the values of m for S_3 are larger than those for S_1 and S_2.

The empirical 95%-confidence intervals as a function of the number of observations n are presented in Figure 7. We study three cases:

1. The confidence intervals are built from $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$. Therefore, it takes into account both the uncertainty related to the metamodel and that related to the Monte Carlo estimation.

Table 1

Numbers of Monte Carlo samples m for different values of the number of observations n for the estimation of S_3.

n	60	70	80	90	100	110
m	1,000	3,000	5,000	10,000	40,000	60,000

2. The confidence intervals are built from $\left(\hat{S}^i_{m,n,k,1}\right)$, $k = 1, \ldots, N_Z$. In this case, we do not use the bootstrap procedure to evaluate the uncertainty due to the Monte Carlo procedure. Therefore, we only take into account that due to the metamodel.
3. The confidence intervals are built from the estimator $\tilde{\mathcal{S}}_n^{X^{d_1}}$ (6.1) with a bootstrap procedure. Here, we estimate the Sobol indices with the kriging mean and we do not infer from the uncertainty of the metamodel. Therefore, we only take into account the uncertainty related to the Monte Carlo estimation.

We see in Figure 7 that the confidence intervals provided by the approach presented in section 4 are well evaluated for indices S_1 and S_3. Furthermore, they are underestimated when we take into account only the metamodel or the Monte Carlo uncertainty. This highlights the relevance of the suggested approach for performing uncertainty quantification on the Sobol index estimates. However, the coverage rate is underestimated for index S_2. This is even worse if we consider only the metamodel error. This may be due to poor learning in the direction x_2 for the the surrogate model. This emphasizes that the suggested method is valid only if the kriging variance well represents the modeling error. We note that methods are suggested in [2] to assess the relevance of the Gaussian process regression model.

7. Application of multifidelity sensitivity analysis.

In this section, we illustrate the multifidelity cokriging-based sensitivity analysis presented in section 5.1 on an example about a spherical tank under internal pressure.

7.1. Presentation of the problem.

The scheme of the considered tank is presented in Figure 8. We are interested in the von Mises stress on the point labeled 1 in Figure 8. It corresponds to the point where the stress is maximal. The von Mises stresses are of interest since the material yielding occurs when they reach the critical yield strength.

The system illustrated in Figure 8 depends on eight parameters:

- $P\,(MPa) \in [30, 50]$: the value of the internal pressure.
- $R_{int}\,(mm) \in [1500, 2500]$: the length of the internal radius of the shell.
- $T_{shell}\,(mm) \in [300, 500]$: the thickness of the shell.
- $T_{cap}\,(mm) \in [100, 300]$: the thickness of the cap.
- $E_{shell}\,(GPa) \in [63, 77]$: Young's modulus of the shell material.
- $E_{cap}\,(GPa) \in [189, 231]$: Young's modulus of the cap material.
- $\sigma_{y,shell}\,(MPa) \in [200, 300]$: the yield stress of the shell material.
- $\sigma_{y,cap}\,(MPa) \in [400, 800]$: the yield stress of the cap material.

The von Mises stress $z_2(x)$, $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$, is provided by a finite element code, called Aster, modeling the tank under pressure. The material properties of the shell correspond to high quality aluminums, and those of the cap correspond to steels from classical to high quality.

The cheaper version $z_1(x)$ of $z_2(x)$ is obtained by the one-dimensional simplification of

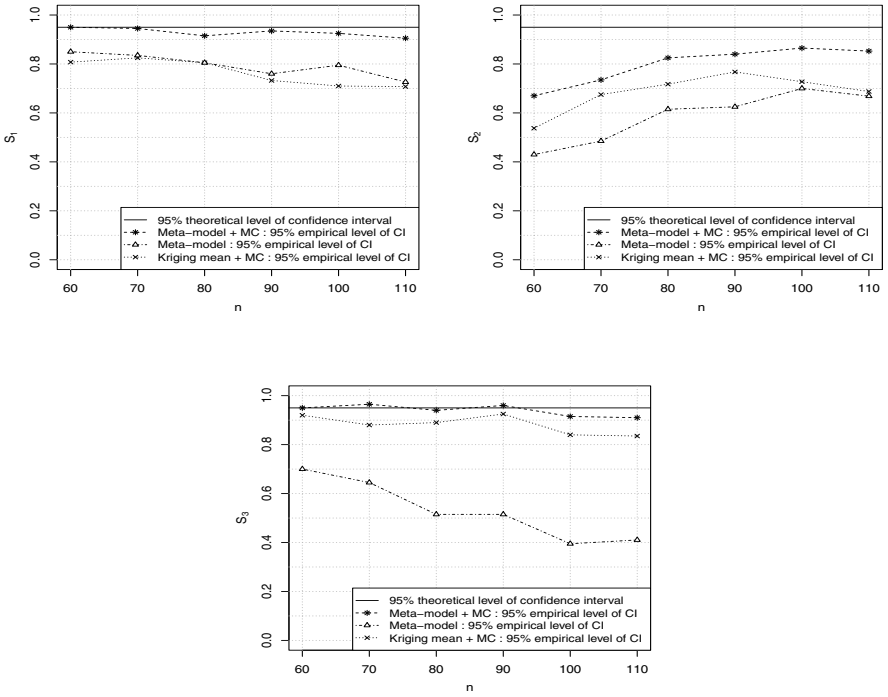

Figure 7. *Empirical* 95% *confidence intervals with respect to the number of observations* n *for* S_1 *(on top left),* S_2 *(on top right), and* S_3 *(at bottom). The empirical coverage rates are evaluated from* 200 *kriging models built from different random LHSs.*

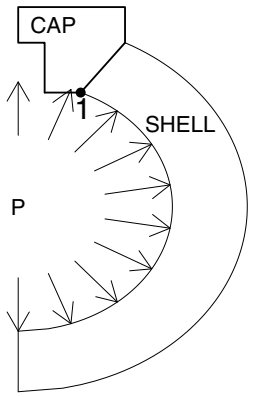

Figure 8. *Scheme of the spherical tank under pressure.*

the tank corresponding to a perfect spherical tank, i.e., without the cap:

$$z_1(x) = \frac{3}{2}\frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3}P.$$

7.2. Multifidelity model building. We present here the construction of the model presented in section 5.1.

First, we build two LHS design sets $\tilde{\mathbf{D}}_1$ and $\mathbf{D}_2$ of sizes $n_1 \times 8$ and $n_2 \times 8$ optimized with respect to the centered L_2-discrepancy criterion, with $n_1 = 100$ and $n_2 = 20$. We note that the input parameter x is normalized so that the measure $\mu(x)$ of the input parameters is uniform on $[0,1]^8$. In order to respect the nested property for the experimental design sets, we remove from $\tilde{\mathbf{D}}_1$ the n_2 points that are the closest to those of $\mathbf{D}_2$ and we set that $\mathbf{D}_1$ is the concatenation of $\mathbf{D}_2$ and $\tilde{\mathbf{D}}_1$. This procedure ensures that $\mathbf{D}_2 \subset \mathbf{D}_1$ without operating any transformation on $\mathbf{D}_2$.

Second, we run the expensive code $z_2(x)$ on points in $\mathbf{D}_2$ and the coarse code $z_1(x)$ on points in $\mathbf{D}_1$. The CPU time of the expensive code is around 1 minute. Furthermore, in order to have a fair illustration, we consider that the CPU time of the coarse code $z_1(x)$ is not negligible and we restrict its runs to $n_1 = 100$. We note that if the computational cost of the coarse code is negligible, we can integrate it into the trend functions $\mathbf{f}(x)$ of a kriging model. Therefore, in this case we join the framework of classical kriging models.

Third, we use tensorized 5/2-Matérn covariance kernels for $\sigma_1^2 r_1(x, \tilde{x})$ and $\sigma_2^2 r_2(x, \tilde{x})$ with characteristic length scales $(\theta_1^i)_{i=1,\dots,8}$ and $(\theta_2^i)_{i=1,\dots,8}$. Furthermore, we set that the regression functions are constants, i.e., $\mathbf{f}_1(x) = 1$ and $\mathbf{f}_2(x) = 1$. Therefore, the trend parameters $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are scalars representing the constant means of $Z_1(x)$ and $Z_2(x)$.

The estimates of the characteristic length scales are given in Table 2.

Table 2

Maximum likelihood estimates of the characteristic length scales of the tensorized 5/2-Matérn covariance kernels used in the multifidelity cokriging model. $\hat{\boldsymbol{\theta}}_1$ represents the estimates for the code level 1, and $\hat{\boldsymbol{\theta}}_2$ represents those for the bias between the code levels 1 and 2.

$\hat{\boldsymbol{\theta}}_1$	1.71	1.38	1.97	1.98	1.98	1.99	1.95	1.41
$\hat{\boldsymbol{\theta}}_2$	1.83	1.89	0.5	1.93	1.93	0.64	1.89	0.79

The estimates of the characteristic length scales given in Table 2 show that the model is very smooth. Then, Table 3 gives the posterior mean of the parameters $(\rho_1, \boldsymbol{\beta}_2)$ and $\boldsymbol{\beta}_1$ and the restricted maximum likelihood estimate of σ_1^2 and σ_2^2.

Table 3

Posterior means of the trend parameters $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and the adjustment parameter ρ_1 and maximum likelihood estimates of the variance parameters σ_1^2 and σ_2^2.

$\hat{\boldsymbol{\beta}}_1$	148.67	$\hat{\sigma}_1^2$	495.63
$(\hat{\rho}_1, \hat{\boldsymbol{\beta}}_2)$	(0.92, 57.61)	$\hat{\sigma}_2^2$	551.07

The parameter estimates presented in Table 3 show that there is an important bias between the cheap code and the expensive code since $\hat{\boldsymbol{\beta}}_2 \approx 58$, whereas the trend of the cheap code is $\hat{\boldsymbol{\beta}}_1 \approx 150$. In particular, it is greater than the standard deviation of the bias, which is $\hat{\sigma}_2 \approx 23$. Then, the posterior mean of the adjustment parameter $\hat{\rho}_1 = 0.92$ does not indicate a perfect correlation between the two levels of code. Indeed, the estimated correlation between $z_2(x)$ and $z_1(x)$ is 0.77. Furthermore, their estimated variance equals 1514 for $z_2(x)$ and 810

for $z_1(x)$. In fact, the adjustment parameter

$$\rho_1 = \frac{\mathrm{cov}(Z_2(x), Z_1(x))}{\mathrm{var}(Z_1(x))}$$

represents both the correlation degree and the scale factor between the codes $z_2(x)$ and $z_1(x)$.

Finally, we can estimate the accuracy of the suggested model with a leave-one-out cross validation procedure. From the leave-one-out errors, we estimate the Nash–Sutcliffe model efficiency $Eff_{LOO} = 83\%$. This means that the suggested multifidelity cokriging model explains 83% of the variability of the model. We note that the closer Eff_{LOO} is to 1, the more accurate is the model. Therefore, we have an excellent model despite the small number of observations $n_2 = 20$ used for the expensive code $z_2(x)$. In order to strengthen this result, we test the multifidelity model on an external test set of $7,000$ points, and the estimated efficiency is 86%, which is even better.

7.3. Multifidelity sensitivity analysis. Now let us perform a multifidelity sensitivity analysis using the approach presented in subsection 5.2. We are interested in the first-order sensitivity indices.

The principle of the method is to sample from the distribution (5.7) using Algorithm 2. We note that we use the Monte Carlo estimator (2.10) instead of (2.9) since it is asymptotically efficient for the first-order indices. We repeat Algorithm 2 to have $N_Z = 200$ realizations of the predictive distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$, and for each realization we generate $B = 150$ bootstrap samples. Furthermore, we choose $m = 20,000$ for the Monte Carlo sampling size so that the error due to the Monte Carlo integration is negligible compared to that due to the surrogate modeling (see subsections 4.2 and 6.4).

Sensitivity analysis for the cheap code. First, let us present the result of the sensitivity analysis for the cheap code. As emphasized in subsection 5.2, once samples with respect to the distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$ are available, samples for $[Z_1(x)|\mathbf{Z}^{(1)} = \mathbf{z}^{(1)}, \sigma_1^2]$ are also available. Therefore, from them we can perform a sensitivity analysis as presented in section 4. Moreover, from the explicit formula of $z_1(x)$ we expect that only the three variables P, R_{int}, and T_{shell} have an impact on the output.

The result of the sensitivity analysis for the cheap code $z_1(x)$ is given in Figure 9. We see in Figure 9 that only the three parameters P, R_{int}, and T_{shell} are influent as expected. Furthermore, the internal pressure is the most important parameter, whereas the geometrical parameters R_{int} and T_{shell} have equivalent impact on the output. The sum of the first-order sensitivity index means informs us that 97% of the variability of the output is explained by the first-order indices. The interactions between the parameters are thus negligible. Further, we see that the confidence intervals are tight and that the uncertainty on the Sobol index estimator is essentially due to the Monte Carlo integration. This means that the model's error on the cheap code is very low.

Sensitivity analysis for the expensive code. Second, we perform a sensitivity analysis for the expensive code $z_2(x)$ using the predictive distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$. The result of the analysis is presented in Figure 10.

We see in Figure 10 that the result of the sensitivity analysis for the expensive code is substantially different from that for the cheap code. First, the importance measure of the

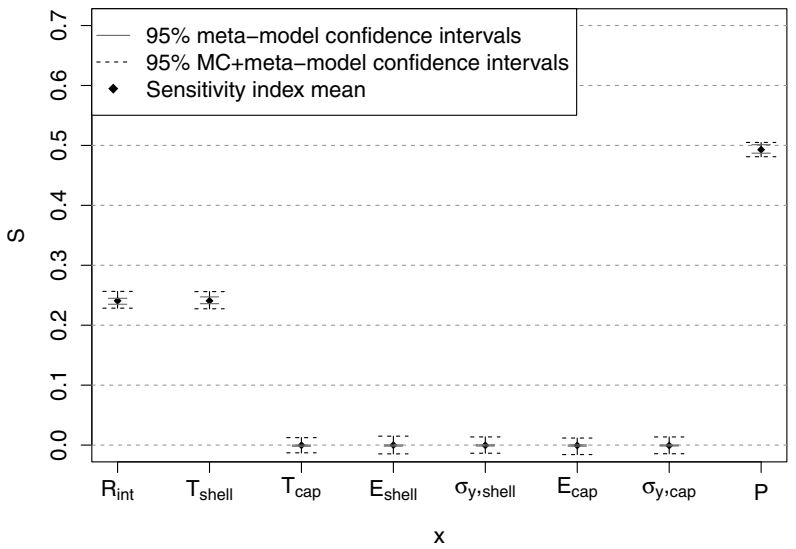

Figure 9. *Kriging-based sensitivity analysis for the cheap code. The diamonds represent the means of the first-order sensitivity index estimators, the solid gray lines represent the* 95% *confidence intervals taking into account only the metamodeling uncertainty, and the dashed lines represent the* 95% *confidence intervals taking into account the uncertainty due to both the Monte Carlo integration and the metamodeling. The means and the confidence intervals are obtained with Algorithm* 1.

parameters P, R_{int}, and T_{shell} decreases, although the internal pressure P remains the most influent parameter. Second, the material parameters E_{shell}, E_{cap}, $\sigma_{y,shell}$, and $\sigma_{y,cap}$ still have a negligible influence except for the rigidity of the cap E_{cap}. Then, the most noticeable difference is for the thickness of the cap T_{cap}, which is now the second most important parameter. Then the sum of the index estimator means equals 96.7%. This means that the first-order indices still explain the main part of the model variability.

The hierarchy between the parameters can be easily interpreted. Indeed, the coarse code corresponds to the approximation of the tank without the cap. Therefore, it is natural that the parameters related to the cap have no influence. On the contrary, for the expensive code, we are interested in the von Mises stress at the junction between the cap and the shell. Consequently, the parameters related to the cap now have an influence. However, it was difficult to have an intuition on the impact of the cap. We deduce from this analysis that it is in fact very important.

Influences of material parameters are negligible because the model stands in the regime of elastic deformations. It is thus physically coherent. In fact, they would be more influent in a plastic deformation regime, which can occur for more important internal pressure P.

The other important differences between the two sensitivity analyses is the magnitude of the confidence intervals. Indeed, we see in Figure 10 that, contrary to the cheap code, the confidence intervals for the sensitivity index estimators of the expensive code are very large. Therefore, despite the good multifidelity approximation of the expensive code, we have an important uncertainty on it. This is natural since we only use 20 runs of $z_2(x)$ to learn it. Finally, we note that the most important uncertainty is for T_{cap}. This is explained by the

fact that this parameter is not considered by the cheap code. Therefore, $z_1(x)$ brings no information about T_{cap} contrary to R_{int}, T_{shell}, and P.

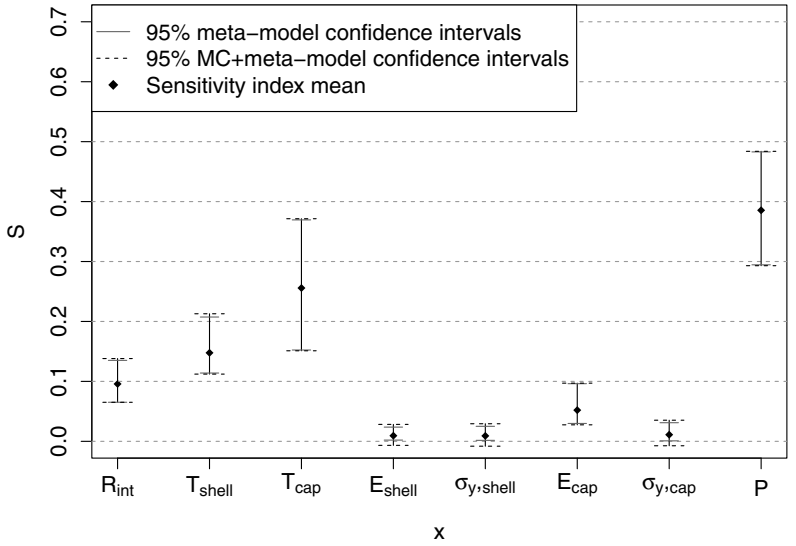

Figure 10. *Cokriging-based sensitivity analysis for the expensive code. The diamonds represent the means of the first-order sensitivity index estimators, the solid gray lines represent the 95% confidence intervals taking into account only the metamodeling uncertainty, and the dashed lines represent the 95% confidence intervals taking into account the uncertainty due to both the Monte Carlo integration and the metamodeling.*

Comparison between kriging- and cokriging-based sensitivity analysis. Finally, we perform in this paragraph a kriging-based sensitivity analysis using only the $n_2 = 20$ outputs $\mathbf{z}_2$ of $z_2(x)$. We note that we consider the same experimental design set $\mathbf{D}_2$ as previously. For building the kriging model, we consider a constant mean (i.e., $\mathbf{f}(x) = 1$) and a 5/2-Matérn covariance kernel parametrized with the characteristic length scales $\boldsymbol{\theta} = (\theta^i)_{i=1,\dots,8}$ and the variance parameter σ^2.

The model parameters are estimated with a maximum likelihood method (see [31]). We have $\hat{\sigma}^2 = 1877$ and $\hat{\boldsymbol{\theta}}$ given in Table 4.

Table 4

Characteristic length scale estimates for the kriging model of the complex code $z_2(x)$.

$\hat{\boldsymbol{\theta}}$	1.83	1.04	0.98	1.93	1.93	1.89	1.89	0.82

Furthermore, the efficiency of the model is 78%. Therefore, it is less accurate than the multifidelity cokriging model.

The Sobol index estimate is given by (4.2), where $N_Z = 20,000$ and $B = 150$, $n = 20$ (see Algorithm 1). From Algorithm 1, we can also deduce the 95% confidence intervals of the Sobol index estimates.

The results of the kriging-based sensitivity analysis are illustrated in Figure 11. We see a dramatic difference between this analysis and that performed with a multifidelity cokriging

model. Indeed, for the kriging-based analysis, the uncertainty on the Sobol index estimates is significantly larger. Therefore, we cannot give a hierarchy between the importance measure of the input parameters. On the other hand, the results given by the multifidelity approach are more discriminant.

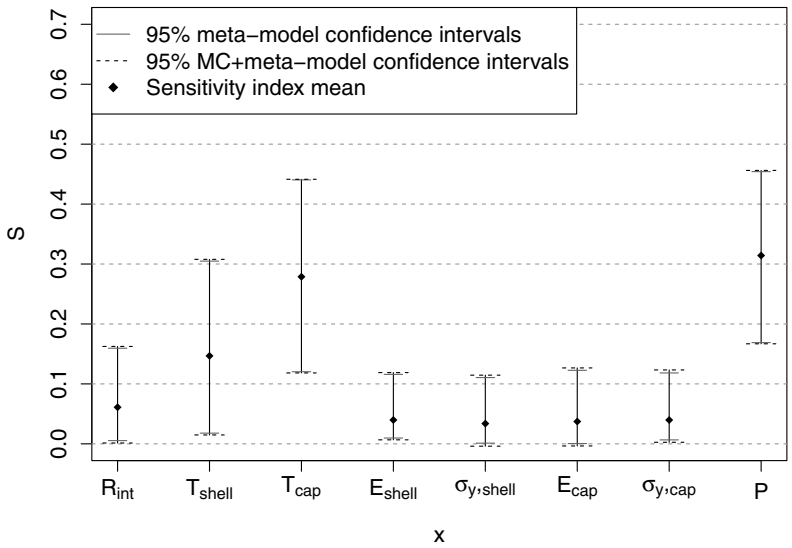

Figure 11. *Kriging-based sensitivity analysis for the expensive code. The diamonds represent the means of the first-order sensitivity index estimators, the solid gray lines represent the* 95% *confidence intervals taking into account only the metamodeling uncertainty, and the dashed lines represent the* 95% *confidence intervals taking into account the uncertainty due to both the Monte Carlo integration and the metamodeling.*

8. Conclusion. This paper deals with the sensitivity analysis of complex computer codes using Gaussian process regression. The purpose of the paper is to build Sobol index estimators taking into account both the uncertainty due to the surrogate modeling and that due to the numerical evaluations of the variances and covariances involved in the Sobol index definition. The aim is to provide relevant confidence intervals for the index estimator.

To provide such estimators, we suggest a method which mixes a Gaussian process regression model with Monte Carlo–based integration. From it, we can quantify the impact of both the Gaussian process regression and the Monte Carlo procedure on the index estimator variability. In particular, we present a procedure for balancing these two sources of uncertainty. Furthermore, we suggest numerical methods to avoid ill-conditioned problems and to easily handle the suggested index estimator.

Then, we propose an extension of the suggested approach for multifidelity computer codes. These codes possess the characteristic of having coarser but computationally cheaper versions. They are of practical interest since they allow for dealing with the problem of very expensive simulations. To deal with these codes, we use a multivariate Gaussian process regression model called *multifidelity cokriging*.

Finally, we perform several numerical tests which confirm the relevance of this new approach. We illustrate the suggested strategy on an academic example for the univariate case

and with a real application on a tank under internal pressure for the multifidelity analysis.

From this work, two points can naturally be investigated. First, we could improve the uncertainty quantification for the metamodel. Indeed, in this paper, we do not take into account the uncertainty due to the estimation of the hyperparameters of the covariance kernels. This can imply an underestimation of the predictive variance, and thus it can be worth inferring from these parameters. The natural way to perform such analysis is to use a full-Bayesian approach. Second, the metamodel considered is built from a fixed experimental design set. Several methods exist to sequentially add new points on the design in order to perform optimization, to quantify a probability of failure, or to improve the accuracy of the metamodel. However, there are no methods that focus on the error reduction of the sensitivity index estimates. It would be of practical interest to develop sequential design strategies for a sensitivity analysis purpose.

Acknowledgments. We thank Josselin Garnier for several discussions and Gilles Defaux for providing the data for the application. All the numerical tests have been performed within the R environment, by using the sensitivity, DiceKriging, and MuFiCokriging packages.

REFERENCES

[1] G.E.B. Archer, A. Saltelli, and I.M. Sobol, *Sensitivity measures, ANOVA-like techniques and the use of bootstrap*, J. Stat. Comput. Simul., 58 (1997), pp. 99–120.

[2] L.S. Bastos and A. O'Hagan, *Diagnostics for Gaussian process emulators*, Technometrics, 51 (2009), pp. 425–438.

[3] P.J. Bradley, *A multi-fidelity based adaptive sampling optimisation approach for the rapid design of double-negative metamaterials*, PIER B, 55 (2013), pp. 87–114.

[4] G. Chastaing, F. Gamboa, and C. Prieur, *Generalized Hoeffding-Sobol decomposition for dependent variables - Application to sensitivity analysis*, Electron. J. Statist., 6 (2012), pp. 2420–2448.

[5] J.P. Chilès and P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty*, Wiley Ser. Probab. Statist. Appl. Probab. Statist., John Wiley and Sons, New York, 1999.

[6] G. Damblin, M. Couplet, and B. Iooss, *Numerical studies of space-filling designs: Optimization of Latin Hypercube Samples and subprojection properties*, J. Simulation, 7 (2013), pp. 276–289.

[7] J.-C. Ferreira and V.A. Menegatto, *Eigenvalues of integral operators defined by smooth positive definite kernels*, Integral Equations Operator Theory, 64 (2009), pp. 61–81.

[8] R.B. Gramacy and M. Taddy, *Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models*, J. Stat. Softw., 33 (2012), pp. 1–48.

[9] J. Hegenderfer, S. Atamturktur, and A. Gillen, *Damage detection in steel structures using Bayesian calibration techniques*, in Topics in Modal Analysis II, Vol. 6, Springer, New York, 2012, pp. 179–193.

[10] D. Higdon, K. Heitmann, E. Lawrence, and S. Habib, *Using the Bayesian framework to combine simulations and physical observations for statistical inference*, in Large-Scale Inverse Problems and Quantification of Uncertainty, Wiley, Chichester, 2010, pp. 87–105.

[11] D. Higdon, M. Kennedy, J.C. Cavendish, J.A. Cafeo, and R.D. Ryne, *Combining field data and computer simulation for calibration and prediction*, SIAM J. Sci. Comput., 26 (2004), pp. 448–466.

[12] W. Hoeffding, *A class of statistics with asymptotically normal distribution*, Ann. Math. Statistics, 19 (1948), pp. 293–325.

[13] B. Iooss, F. Van Dorpe, and N. Devictor, *Response surfaces and sensitivity analyses for an environmental model of dose calculations*, Reliab. Eng. Syst. Saf., 91 (2006), pp. 1241–1251.

[14] J. Jacques, C. Lavergne, and N. Devictor, *Sensitivity analysis in presence of model uncertainty and correlated inputs*, Reliab. Eng. Syst. Saf., 91 (2006), pp. 1126–1134.

[15] A. Janon, T. Klein, A. Lagnoux, M. Nodet, and C. Prieur, *Asymptotic normality and efficiency of two Sobol index estimators*, ESAIM Probab. Stat., to appear; available online from http://dx.doi.org/10.1051/ps/2013040.
[16] A. Janon, M. Nodet, and C. Prieur, *Uncertainties assessment in global sensitivity indices estimation from metamodels*, Int. J. Uncertain. Quantif., 4 (2014), pp. 21–36.
[17] M.C. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, Biometrika, 87 (2000), pp. 1–13.
[18] M.C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 63 (2001), pp. 425–464.
[19] H. König, *Eigenvalue Distribution of Compact Operators*, Birkhäuser, Basel, 1986.
[20] S. Kucherenko, S. Tarantola, and P. Annoni, *Estimation of global sensitivity indices for models with dependent variables*, Comput. Phys. Commun., 183 (2012), pp. 937–946.
[21] B. Malouin, J.-Y. Trépanier, and M. Gariépy, *Interpolation of transonic flows using a proper orthogonal decomposition method*, Int. J. Aero. Eng., 2013 (2013), 928904.
[22] A. Marrel, B. Iooss, B. Laurent, and O. Roustant, *Calculations of Sobol indices for the Gaussian process metamodel*, Reliab. Eng. Syst. Saf., 94 (2009), pp. 742–751.
[23] A. Marrel, B. Iooss, F. Van Dorpe, and E. Volkova, *An efficient methodology for modeling complex computer codes with Gaussian processes*, Comput. Statist. Data Anal., 52 (2008), pp. 4731–4744.
[24] M. Marseguerra, R. Masini, E. Zio, and G. Cojazzi, *Variance decomposition-based sensitivity analysis via neural networks*, Reliab. Eng. Syst. Saf., 79 (2003), pp. 229–238.
[25] J.E. Oakley and A. O'Hagan, *Probabilistic sensitivity analysis of complex models a Bayesian approach*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 66 (2004), pp. 751–769.
[26] A. O'Hagan, *Bayesian analysis of computer code outputs: A tutorial*, Reliab. Eng. Syst. Saf., 91 (2006), pp. 1290–1300.
[27] P.Z.G. Qian and C.F.J. Wu, *Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments*, Technometrics, 50 (2008), pp. 192–204.
[28] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
[29] C.S. Reese, A.G. Wilson, M. Hamada, H.F. Martz, and K.J. Ryan, *Integrated analysis of computer and physical experiments*, Technometrics, 46 (2004), pp. 153–164.
[30] A. Saltelli, K. Chan, and E.M. Scott, *Sensitivity Analysis*, Wiley Ser. Probab. Stat., Wiley, Chichester, 2000.
[31] T.J. Santner, B.J. Williams, and W.I. Notz, *The Design and Analysis of Computer Experiments*, Springer, New York, 2003.
[32] I.M. Sobol, *Sensitivity estimates for non linear mathematical models*, Math. Modeling Comput. Experiment, 1 (1993), pp. 407–414.
[33] I.M. Sobol, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Math. Comput. Simulation, 55 (2001), pp. 271–280.
[34] I.M. Sobol, S. Tarantola, D. Gatelli, S.S. Kucherenko, and W. Mauntz, *Estimating the approximation error when fixing unessential factors in global sensitivity analysis*, Reliab. Eng. Syst. Saf., 92 (2007), pp. 957–960.
[35] M.L. Stein, *Interpolation of Spatial Data*, Springer Ser. Statist., Springer, New York, 1999.
[36] C.B. Storlie, L.P. Swiler, J.C. Helton, and C.J. Sallaberry, *Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models*, Reliab. Eng. Syst. Saf., 94 (2009), pp. 1735–1763.