Uncertainties assessment in global sensitivity indices estimation from metamodels

Alexandre Janon, Maëlle Nodet, Clémentine Prieur

Uncertainties assessment in global sensitivity indices estimation from metamodels

Alexandre Janon, Maëlle Nodet, Clémentine Prieur
Joseph Fourier University, Laboratoire Jean Kuntzmann, MOISE team,
BP 53, 38041 Grenoble Cedex, France

May 29, 2012

Abstract: Global sensitivity analysis is often impracticable for complex and resource intensive numerical models, as it requires a large number of runs. The metamodel approach replaces the original model by an approximated code that is much faster to run. This paper deals with the information loss in the estimation of sensitivity indices due to the metamodel approximation. A method for providing a robust error assessment is presented, hence enabling significant time savings without sacrificing on precision and rigor. The methodology is illustrated on two different types of metamodels: one based on reduced basis, the other one on RKHS interpolation.

Contents

1 Introduction

Many mathematical models use a large number of poorly-known parameters as inputs. The impact of parameter uncertainty on the model output is important for the user of these models. This problem can be tackled by considering the uncertain input parameters as random variables, whose probability distribution reflects the practitioner's belief in the precision of the parameter values. Model output, as function of the model inputs, is then a random variable. Its probability distribution, uniquely determined by the model and the distribution of the inputs, can give detailed and valuable information about the behavior of the output when input parameters vary: range of attained values, mean value and dispersion about the mean, most probable values (modes), *etc.*

Sensitivity analysis aims to identify the sensitive parameters, that is, parameters for which a small variation implies a large variation of the model output. In global sensitivity analysis, one makes use of the probability distribution of the outputs to define (amongst other sensitivity measures) *sensitivity indices* (also known as *Sobol indices*). The sensitivity index of an output with respect to an input variable is the fraction of output variance which can be "explained" by the variation of the input variable, either alone (one then speaks about main effect), or in conjunction with other input variables (total effect). This way, input variables can be sorted by the order of importance they have on the output. One can also consider the proportion of variance due to the variation of groups of two or more inputs, although main effects and total effects are generally sufficient to produce a satisfying sensitivity analysis; see, e.g., [11, 18] for more information about uncertainty and sensitivity analysis.

Once these indices have been defined, the question of their effective calculation remains open. For most models, an exact, analytic computation is not attainable (even expressing an output as an analytic function of the inputs is infeasible) and one has to use numerical approximations.

A robust, popular way to obtain such approximations is Monte Carlo estimation. This method simulates randomness in inputs by sampling a large number of parameter values (from the selected input distribution). The model output is then computed for each sampled value of the parameters.

This way, one obtains a sample of outputs, under the conjugate action of the model and the input distribution. A suitable statistical estimator can then be applied to form a numerical estimate of the sensitivity index based on the sample of outputs. The Monte Carlo approach to Sobol indices computation is described in [22], together with improvements in [12, 17].
A major drawback of the Monte Carlo estimation is that a large number of model outputs have to be evaluated for the resulting approximation of the sensitivity index to be accurate enough to be useful. In complex models, in which a simulation for one single value of the parameters may take several minutes, using these methods "as-is" is impracticable. In those cases, one generally makes use of a *surrogate model* (also known as *reduced model*, *emulator*, *metamodel* or *response surface*). The surrogate model has to approximate the original model (called the *full* model) well, while being many times faster to evaluate. The sensitivity index is then calculated *via* a sample of outputs, each generated by a call to the surrogate model, thus accelerating the overall computation. The aim of this paper is to quantify accuracy loss due to the use of a metamodel combined to a Monte-Carlo method to compute sensitivity indices.
The sensitivity index produced by Monte Carlo estimation with a surrogate model is tainted by a twofold error. Firstly, our Monte-Carlo sampling procedure assimilates the whole (generally infinite) population of possible inputs with the finite, randomly chosen, sample; this produces *sampling*, or *Monte-Carlo error*. Secondly, using a surrogate model biases the estimation of the Sobol index, as what is actually estimated is sensitivity of surrogate output, and not the full one; we call this bias the *metamodel error*.
A variation on the bootstrap, which addresses sampling error as well as metamodel error, has been proposed in [23]; in this work, the authors propose to use a bootstrap strategy on the metamodel residuals to estimate metamodel error and Monte-Carlo error simultaneously. In [14], confidence intervals for the Sobol index are estimated using the conditional distribution of the Kriging predictor given the learning sample. This approach is limited to Kriging metamodels. Finally, the paper by [2] makes use of the reduced-basis output error bound to certify computation of the expectation and the variance of a model output (and not, as in this paper, the Sobol indices) with neglected sampling error.
In this paper, we want to make a rigorous sensitivity analysis, so it is important to assess the magnitude of these two combined errors on the estimated sensitivity indices. We will also use such assessment to help with the choice of correct approximation parameters (Monte-Carlo sample size and metamodel fidelity) to achieve a desired precision in estimated indices.
We estimate sampling error by using a classical method, which comes at a moderate numerical cost: bootstrap resampling ([8, 1]). Based on statistical estimation theory, the bootstrap technique involves the generation of a sample of sensitivity index estimator replications, whose empirical distribution

serves as approximation of the true (unknown) estimator distribution, in order to produce asymptotic confidence intervals that give good results in many practical cases.

To estimate metamodel error, we will use the surrogate models coming with *error bounds*, that is, computable (or at least estimable) upper bounds on the error between the original and the surrogate outputs. The reduced basis (RB) method ([15, 10, 25, 9]) is a method leading to such rigorously certified metamodels; it is applicable when the original model is a discretization of a partial differential equation (PDE) depending on the input parameters. Kriging ([19]) – also known as Gaussian process metamodelling, which is equivalent to RKHS (reproducing kernel Hilbert space) interpolation ([20]) – also provides error indicators. In contrast to the RB method, Kriging/RKHS interpolation only requires "blackbox" training data, i.e. a (finite) set of input-output pairs from the original model. This makes it more versatile and easier to use, at the expense of rigor in the error bounds and quality in the approximation.

Our new approach is based on the separation of the estimation of the metamodel and the sampling error. The advantages brought by our approach are: its rigorousness (the impact of the use of a surrogate model is provably bounded); its efficiency (our bounds are rather sharp, and go to zero when metamodel errors decrease); its moderate computational requirements (time is better spent on making a precise computation than at measuring precision) and its versatility with respect to metamodel choice (the user can choose any metamodel that comes with computable error bound or error indicator). In other words, our method allows us to estimate sensitivity indices by using a metamodel which greatly speeds up computation times, while rigorously keeping track of the precision of the estimation. The model has to exhibit some regularity which can be captured by a metamodel. Moreover, the metamodel error bound should not be too pessimistic.

This paper is organized as follows: in the first part, we describe the prerequisites for our approach: we give the definition of the standard Monte Carlo estimator of the sensitivity indices of interest; in the second part, we present our confidence interval estimation technique for the sensitivity index, which accounts for the two sources of error described earlier (sampling and metamodel). In the third part, we present applications of our method for reduced-basis and RKHS/Kriging interpolation metamodels, and compare our method with the method described in [23]. Notations are gathered in B.

2 Review of sensitivity indices

2.1 Definition

In order to define sensitivity indices, we choose a probability distribution for the input variables, considering each input variable X_i ($i = 1, \ldots, p$) as a random variable with known distribution; the model output $Y = f(X_1, \ldots, X_p)$ (assumed to be square-integrable, non a.s. constant and scalar: multiple outputs can be treated separately) is thus for $\mathbf{X} = (X_1, \ldots, X_p)$ a $\sigma(\mathbf{X})$-measurable random variable (assuming that f is a $\sigma(\mathbf{X})$-measurable function). We further assume that the X_i's are mutually independent. We also fix throughout all this paper an input variable of interest $1 \leq i \leq p$. We define the *first-order main effect* of X_i on Y by:

$$S_i = \frac{\mathrm{Var}\mathbf{E}(Y|X_i)}{\mathrm{Var}Y} \tag{1}$$

S_i is the sensitivity index in which we are interested in this paper but other indices (total effect, high-order effects) exist and our methodology can readily be extended to these indices.

2.2 Monte-Carlo estimator

We are interested in the following Monte-Carlo estimator for S_i ([12, 17]): a sample size $N \in \mathbf{N}$ being given, let $\left\{\mathbf{X}^k\right\}_{k=1,\ldots,N}$ and $\left\{\mathbf{X}'^k\right\}_{k=1,\ldots,N}$ be two random i.i.d. samples of size N each, drawn from the distribution of the input vector $\mathbf{X}$.
For $k = 1, \ldots, N$, we note:

$$y_k = f(\mathbf{X}^k) \tag{2}$$

and:

$$y'_k = f(X'^k_1, \ldots, X'^k_{i-1}, X^k_i, X'^k_{i+1}, \ldots, X'^k_p) \tag{3}$$

The Monte-Carlo estimator of S_i is then given by:

$$\widehat{S_i} = \frac{\overline{yy'} - \overline{y} \times \overline{y'}}{\overline{y^2} - \overline{y}^2} \tag{4}$$

where, for any vector $v = (v_1, \ldots, v_N)$,

$$\overline{v} = \frac{1}{N} \sum_{k=1}^{N} v_i$$

It can be shown that $\widehat{S_i}$ is a strongly consistent estimator of S_i.
Remark: our methodology can be extended to higher order Sobol indices by choosing primed output sample $\{y'_k\}$ appropriately. More specifically, for a

subset of variables $I \subset \{1, \ldots, p\}$, the closed index with respect to I, defined by:

$$S_I = \frac{\mathrm{Var}\mathbf{E}(Y|(X_i)_{i\in I})}{\mathrm{Var}Y}$$

is estimated by using:

$$y'_k = f(\mathbf{X}'^k_I), \text{ where } X'^k_{I,i} = \begin{cases} X^k_i \text{ if } i \in I \\ X'^k_i \text{ else} \end{cases}.$$

and standard high-order indices can be treated by subtracting the effects of the proper subsets of I.

3 Quantification of the two types of error in index estimation

We now present our method for estimating the two types of error that occur in Monte-Carlo sensitivity index estimation on a reduced-basis metamodel. In Sections 3.1 and 3.2, we show two approaches for taking metamodel error in account. In Section 3.3, we review the bootstrap, which we will use for the treatment of sampling error. Section 3.4 shows how to combine metamodel error and Monte-Carlo estimation in order to provide the final index interval estimation.

3.1 Metamodel error in index estimation

We now denote by $\widetilde{f} : \mathcal{P} \subseteq \mathbf{R}^p \to \mathbf{R}$ the metamodel output approximating $f : \mathcal{P} \to \mathbf{R}$, and by ε the pointwise error bound that certifies the metamodel approximation, i.e. we have:

$$\left|f(\mathbf{x}) - \widetilde{f}(\mathbf{x})\right| \leq \varepsilon(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{P}$$

For a pair of samples $\left(\{\mathbf{X}^k\}_{k=1,\ldots,N}, \{\mathbf{X}'^k\}_{k=1,\ldots,N}\right)$ of inputs, we can use our metamodel output $\widetilde{f}$ and our metamodel error bound ε to compute, for $k = 1, \ldots, N$:

$$\widetilde{y}_k = \widetilde{f}(\mathbf{X}^k), \quad \widetilde{y}'_k = \widetilde{f}(X'^k_1, \ldots, X'^k_{i-1}, X^k_i, X'^k_{i+1}, \ldots, X'^k_p) \tag{5}$$

and:

$$\varepsilon_k = \varepsilon(\mathbf{X}^k), \quad \varepsilon'_k = \varepsilon(X'^k_1, \ldots, X'^k_{i-1}, X^k_i, X'^k_{i+1}, \ldots, X'^k_p) \tag{6}$$

In this section, we find accurate, explicitly and efficiently computable bounds $\widehat{S}^m_i$ and $\widehat{S}^M_i$, depending only on $\widetilde{y}_k, \widetilde{y}'_k, \varepsilon_k$ and ε'_k so that:

$$\widehat{S}^m_i \leq \widehat{S}_i \leq \widehat{S}^M_i \tag{7}$$

In other words, we want lower and upper bounds on the full model based sensitivity index estimator $\widehat{S}_i$ computable from surrogate model calls.
We now define, for any $\mathbf{z} = (z_1, \ldots, z_N, z'_1, \ldots, z'_N) \in \mathbf{R}^{2N}$ and any $a, t, t' \in \mathbf{R}$ the R function by:

$$R(a; \mathbf{z}, t, t') = \sum_{k=1}^{N} \left(z'_k - (a(z_k - t) + t')\right)^2 .$$

Let $\mathbf{y} = (y_1, \ldots, y_N, y'_1, \ldots, y'_N)$. By setting first derivative of R with respect to a to zero, making use of the convexity of $R(\cdot; \mathbf{y}, \overline{y}, \overline{y}')$ and using the definition (4) of $\widehat{S_i}$, one easily shows that:

$$\widehat{S}_i = \underset{a \in \mathbf{R}}{\operatorname{argmin}}\, R(a; \mathbf{y}, \overline{y}, \overline{y}'). \tag{8}$$

In other words, $\widehat{S}_i$ is the slope of the linear least squares regression of the $\{y'_k\}_k$ on the $\{y_k\}_k$. The key of our approach is to bound $R(a; \mathbf{y}, \overline{y}, \overline{y}')$ between two quantities which are computable without knowing $\mathbf{y}$, nor $\overline{y}$ and $\overline{y}'$ and by using (8) to deduce bounds for any realization (ie., evaluation on a sample) of $\widehat{S}_i$ which depend only on metamodel outputs and error bounds, that is $y_k, \varepsilon_k, y'_k, \varepsilon'_k, k = 1, \ldots, N$.
Recall that:

$$\widetilde{y}_k = \widetilde{f}(\mathbf{X}^k), \quad \widetilde{y}'_k = \widetilde{f}(X'^k_1, \ldots, X'^k_{i-1}, X^k_i, X'^k_{i+1}, \ldots, X'^k_p)$$

and:

$$\varepsilon_k = \varepsilon(\mathbf{X}^k), \quad \varepsilon'_k = \varepsilon(X'^k_1, \ldots, X'^k_{i-1}, X^k_i, X'^k_{i+1}, \ldots, X'^k_p).$$

Define:

$$R_{inf}(a; \widetilde{\mathbf{y}}, \boldsymbol{\varepsilon}, t, t') = \sum_{k=1}^{N} \left\{ \inf_{z_k \in [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k], z'_k \in [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k]} \left(z'_k - (a(z_k - t) + t')\right)^2 \right\} \tag{9}$$

and:

$$R_{sup}(a; \widetilde{\mathbf{y}}, \boldsymbol{\varepsilon}, t, t') = \sum_{k=1}^{N} \left\{ \sup_{z_k \in [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k], z'_k \in [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k]} \left(z'_k - (a(z - t) + t')\right)^2 \right\} \tag{10}$$

where $\widetilde{\mathbf{y}} = (\widetilde{y}_1, \ldots, \widetilde{y}_N, \widetilde{y}'_1, \ldots, \widetilde{y}'_N)$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N, \varepsilon'_1, \ldots, \varepsilon'_N)$.
It is clear that:

$$R_{inf}(a; \widetilde{\mathbf{y}}, \boldsymbol{\varepsilon}, t, t') \leq R(a; \mathbf{y}, t, t') \leq R_{sup}(a; \widetilde{\mathbf{y}}, \boldsymbol{\varepsilon}, t, t') \quad \forall a, t, t' \in \mathbf{R} \tag{11}$$

Note that R, R_{inf} and R_{sup} are quadratic polynomials in a. We name $\alpha, \beta, \gamma, \alpha_{inf}, \beta_{inf}, \gamma_{inf}, \alpha_{sup}, \beta_{sup}$ and γ_{sup} their respective coefficients. In other words, we have:

$$R(a; \mathbf{y}, t, t') = \alpha a^2 + \beta a + \gamma$$

$$R_{inf}(a;\widetilde{\mathbf{y}},\boldsymbol{\varepsilon},t,t') = \alpha_{inf}a^2 + \beta_{inf}a + \gamma_{inf} \tag{12}$$

$$R_{sup}(a;\widetilde{\mathbf{y}},\boldsymbol{\varepsilon},t,t') = \alpha_{sup}a^2 + \beta_{sup}a + \gamma_{sup} \tag{13}$$

These coefficients are computable by evaluating $R_{inf}(a;\widetilde{\mathbf{y}},\boldsymbol{\varepsilon},t,t')$ and $R_{sup}(a;\widetilde{\mathbf{y}},\boldsymbol{\varepsilon},t,t')$ for three different values of a and solve for the interpolating quadratic functions. These coefficients depend on t and t', as well on $\mathbf{y}$ (for α,β,γ) and $\widetilde{\mathbf{y}}$ and $\boldsymbol{\varepsilon}$ (for the other coefficients). We do not explicitly write this dependence until the last part of our discussion.
Using (11) we see that the quadratic function of a:

$$(\alpha_{inf}-\alpha)a^2 + (\beta_{inf}-\beta)a + \gamma_{inf} - \gamma$$

is negative or zero; hence it takes a non-positive value for $a=0$, and has a non-positive discriminant:

$$\gamma_{inf} - \gamma \leq 0 \tag{14}$$

$$(\beta_{inf}-\beta)^2 \leq 4(\alpha_{inf}-\alpha)(\gamma_{inf}-\gamma) \tag{15}$$

As $(\beta_{inf}-\beta)^2 \geq 0$, Equations (14) and (15) above imply that $\alpha_{inf}-\alpha \leq 0$, and that:

$$\beta_{inf} - \delta_{inf} \leq \beta \leq \beta_{inf} + \delta_{inf}$$

for $\delta_{inf} = 2\sqrt{(\alpha_{inf}-\alpha)(\gamma_{inf}-\gamma)}$.
We now suppose that $\alpha_{inf} > 0$. As α_{inf} is computable from $\widetilde{y}_k, \widetilde{y}'_k, \varepsilon_k$ and ε'_k, one can practically check if this condition is met. If it is not the case, our bound can not be used. We expect that if the metamodel error is not too large, we will have $\alpha_{inf} \approx \alpha$ and, as $\alpha > 0$, the hypothesis $\alpha_{inf} > 0$ is realistic.
So, under this supplementary assumption, we have:

$$\operatorname*{argmin}_a R(a;\mathbf{y},t,t') = -\frac{\beta}{2\alpha} \geq -\frac{\beta_{inf}+\delta_{inf}}{2\alpha_{inf}}$$

Now using the second part of (11) and the same reasoning on the non-positive quadratic function of a: $R(a;\mathbf{y},t,t') - R_{sup}(a;\widetilde{\mathbf{y}},\boldsymbol{\varepsilon},t,t')$, we find that: $\alpha \leq \alpha_{sup}$, and: $\beta_{sup} - \delta_{sup} \leq \beta \leq \beta_{sup} + \delta_{sup}$. Hence,

$$\operatorname*{argmin}_a R(a;\mathbf{y},t,t') \leq -\frac{\beta_{sup}-\delta_{sup}}{2\alpha_{sup}}$$

where $\delta_{sup} = 2\sqrt{(\alpha-\alpha_{sup})(\gamma-\gamma_{sup})}$. This comes without supplementary assumptions, because $\alpha_{sup} \geq \alpha$ and $\alpha > 0$, or else $R(\cdot;\mathbf{y},t,t')$ would have no minimum (even if the case $\alpha = 0$ can seldom occur due to sampling fluctuations, the hypothesis Y not constant a.s. ensures that increasing N and/or changing the sample will lead to a case where $\alpha > 0$).

As we clearly have δ_{inf} and δ_{sup} smaller than (or equal to) $\widehat{\delta} := 2\sqrt{(\alpha_{inf} - \alpha_{sup})(\gamma_{inf} - \gamma_{sup})}$, we deduce that:

$$-\frac{\beta_{inf}(t,t') + \widehat{\delta}(t,t')}{2\alpha_{inf}(t,t')} \leq \underset{a}{\operatorname{argmin}}\, R(a;\mathbf{y},t,t') \leq -\frac{\beta_{sup}(t,t') - \widehat{\delta}(t,t')}{2\alpha_{sup}(t,t')}$$

where we have made explicit the dependencies in t and t'.
To finish, it is easy to see that we have:

$$\overline{\mathcal{P}} := [\overline{\widetilde{y}} - \overline{\varepsilon}; \overline{\widetilde{y}} + \overline{\varepsilon}] \ni \overline{y} \tag{16}$$

and:

$$\overline{\mathcal{P}'} := [\overline{\widetilde{y}'} - \overline{\varepsilon}'; \overline{\widetilde{y}'} + \overline{\varepsilon}'] \ni \overline{y}' \tag{17}$$

(where $\overline{\widetilde{y}}, \overline{\widetilde{y}'}, \overline{\varepsilon}$ and $\overline{\varepsilon'}$ denote, respectively, the empirical means of $(\widetilde{y}_k)_k$, $(\widetilde{y}'_k)_k$, $(\varepsilon_k)_k$ and $(\varepsilon'_k)_k$) so that:

$$\min_{t\in\overline{\mathcal{P}},t'\in\overline{\mathcal{P}}'} \left(-\frac{\beta_{inf}(t,t') + \widehat{\delta}(t,t')}{2\alpha_{inf}(t,t')}\right) \leq \widehat{S}_i = \underset{a}{\operatorname{argmin}}\, R(a;\mathbf{y},t,t') \leq \max_{t\in\overline{\mathcal{P}},t'\in\overline{\mathcal{P}}'} \left(-\frac{\beta_{sup}(t,t') - \widehat{\delta}(t,t')}{2\alpha_{sup}(t,t')}\right)$$

Hence, (7) is verified with:

$$\widehat{S}_i^m = \min_{t\in\overline{\mathcal{P}},t'\in\overline{\mathcal{P}}'} \left(-\frac{\beta_{inf}(t,t') + \widehat{\delta}(t,t')}{2\alpha_{inf}(t,t')}\right), \quad \widehat{S}_i^M = \max_{t\in\overline{\mathcal{P}},t'\in\overline{\mathcal{P}}'} \left(-\frac{\beta_{sup}(t,t') - \widehat{\delta}(t,t')}{2\alpha_{sup}(t,t')}\right) \tag{18}$$

It is clear that $\widehat{S}_i^m$ and $\widehat{S}_i^M$ are computable without knowing the y_ks and y'_ks.
In practice, we compute approximate values of $\widehat{S}_i^m$ and $\widehat{S}_i^M$ by replacing the min and max over $\overline{\mathcal{P}} \times \overline{\mathcal{P}'}$ by the min and max over a finite sample $\Xi \subset \overline{\mathcal{P}} \times \overline{\mathcal{P}'}$.

3.2 Metamodel error in index estimation: a smoothed alternative

The bounds we presented in the last section are sometimes uninteresting, as, when ε is large enough with respect to f, the bounds $\widehat{S_i^m}$ and $\widehat{S_i^M}$ are not tight enough to be useful: it happens for example that $[0;1] \subset [\widehat{S_i^m}; \widehat{S_i^M}]$. Such a case is the RKHS interpolation metamodel example we present in Section 4.2.
In this section, we present an alternative to the bound described in the previous section. This alternative is interesting when ε is only moderately small with respect to f.
We begin by defining $\psi_i(\mathbf{z})$ as the Monte-Carlo sensitivity index estimator for variable i using $\mathbf{z} = (z_1, \ldots, z_N, z'_1, \ldots, z'_N)$ as function evaluations sample:

$$\psi_i(\mathbf{z}) = \frac{\overline{zz'} - \overline{z}\,\overline{z'}}{\overline{z^2} - \overline{z}^2}.$$

It is clear that:

$$\psi_i(y_1, \ldots, y_N, y'_1, \ldots, y'_N) = \widehat{S}_i,$$

where $\{y_k\}$ and $\{y'_k\}$ are defined in (2) and (3).
Also let:

$$\mathcal{Z} = \prod_{k=1}^{N} [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k] \times \prod_{k=1}^{N} [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k] \,.$$

Now, since:

$$\forall k = 1, \ldots, N \quad y_k \in [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k] \text{ and } y'_k \in [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k],$$

we certainly have $(y_1, \ldots, y_N, y'_1, \ldots, y'_N) \in \mathcal{Z}$ and hence:

$$\min_{\mathbf{z} \in \mathcal{Z}} \psi_i(\mathbf{z}) \leq \widehat{S}_i \leq \max_{\mathbf{z} \in \mathcal{Z}} \psi_i(\mathbf{z}). \tag{19}$$

Minimizers and maximizers of optimization problems in (19) often display a very irregular (nonsmooth) behavior (as a function of the sampled inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}'_1, \ldots, \mathbf{x}'_N\}$), even for a smooth output f. This leads to overly pessimistic bounds in (19). To overcome this difficulty, we propose to ensure smoothness of the solution by introducing a penalty factor $\lambda \geq 0$ and to take:

$$S_i^{m,\lambda} = \psi_i(\mathbf{z}^{m,\lambda}) \quad S_i^{M,\lambda} = \psi_i(\mathbf{z}^{M,\lambda})$$

where:

$$\mathbf{z}^{m,\lambda} = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}} (\psi_i(\mathbf{z}) + \lambda \Pi(\mathbf{z})) \quad \mathbf{z}^{M,\lambda} = \operatorname*{argmax}_{\mathbf{z} \in \mathcal{Z}} (\psi_i(\mathbf{z}) - \lambda \Pi(\mathbf{z})) \tag{20}$$

for $\Pi(\mathbf{z})$ a (non-negative) indicator of the smoothness of the function which take $\mathbf{z}$ as values when evaluated on the input sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}'_1, \ldots, \mathbf{x}'_N\}$. For instance, one may take:

$$\Pi(\mathbf{z}) = \frac{1}{2N} \sum_{k=1}^{2N} \left(z_k - z_k^S \right)^2 \tag{21}$$

with $\{z_k^S\}$ a kernel-smoothed version of $\{z_k\}$:

$$z_k^S = \frac{\sum_{k'=1}^{2N} K\left(\frac{\|\mathbf{x}_{k'} - \mathbf{x}_k\|}{h} \right) z_{k'}}{\sum_{k'=1}^{2N} K\left(\frac{\|\mathbf{x}_{k'} - \mathbf{x}_k\|}{h} \right)} \tag{22}$$

for K an appropriate kernel (preferably with compact support, for reasons of efficiency of the implementation), h a suitable bandwidth (in the sum we have set $\mathbf{x}_{k+N} = \mathbf{x}'_k$ for $k = 1, \ldots, N$), and $\|\cdot\|$ the Euclidean norm on $\mathbf{R}^p$. The main difficulty is that these optimization problems are in potentially large dimension $2N$. However, the gradient of the objective functions are

analytically available – under (21) and (22) – and can be cheaply evaluated, so we choose to use a L-BFGS quasi-Newton algorithm, as implemented in `L-BFGS-B` ([26]), to solve problems (20).
Concerning the choice of λ and h, we opt for the following: h should be chosen so as to have a reasonable proportion (say, 1 to 5 percent) of "neighbor" points (that is to say, couples of points in $\{\mathbf{x}_k\}_{k=1,\dots,N}$ having a significantly nonzero value on the kernel K). For the choice of λ: we plot, for a chosen h held fixed, the length $\left|S_i^{M,\lambda} - S_i^{m,\lambda}\right|$ as function of λ, and then choose λ to be the abscissa of the bottom-left corner of the "L-shaped" curve obtained.

3.3 Sampling error : bootstrap confidence intervals

Sampling error, due to the Monte-Carlo evaluation of the variances in (1), can be quantified through an approximate confidence interval calculated using bootstrap ([1]).
We use the bias-corrected (BC) percentile method presented in [6, 7]. The principle of this method can be summed up the following way: let $\widehat{\theta}(X_1, \dots, X_n)$ be an estimator for an unknown parameter θ in a reference population $\mathcal{P}$. We generate a random i.i.d. n-sample $\{x_1, \dots, x_n\}$ from $\mathcal{P}$, then we repeatedly, for $b = 1, \dots, B$, randomly draw $\{x_1[b], \dots, x_n[b]\}$ with replacement from this sample and get a *replication* of $\widehat{\theta}$ by computing $\widehat{\theta}[b] = \widehat{\theta}(x_1[b], \dots, x_n[b])$. This way we obtain a set $\mathcal{R} = \{\widehat{\theta}[1], \dots, \widehat{\theta}[B]\}$ of replications of $\widehat{\theta}$.
We now show how this sample can be used to estimate a confidence interval for θ. We denote by Φ the standard normal cdf:

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t$$

and by Φ^{-1} its inverse.
Using $\mathcal{R}$ and the point estimate $\widehat{\theta} = \widehat{\theta}(x_1, \dots, x_n)$, a "bias correction constant" z_0 can be estimated:

$$\widehat{z_0} = \Phi^{-1}\left(\frac{\#\{\widehat{\theta}[b] \in \mathcal{R} \text{ s.t. } \widehat{\theta}[b] \leq \widehat{\theta}\}}{B}\right)$$

where $\#A$ denotes the number of elements in the set A.
Then, for $\beta \in]0;1[$, we define the "corrected quantile estimate" $\widehat{q}(\beta)$:

$$\widehat{q}(\beta) = \Phi(2\widehat{z_0} + z_\beta)$$

where z_β satisfies $\Phi(z_\beta) = \beta$.
The central BC bootstrap confidence interval of level $1-\alpha$ is then estimated by the interval whose endpoints are the $\widehat{q}(\alpha/2)$ and $\widehat{q}(1-\alpha/2)$ quantiles of $\mathcal{R}$.
The BC bootstrap procedure is known to be robust to non-normality of $\widehat{\theta}$.
The key advantage of bootstrapping our sensitivity estimators is that we

do not require supplementary model evaluations to estimate a confidence interval; hence the computational overhead for getting a confidence interval (versus pointwise estimation only) remains quite modest.

3.4 Combined confidence intervals

From Section 3.1 to 3.3, we have seen how to separately assess sampling error and metamodel error. To take both sources of error into account simultaneously, we propose using bootstrap confidence intervals (see Section 3.3) by calculating B bootstrap replications of $\widehat{S}_i^m$ and $\widehat{S}_i^M$, where, for $b = 1, \ldots, B$ each bootstrap pair $\left(\widehat{S}_i^m[b]; \widehat{S}_i^M[b]\right)$ is computed using $(\widetilde{y}_k)_{k\in L_b}, (\widetilde{y}'_k)_{k\in L_b}$ as surrogate output samples, and associated error bounds $(\widetilde{\varepsilon}_k)_{k\in L_b}, (\widetilde{\varepsilon}'_k)_{k\in L_b}$, where L_b is a list of N integers sampled with replacement from $\{1, \ldots, N\}$. Here the bounds $\widehat{S}_i^m$ and $\widehat{S}_i^M$ can be computed using either the method described in Section 3.1, or the one described in Section 3.2.
The BC bootstrap confidence interval procedure (see Section 3.3) can then be used to produce $(1-\alpha)$-level confidence intervals $[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^m_{i,1-\alpha/2}]$ for S_i^m, and $[\widehat{S}^M_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}]$ for S_i^M. We then take as combined confidence interval of level $1-\alpha$ for S_i the range $[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}]$. This interval accounts for sampling and metamodels error simultaneously: let, for $b = 1, \ldots, B$, $\widehat{S}_i[b]$ be the b^{th} bootstrap replication, computed using the true outputs $(y_k)_{k\in L_b}$. As, for each b, we have:

$$\widehat{S}_i^m[b] \leq \widehat{S}_i[b] \leq \widehat{S}_i^M[b],$$

it follows that

$$\widehat{S}^m_{i,\alpha/2} \leq \widehat{S}_{i,\alpha/2}, \text{ and } \widehat{S}_{i,1-\alpha/2} \leq \widehat{S}^M_{i,1-\alpha/2},$$

where $\widehat{S}_{i,\alpha/2}$ and $\widehat{S}_{i,1-\alpha/2}$ are the endpoints of the $(1-\alpha)$ BC-confidence interval computed using $(\widehat{S}_i[b])_b$ as replications. It follows that $[\widehat{S}_{i,\alpha/2}; \widehat{S}_{i,1-\alpha/2}]$ is contained in $[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}]$ and, hence, the level of the latter interval is greater than the level of the former. As the level of the $[\widehat{S}_{i,\alpha/2}; \widehat{S}_{i,1-\alpha/2}]$ is (asymptotically) equal to $1-\alpha$, the asymptotic level of the combined confidence interval $[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}]$ should theoretically be, by design, greater than $1-\alpha$.

4 Applications

4.1 Application to a reduced basis metamodel

The reduced basis method ([15, 16]) can be applied when the output is a functional of the discretized solution of a partial differential equation (PDE).

In particular cases (for instance [15] for elliptic equations, [13] for viscous Burgers equation, [24] for parabolic equations), it has been shown that the reduced basis output error bounds are accurate and useful. Due to the intrusive nature of the reduced basis approach and "problem-dependent" considerations made during the construction of the metamodel, one can expect fastly-convergent, tight and fully justified error bounds for the sensitivity indices.
In this section, we test our combined confidence interval procedure described in Sections 3.1 and 3.4, and compare it with Monte-Carlo estimation on the full model (with bias-corrected bootstrap to assess sampling error). Our criteria for comparison are the CPU times needed to compute the intervals and the lengths of these intervals (the smaller the better). Note that the error bounds given by the reduced basis method are sufficiently small so that the "smoothness" bounds described in Section 3.2 give similar results to the method of Section 3.1.
In all our tests we take $\alpha = .05$ and $B = 2000$ bootstrap replications. We checked that this value of B is large enough by increasing B (ie., setting $B = 4000$), and notice that our results remain significantly unchanged.

4.1.1 Model set-up

Let u, a function of space $x \in [0;1]$ (note that space variable x is unrelated to input parameter vector $\mathbf{x}$) and time $t \in [0,T]$ ($T > 0$ is a fixed (i.e., known) parameter) satisfying the *viscous Burgers' equation*:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}(u^2) - \nu\frac{\partial^2 u}{\partial x^2} = \psi \tag{23}$$

where $\nu \in \mathbf{R}_*^+$ is the *viscosity*, and $\psi \in C^0\left([0,T], L^2(]0,1[)\right)$ is the *source term*.
For u to be well-defined, we also prescribe initial value $u_0 \in H^1(]0,1[)$ (ie. u_0 is in the Sobolev space of square-integrable functions whose first derivative is square-integrable), and continuous boundary values $b_0, b_1 \in C^0([0,T])$.
The initial u_0 and boundary values b_0 and b_1 are parametrized the following way:

$$b_0(t) = (u_{0m})^2 + \sum_{l=1}^{n(b_0)} A_l^{b_0} \sin(\omega_l^{b_0} t) \qquad b_1(t) = u_0(1) + \sum_{l=1}^{n(b_1)} A_l^{b_1} \sin(\omega_l^{b_1} t)$$

$$\psi(t,x) = \psi_m + \sum_{l=1}^{n_T(\psi)} \sum_{p=1}^{n_S(\psi)} A_{lp}^{\psi} \sin(\omega_l^{\psi T} t) \sin(\omega_p^{\psi S} x) \qquad u_0(x) = (u_{0m})^2 + \sum_{l=1}^{n(u_0)} A_l^{u_0} \sin(\omega_l^{u_0} x)$$

The values of the angular frequencies $\omega_l^{b_0}$, $\omega_l^{b_1}$, $\omega_l^{\psi T}$, $\omega_p^{\psi S}$ and $\omega_l^{u_0}$, as well as their cardinalities $n(b_0)$, $n(b_1)$, $n_T(\psi)$, $n_S(\psi)$ and $n(u_0)$ are fixed (known), while our uncertain parameters, namely: viscosity ν, coefficients ψ_m and

u_{0m}, and amplitudes $\left(A_l^{b_0}\right)_{l=1,\dots,n(b_0)}$, $\left(A_l^{b_1}\right)_{l=1,\dots,n(b_1)}$, $\left(A_{lp}^{\psi}\right)_{l=1,\dots,n_T(\psi);p=1,\dots,n_S(\psi)}$ and $(A_l^{u_0})_{l=1,\dots,n(u_0)}$ live in some Cartesian product of intervals $\mathcal{P}$ defined by:

$$\mathcal{P} = \Big\{\mathbf{x} = (\nu, A_1^{b_0}, \dots, A_{n(b_0)}^{b_0}, A_1^{b_1}, \dots, A_{n(b_1)}^{b_0}, \psi_m, A_{11}^{\psi}, A_{12}^{\psi}, \dots, A_{1,n_S(\psi)}^{\psi}, A_{2,1}^{\psi}, \dots, A_{2,n_S(\psi)}^{\psi}, \dots, A_{n_T(\psi),n_S(\psi)}^{\psi}, u_{0m}, A_1^{u_0}, \dots, A_{n(u_0)}^{u_0})\Big\} \quad (24)$$

The solution $u = u(\mathbf{x})$ depends on the parameter vector $\mathbf{x}$ above.
The "full" model is obtained by discretizing the initial-boundary value problem, using a discrete time grid $\{t_k = k\Delta t\}_{k=0,\dots,T/\Delta t}$, where $\Delta t > 0$ is the time step, and, space-wise, using $\mathbf{P}^1$ Lagrange finite elements built upon an uniform subdivision of $[0;1]$: $\{x_i = i/\mathcal{N}\}$, for $i = 0, \dots, \mathcal{N}$. Our full output is:

$$f(\mathbf{x}) = f(u(\mathbf{x})) = \frac{1}{\mathcal{N}} \sum_{i=0}^{\mathcal{N}} u(t = T, x = x_i)$$

The reduced basis method is then applied to yield a surrogate solution $\widetilde{u}$ of (23), as well as an error bound ε_u on u. The reader can refer to [13] for full details on discretization, reduction and derivation of the RB error bound for this model. The main idea of the reduced basis method is to project u onto a well-chosen subspace of X whose dimension n (the reduced basis size) is smaller than the dimension of the finite element space that can be used to solve (23) numerically.
In our numerical experiments, we take $\mathcal{N} = 60$, $\Delta t = .01$, $T = .05$, $n_S(\psi) = n_T(\psi) = n(b_0) = n(b_1) = 0$, $n(u_0) = 1$, $\omega_1^{u_0} = 0.5$, $A_1^{u_0} = 5$ and $\psi_m = 1$.
The two input parameters are independent and uniformly sampled. The table below contains the ranges for them, and also the "true" values of the sensitivity indices, which have been calculated (in more than 14h CPU time) using a Monte-Carlo simulation with large sample size $N = 4 \times 10^6$ (so as to BC bootstrap confidence intervals of length $< 10^{-2}$) on the full model:

Parameter	Range	Confidence interval for sensitivity index
ν	[1 ; 20]	[0.0815;0.0832]
u_{0m}	[0 ; 0.3]	[0.9175;0.9182]

For benchmarking purposes, as they were computed using the true model with a large Monte-Carlo sample size, we can take the following values:

$$S_\nu = 0.082 \text{ and } S_{u_{0m}} = 0.918 \quad (25)$$

as "true" sensitivity indices.

4.1.2 Empirical coverage of combined confidence intervals

We test the performance of our combined confidence intervals by assessing their empirical *coverage*. This empirical coverage is measured by computing, for each input variable, 100 combined confidence intervals (each time with different input sample), and counting the proportion of intervals containing the true values in (25). The reduced basis size is $n = 9$; Monte-Carlo sample size is $N = 300$. Results are gathered in the table below:

Parameter	Mean confidence interval	Empirical coverage
ν	[0.0139;0.2083]	0.91
u_{0m}	[0.8421;0.9491]	0.87

4.1.3 Convergence benchmark

Figure 1 shows the lower $\widehat{S^m}$ and upper $\widehat{S^M}$ bounds (defined in Section 3.1) for different reduced basis sizes (hence different metamodel precisions) but fixed sample of size $N = 2000$, as well as the bootstrap confidence intervals computed using the procedure presented in Section 3.4. This figure exhibits the fast convergence of our bounds to the true value of the sensitivity index as the reduced basis size increases. We also see that the part of the error due to sampling remains constant, as sample size stays the same. We also notice that the true values lie between $\widehat{S}^m_{i,.025}$ and $\widehat{S}^M_{i,1-.025}$, and not always between $\widehat{S}^m_i$ and $\widehat{S}^M_i$; this is due to the Monte-Carlo part of the error, which is not taken into account by the $[\widehat{S}^m_i; \widehat{S}^M_i]$ bounds.

4.1.4 Choice of n and N

In practice, one wants to estimate sensitivity indices with a given precision (*ie.* to produce $(1-\alpha)$-level confidence intervals with prescribed length), and has no *a priori* indication on how to choose N and n to do so. Moreover, for one given precision, there may be multiple choices of suitable couples (N, n), balancing between sampling and metamodel error. Increasing N and/or n will increase the overall time for computation and improve the precision of the calculation (thanks to reduction in sampling error for increased N, or reduction in metamodel error for increased n).
We wish to choose the best compromise, that is, the one that gives the smallest computation time; for the reduced basis method, this time is roughly proportional to $N \times n^3$ (as a number proportional to N of linear systems of size n have to be solved). On the other hand, the length of the combined confidence interval can be modelled as the sum of two terms:

$$\frac{Z_\alpha}{\sqrt{N}} + \frac{C}{a^n} \tag{26}$$

where Z_α, C are positive constants and $a > 1$ is a constant. The first term accounts for sampling error; its form is heuristically deduced from central

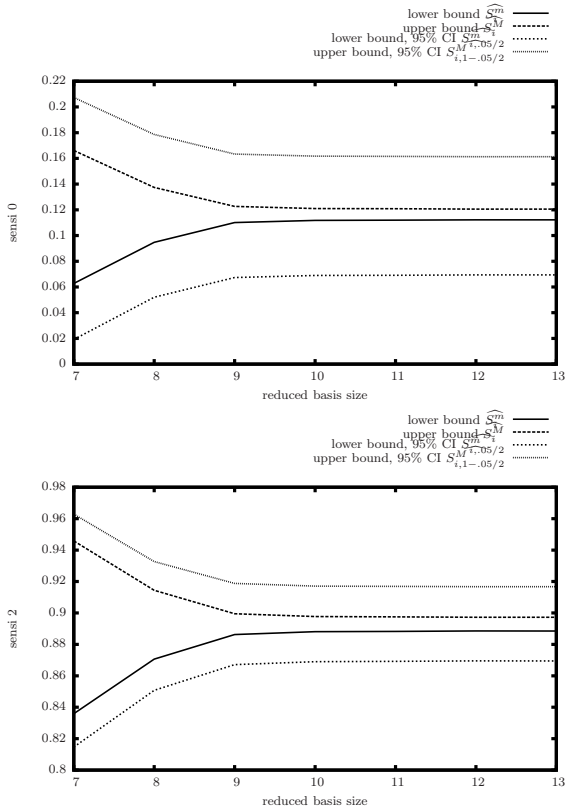

Figure 1: Convergence benchmark for sensitivity indices estimation in the Burgers' model, left: variable ν, right: variable u_{0m}. We plotted, for a fixed sample size $N = 2000$, estimator bounds $\widehat{S}_i^m$ and $\widehat{S}_i^M$ defined in (3.1), and endpoints $\widehat{S}_{i,.025}^m$ and $\widehat{S}_{i,1-.025}^M$ of the 95% confidence interval, for different reduced basis sizes.

N	5210	*8600*	23000
n	13	*11*	8
$N \times n^3$	11446370	*11446600*	11776000
Mean CI length	0.0416	*0.0389*	0.165

Table 1: Mean confidence interval lengths for different choices of n and N.

limit theorem. The second accounts for metamodel error; its exponential decay is backed up by numerical experiments as well as theoretical works (e.g. [3]). The parameters Z_α, C and a can be found by doing "calibration runs" with a fixed small sample size N and different reduced basis sizes $n_1, \ldots, n_K$, and by regressing the obtained confidence interval lengths using (26).
Minimizing the computation time under the constraint that error is equal to a given "target precision" leads to a straightforward constrained optimization problem (P), which can be solved, after estimation of Z_α, C and n, to find (after rounding to the closest integer) the optimal n and N for a desired precision.
The results of the optimization procedure, for two different target precisions, can be found in the table below. We also present the mean confidence interval length:

$$\frac{1}{p}\sum_{i=1}^{p}\left(S^M_{i,1-\alpha/2} - S^m_{i,\alpha/2}\right),$$

to check that the obtained confidence intervals (CI) have the desired length.

	Target precision= 0.04	Target precision= 0.02
N	8600	22000
n	11	11
CI for S_ν	[0.0691614 ; 0.119096]	[0.0659997 ; 0.0937285]
CI for $S_{u_{0m}}$	[0.890985 ; 0.919504]	[0.914266 ; 0.926452]
Mean CI length	0.0389	0.0199

To check for the optimality of the choices of n and N, we ran the estimation procedure for different n and N but (approximately) fixed time budget $N \times n^3$. Results are in Table 1. Only the choice in the middle column, in italics, is the result of the optimization procedure described in this paragraph. We show that it leads to better (smaller) confidence intervals than the other choices.

4.1.5 Full-scale example

We now present an example with more parameters. We take $\mathcal{N} = 300$, $\Delta t = .01$, $T = .05$, $n(\psi) = 0$, $n(u_0) = 5$, $n(b_0) = n(b_1) = 2$, $\psi_m = 1$,

$\omega_1^{u_0} = .5$, $\omega_2^{u_0} = 1$, $\omega_3^{u_0} = 1.5$, $\omega_4^{u_0} = 2$, $\omega_5^{u_0} = 2.5$, $\omega_1^{b_0} = \omega_1^{b_1} = .3$ and $\omega_2^{b_0} = \omega_2^{b_1} = .6$. Input parameters are assumed independent and uniformly distributed in the ranges below:

Parameter(s)	Range
ν	[1 ; 20]
u_{0m},$A_i^{u_0}$ $(i = 1, \ldots, 5)$	[0; 0.2]
$A_1^{b_0}$	[1;1.3]
$A_2^{b_0}$	[0;0.2]
$A_i^{b_1}$ $(i = 1, 2)$	[0;0.2]
f_m	[0;0.2]

We used a reduced basis of size $n = 13$. The combined confidence intervals obtained, for $N = 7000$, have mean length ≈ 0.08. Their computation required $N \times (p + 1) = 91000$ calls to the metamodel; these calls took 85 s of CPU time. By extrapolating the time necessary to generate the 200 (full) model evaluations for the choice of the reduced basis, we found that creating bootstrap confidence intervals of length 0.08 using the full model would require approximately 514 s. The use of a metamodel hence enabled a 6x speedup.

4.2 Application to a RKHS interpolation metamodel

We test our method using a RKHS interpolation metamodel based on the Ishigami function:

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

for $(X_j)_{j=1,2,3}$ i.i.d. uniform in $[-\pi; \pi]$.
The analytical values of sensitivities are known:

$$S_1 = 0.3139, \quad S_2 = 0.4424, \quad S_3 = 0.$$

We refer to A for details on the RKHS methodology and on the method of error estimation.
The experimental design $\mathcal{D}$, of size denoted by n, have been generated using maximin latin hypercube samples (using `maximinLHS` of R `lhs` package ([4]). The R package `mlegp` ([5]) is used as our RKHS interpolation/Kriging toolbox.
Smoothing parameters are set to: $\lambda = .6$, $h = .2$. We computed, for different learning sample sizes n, the empirical coverages and mean lengths of the combined confidence intervals (with the "smoothed" alternative described in Section 3.2). Monte-Carlo sample size is $N = 1000$, and $B = 300$ bootstrap replicates are computed. Finally, as Sobol indices are always bounded by 0 and 1, the reported confidence interval is the intersection of the computed interval with $[0; 1]$ (note that one can instead shift the interval to be a subset of $[0; 1]$ while maintaining its original length as done in [23]). Results are given in Table 2.

Learning sample size n	Variable (true index)	Mean 95% combined conf. int.	Mean length	Coverage
110	X_1 (0.3139)	[0.207;0.759]	0.552	0.94
110	X_2 (0.4424)	[0.0169;0.561]	0.545	0.98
110	X_3 (0)	[0;0.356]	0.357	1
130	X_1 (0.3139)	[0.262;0.626]	0.364	0.83
130	X_2 (0.4424)	[0.083;0.491]	0.408	0.93
130	X_3 (0)	[0;0.256]	0.257	1
160	X_1 (0.3139)	[0.274;0.509]	0.235	0.92
160	X_2 (0.4424)	[0.216;0.486]	0.27	0.92
160	X_3 (0)	[0;0.180]	0.18	1

Table 2: Obtained confidence intervals for the RKHS interpolation metamodel.

Variable (true index)	Mean 95% Combined conf. int.	Mean length	Coverage
X_1 (0.3139)	[0.0164222;0.218084]	0.202	0.02
X_2 (0.4424)	[0.0877679;0.351651]	0.264	0.11
X_3 (0)	[0.00750793;0.173349]	0.166	0.82

Table 3: Confidence intervals for the RKHS interpolation metamodel obtained with `CompModSA`.

The bound described in Section 3.1 has been tested on this example but did not give interesting results (ie. the produced intervals satisfied $[\widehat{S}_i^m; \widehat{S}_i^M][0;1]$). The reason is that the RKHS error estimator remains large, even for the larger n values for which the metamodel construction is practicable.
We compared these results with the ones obtained with `CompModSA`, a software package implementing the methodology described in [23]. We used as parameters: surface='mlegp' (Kriging metamodel), `n.mc.T=0` (we do not want any total index computation), `n.mc.S=1000` (sample size), `n.samples=1` (one run), and `n.CI=300` (generate confidence intervals using 300 bootstrap replications). We contributed a patch for `CompModSA`, available at `http://ljk.imag.fr/membres/Alexandre.Janon/compmodsa.php`, which adds to `sensitivity` the option `CI.S`, set to `TRUE` to compute bootstrap confidence intervals for the main effect index. The results, for a learning sample of size $n = 160$, are shown in Table 3.
This comparison clearly shows that our method is able to account for metamodel error, so as to keep the actual coverage of the produced confidence interval close to the expected one. We have been unable to fully compare our approach with the one of [14], as the lengths of the confidence interval obtained with this approach were not available; however we can state that, in this example, our method produces intervals of correct coverages for every variable and every training set size we have tested.

Conclusion and perspectives

We presented a methodology to quantify the impact of the sampling error and the metamodel error on the sensitivity indices computation, when the metamodel provides a pointwise error bound (or, at least, an error indicator) on the output of interest. Sampling error is handled by a classic bootstrap procedure, while metamodel error is managed using bounds on the sensitivity index estimator. Quantification of those two types of errors permits a certification on the performed estimation We applied our method on two types of metamodels: intrusive (reduced basis) and non-intrusive (RKHS interpolation). On the applications we see that our approach performs well both for reduced-basis and RKHS interpolation. Our method can also be applied with other metamodels, as soon as an assessment for pointwise metamodel error exists.

A RKHS interpolation and error bound

We now briefly recall the RKHS (reproducing kernel Hilbert space) interpolation method, and refer to [20] for the details. RKHS interpolation is known to be (see [21]) equivalent to interpolation by Kriging, also known as Gaussian process metamodelling. Using n evaluations of f, one can build a *training sample* consisting of n input-output pairs $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is the *experimental design* and $y_i = f(\mathbf{x}_i)$ for $i = 1, \ldots, n$. Let $R(\cdot, \cdot)$ be a positive definite kernel; the RKHS interpolator of $\mathcal{D}$ with respect to R is:

$$\widetilde{f}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \Sigma_S^{-1} \mathbf{y}_S$$

where:

$$\Sigma_S = \begin{pmatrix} R(\mathbf{x}_1, \mathbf{x}_1) & R(\mathbf{x}_1, \mathbf{x}_2) & \ldots & R(\mathbf{x}_1, \mathbf{x}_n) \\ R(\mathbf{x}_2, \mathbf{x}_1) & R(\mathbf{x}_2, \mathbf{x}_2) & \ldots & R(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ R(\mathbf{x}_n, \mathbf{x}_1) & R(\mathbf{x}_n, \mathbf{x}_2) & \ldots & R(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

$$\mathbf{k}(\mathbf{x}) = \left(R(\mathbf{x}_1, \mathbf{x}), \ldots, R(\mathbf{x}_n, \mathbf{x})\right)^T, \qquad \mathbf{y}_S = \left(y_1, \ldots, y_n\right)^T$$

In practice, R is assumed to belong to a parametrized family; for instance, R is a gaussian kernel:

$$R(\mathbf{x}, \mathbf{y}) = \exp\left(-\sum_{j=1}^{p} \theta_j \left(x_j - y_j\right)^2\right) \tag{27}$$

whose parameters $\theta_1, \ldots, \theta_p$ are estimated from $\mathcal{D}$ by minimizing some contrast function.

If f belongs to the RKHS associated with the R kernel, there exist a constant C, depending only on f and R, so that:

$$\left|f(\mathbf{x}) - \widetilde{f}(\mathbf{x})\right| \leq C\sqrt{\sigma_Z^2(\mathbf{x})} \tag{28}$$

where:

$$\sigma_Z^2(\mathbf{x}) = R(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^T \Sigma_S^{-1} \mathbf{k}(\mathbf{x})$$

We propose to estimate the constant C, which is, up to a multiplicative constant, the norm of f in the RKHS associated with R, by:

$$\widehat{C} = \max_{i=1,\ldots,n^\tau} \frac{\left|f(\mathbf{x}_i^\tau) - \widetilde{f}(\mathbf{x}_i^\tau)\right|}{\sqrt{\sigma_Z^2(\mathbf{x}_i^\tau)}}$$

where $\{\mathbf{x}_i^\tau\}_{i=1,\ldots,n^\tau}$ is a "test sample" that does not contain any point of the experimental design (so as to ensure that $\sigma_Z^2(\mathbf{x}_i^\tau) \neq 0$ for all $i = 1,\ldots,n^\tau$). For a continuous f, one easily sees that $\widehat{C}$ converges, as $\{\mathbf{x}_i^\tau\}$ fills the parameter set $\mathcal{P}$, to the smallest C so that (28) holds.

B Nomenclature

$\mathbf{R}$	set of all real numbers
$\overline{z}$	for any $\mathbf{z} \in \mathbf{R}^N$, $\overline{z} = \frac{1}{N}\sum_{k=1}^N z_k$
$\mathbf{X} = (X_1,\ldots,X_p)$, $\mathbf{X}^k$, $\mathbf{X}^{k\prime}$ $(k = 1,\ldots,N)$	i.i.d. input parameter samples
f	model output
$\widetilde{f}$	metamodel output
ε	metamodel error bound
N	Monte-Carlo sample size
y_k, y_k' $(k = 1,\ldots,N)$	output samples (defined at (2) and (3))
$\widetilde{y}_k, \widetilde{y}_k'$ $(k = 1,\ldots,N)$	metamodel output samples (defined at (5))
$\varepsilon_k, \varepsilon_k'$ $(k = 1,\ldots,N)$	metamodel error bound samples (defined at (6))
S_i $(i = 1,\ldots,p)$	true Sobol index with respect to X_i (defined at (1))
$\widehat{S}_i$	Monte-Carlo estimator of S_i (defined at (4))
$\widehat{S_i^m}$, $\widehat{S_i^M}$	Bounds for $\widehat{S}_i$
$\widehat{S}_{i,\alpha/2}^m$, $\widehat{S}_{i,1-\alpha/2}^M$	Bounds for $(1-\alpha)$-level confidence interval for S_i

Acknowledgements: We wish to thank Jean-Claude Fort for a suggestion which we exploited to perform our computation of the least square metamodel-induced error bound. We also thank Anestis Antoniadis and Ingrid Van Keilegom for advice on bootstrap methodology, and Robert Miller for language remarks. We finally thank anonymous referees for helpful comments about the

overall presentation of the paper. – This work has been partially supported by the French National Research Agency (ANR) through COSINUS program (project COSTA-BRAVA n°ANR-09-COSI-015).

References

[1] G. Archer, A. Saltelli, and I. Sobol. Sensitivity measures, ANOVA-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, 1997.

[2] S. Boyaval, C. Le Bris, Y. Maday, N. Nguyen, and A. Patera. A reduced basis approach for variational problems with stochastic parameters: Application to heat conduction with variable robin coefficient. *Computer Methods in Applied Mechanics and Engineering*, 198(41-44):3187–3206, 2009.

[3] A. Buffa, Y. Maday, A. Patera, C. Prud'homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis. *Mathematical Modelling and Numerical Analysis*, 2009.

[4] R. Carnell. *lhs: Latin Hypercube Samples*, 2009. R package version 0.5.

[5] G. M. Dancik. *mlegp: Maximum Likelihood Estimates of Gaussian Processes*, 2011. R package version 3.1.2.

[6] B. Efron. Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics*, 9(2):139–158, 1981.

[7] B. Efron and R. Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, 1(1):54–75, 1986.

[8] B. Efron and R. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, 1993.

[9] M. Grepl, Y. Maday, N. Nguyen, and A. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.

[10] M. Grepl and A. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[11] J. Helton, J. Johnson, C. Sallaberry, and C. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10-11):1175–1209, 2006.

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

[13] A. Janon, M. Nodet, and C. Prieur. Certified reduced-basis solutions of viscous Burgers equations parametrized by initial and boundary values. Preprint available at http://hal.inria.fr/inria-00524727/en, 2010, *submitted*.

[14] A. Marrel, B. Iooss, B. Laurent, and O. Roustant. Calculations of sobol indices for the gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751, 2009.

[15] N. Nguyen, K. Veroy, and A. Patera. Certified real-time solution of parametrized partial differential equations. *Handbook of Materials Modeling*, pages 1523–1558, 2005.

[16] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Math. Industry*, 2011.

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

[18] A. Saltelli, K. Chan, and E. Scott. *Sensitivity analysis*. Wiley: New York, 2000.

[19] T. J. Santner, B. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.

[20] R. Schaback. Mathematical results concerning kernel techniques. In *Prep. 13th IFAC Symposium on System Identification, Rotterdam*, pages 1814–1819. Citeseer, 2003.

[21] M. Scheuerer, R. Schaback, and M. Schlather. Interpolation of spatial data – a stochastic or a deterministic problem ? *Preprint, Universität Göttingen.* http://num.math.uni-goettingen.de/schaback/research/papers/IoSD.pdf, 2011.

[22] I. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[23] C. Storlie, L. Swiler, J. Helton, and C. Sallaberry. Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliability Engineering & System Safety*, 94(11):1735–1763, 2009.

[24] K. Urban and A. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *Comptes Rendus Mathematique*, 2012.

[25] K. Veroy and A. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.

[26] C. Zhu, R. Bryd, and J. Nocedal. *L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization*, 1997.