Confidence intervals for sensitivity indices using reduced-basis metamodels

A. Janon*, M. Nodet, C. Prieur

Joseph Fourier University, Laboratoire Jean Kuntzmann, MOISE team, BP 53, 38041 Grenoble Cedex, France

Abstract

Global sensitivity analysis is often impracticable for complex and time demanding numerical models, as it requires a large number of runs. The reduced-basis approach provides a way to replace the original model by a much faster to run code. In this paper, we are interested in the information loss induced by the approximation on the estimation of sensitivity indices. We present a method to provide a robust error assessment, hence enabling significant time savings without sacrifice on precision and rigourousness. We illustrate our method with an experiment where computation time is divided by a factor of nearly 6. We also give directions on tuning some of the parameters used in our estimation algorithms.

Keywords: sensitivity analysis, reduced basis method, Sobol indices, bootstrap method, Monte Carlo method.

Introduction

Many mathematical models use a large number of poorly-known parameters as inputs. When such models are encountered, it is important for the practitioner to quantify whether this uncertainty on the inputs has a large repercussion on the model output. This problem can be tackled by turning the uncertain input parameters into random variables, whose probability distribution reflects the practitioner's belief about the oddness of the fact that an input parameter takes some value. In turn, model output, as function of the model inputs, is a random variable; its probability distribution, uniquely determined by the inputs' distribution and the model itself, 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 (throughout expectation and standard deviation), most probable values (modes), *etc.*

Sensitivity analysis aims to identify the sensitive parameters, that is the parameters for which a small variation implies a large variation of the model output. In *stochastic sensitivity analysis*, one makes use of the output's probability distribution to define (amongst other measures of sensitivity) *sensitivity indices* (also known as *Sobol indices*). Sensitivity index of an output with respect to an input variable is the fraction of the variance of the output 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 part of variance caused by the variation of groups of two or more inputs, although main effects and total effects are generally

*Corresponding author

Email addresses: alexandre.janon@imag.fr (A. Janon), maelle.nodet@inria.fr (M. Nodet), clementine.prieur@imag.fr (C. Prieur)

sufficient to produce a satisfying sensitivity analysis. The reader is referred to Helton et al. (2006); Saltelli et al. 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 parameters' values (from the selected inputs' 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. This sample of outputs is fed into a suitable statistical estimator of the desired sensitivity index to produce a numerical estimate. The Monte Carlo approach to computation of Sobol indices is described in Sobol (2001), together with improvements in Homma and Saltelli (1996); Saltelli (2002).

A major drawback of the Monte Carlo estimation is that a large number of outputs of the model have to be evaluated for the resulting approximation of the sensitivity index to be accurate enough. In complex models, where a simulation for one single value of the parameters may take several minutes, the use of these methods "as-is" is impracticable. In those cases, one generally makes use of a *surrogate model* (also known as *reduced model*, *metamodel* or *response surface*). The surrogate model has to approximate well the original model (called the *full* model), while being many times faster to evaluate. The sensitivity index is then calculated *via* a sample of outputs generated by a call to the surrogate model, thus accelerating the overall computation time. The *reduced-basis* (RB) method Nguyen et al. (2005); Grepl and Patera (2005); Veroy and Patera (2005); Grepl et al. (2007) is a way of defining surrogate models when the original model is a discretization of a partial differential equation (PDE) depending on the input parameters. It comes with an *error bound*, that is, an upper bound on the error between the original output and the surrogate output.

The sensitivity index produced by Monte Carlo estimation on a surrogate model is tainted with 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*.

In order to make a rigorous sensitivity analysis, it is important to assess the magnitude of these two combined errors on the estimated sensitivity indices. Such assessment can also be used to help in the choice of correct approximation parameters (Monte-Carlo sample size and metamodel fidelity) to achieve a desired precision in estimated indices.

Sampling error can be classically estimated for a moderate cost by using *bootstrap resampling* Efron et al. (1993); Archer et al. (1997). Based on statistical estimation theory, the bootstrap technique involves the generation of a sample of replications of sensitivity index estimator, whose empirical distribution serves as approximation of the true (unknown) estimator distribution, in order to produce asymptotic confidence intervals which give good results in many practical cases. A variation on the bootstrap, which addresses sampling error as well as metamodel error, has been proposed in Storlie et al. (2009); also Marrel et al. (2009) develops a methodology in Kriging metamodels. In this paper, we present another confidence interval-based approach for assessing sampling errors, together with errors caused by reduced-basis metamodels, which makes use of the certified, *a posteriori* error bound that comes with the reduced-basis method. Boyaval et al. (2009) also makes use of the reduced-basis output error bound to certify computation of the expectation and the variance of a model output with neglected sampling error.

The advantages of 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 clear separation between estimation (sampling) and metamodel error, and moderate computational requirements (time should rather be spent at making a precise computation than at measuring precision). In other words, our method allows to estimate sensitivity indices by using a reduced basis metamodel which largely speeds up computation times, while rigorously keeping track of the precision of the estimation.
This paper is organized as follows: in the first part, we go through the prerequisites for our approach: we give the definition and standard Monte Carlo estimator of the sensitivity indices we are interested in, and give an overview of the reduced basis method; in the second and third parts, we present our confidence interval estimation technique for the sensitivity index, which accounts for the two sources of error described earlier. In the fourth part, we present the numerical results we obtain on an example of a reduced-basis metamodel.

1. Model output and sensitivity analysis methodology

1.1. Sensitivity indices

1.1.1. General setting

In order to define sensitivity indices, we choose a probability distribution for the input variables, turning each input variable X_i $(i = 1, \dots, p)$ into a random variable with known distribution; the model output $Y = f(X_1, \dots, X_p)$ (assumed to be a scalar; multiple outputs can be treated separately) is thus for $\mathbf{X} = (X_1, \dots, X_p)$ a $\sigma(\mathbf{X})$-measurable random variable. 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.

1.1.2. Monte-Carlo estimator

We are interested in the following Monte-Carlo estimator for S_i Homma and Saltelli (1996); Saltelli (2002): a sample size $N \in \mathbf{N}$ being given, let $\left\{\mathbf{X}^k\right\}_{k=1,\dots,N}$ and $\left\{\mathbf{X}'^k\right\}_{k=1,\dots,N}$ be two random i.i.d. samples of size N each, drawn from the distribution of the input vector $\mathbf{X}$.
For $k = 1, \dots, N$, we note:

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

and:

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

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

$$\widehat{S_i} = \frac{\frac{1}{N}\sum_{k=1}^N y_k y'_k - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)\left(\frac{1}{N}\sum_{k=1}^N y'_k\right)}{\frac{1}{N}\sum_{k=1}^N y_k^2 - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)^2} \tag{2}$$

It can be shown that $\widehat{S_i}$ is a strongly consistent estimator of S_i.

1.2. Metamodel construction: overview of the reduced basis method

We briefly present here the reduced basis method for affinely parametrized elliptic partial differential equations. For more details, see Nguyen et al. (2005).

1.2.1. Offline-online decomposition

The reduced basis method deals with variational problems of the form: given an input parameter vector $\mathbf{X}$ belonging to a parameter set $\mathcal{D} \subset \mathbf{R}^p$,

$$\text{find } u(\mathbf{X}) \in \mathcal{F} \text{ so that} \quad a(u, v; \mathbf{X}) = \psi(v) \ \ \forall v \in \mathcal{F} \tag{3}$$

where $\mathcal{F}$ is an appropriate finite dimensional function space (usually a discretization of a continuous function space such as H^1 or H^1_0), ψ is a linear form on $\mathcal{F}$, and $a(\cdot, \cdot; \mathbf{X})$ is a parameter-dependent bilinear form on $\mathcal{F}$ that can be written under *affine form*:

$$a(v, w; \mathbf{X}) = \sum_{q=1}^{Q} \Theta_q(\mathbf{X}) a_q(v, w) \quad \forall v, w \in \mathcal{F}$$

where Θ_q are arbitrary real functions defined on $\mathcal{D}$, and a_q are parameter-independent bilinear functions on $\mathcal{F}$.

Traditional computation of $u(\mathbf{X})$ for a prescribed $\mathbf{X}$ involves looking for $u(\mathbf{X})$ as a linear combination:

$$u(\mathbf{X}) = \sum_{i=1}^{\dim \mathcal{F}} u_i(\mathbf{X}) \phi_i$$

where $(\phi_i)_{i=1,\dots,\dim \mathcal{F}}$ is a suitable basis of $\mathcal{F}$, and the unknowns are $(u_i(\mathbf{X}))_{i=1,\dots,\dim \mathcal{F}}$. This way, one obtains the following linear system of $(\dim \mathcal{F})$ equations:

$$\sum_{i=1}^{\dim \mathcal{F}} \left(\sum_{q=1}^{Q} \Theta_q(\mathbf{X}) a_q(\phi_i, \phi_j) \right) u_i(\mathbf{X}) = \psi(\phi_j) \quad j = 1, \dots, \dim \mathcal{F} \tag{4}$$

In many cases, the space $\mathcal{F}$ has a large dimension, so as to represent many functions of the continuous function space with a great precision, and the system (4) is very large (although one can generally choose $\mathcal{F}$ and (ϕ_i) so as to produce a very sparse system).

When (3) has to be solved for many values of $\mathbf{X}$ (the so-called *many query* context), the reduced basis method can be used to speed up the overall computation, which is split into two parts. In the *offline* phase, we choose a linearly independent family $\mathcal{B} = \{\zeta_1, \dots, \zeta_n\}$ of n vectors in $\mathcal{F}$, and compute and store the Q n-by-n matrices of each a_q form (*ie.* the matrices A_q whose (i, j) coefficient is $a_q(\zeta_i, \zeta_j)$) and the n-vector representing ψ (*ie.* the vector whose ith coefficient is $\psi(\zeta_i)$) in the basis $\mathcal{B}$ (called the *reduced basis*). The offline phase does not depend on a particular value of $\mathbf{X}$ and can be done once. Then, for each value of $\mathbf{X}$ for which $u(\mathbf{X})$ has to be computed, one can proceed to the *online* phase: using information stored during the offline phase, the following n-by-n linear system is assembled and solved for $(\widetilde{u}_i(\mathbf{X}))_{i=1,\dots,n}$:

$$\sum_{i=1}^{n} \left(\sum_{q=1}^{Q} \Theta_q(\mathbf{X}) a_q(\zeta_i, \zeta_j) \right) \widetilde{u}_i(\mathbf{X}) = \psi(\zeta_j) \quad j = 1, \dots, n \tag{5}$$

Then $\widetilde{u}(\mathbf{X}) \approx u(\mathbf{X})$ is recovered by using $\widetilde{u}(\mathbf{X}) = \sum_{i=1}^{n} \widetilde{u}_i(\mathbf{X}) \zeta_i$. In many cases, $\{u(\mathbf{X}); \mathbf{X} \in \mathcal{D}\}$ lies in a manifold of dimension much smaller than $\dim \mathcal{F}$, and it is possible to choose a linear space of approximation of dimension $n \ll \dim \mathcal{F}$ and thus, solve (5) much faster than (4) while keeping $\widetilde{u}$ sufficiently close to u. At the end of this section we will see a method to automatically choose an "effective" reduced basis, that allows accurate representation of $u(\mathbf{X})$ for $\mathbf{X} \in \mathcal{D}$.

When the model output $f(\mathbf{X})$ is a linear functional $f(u(\mathbf{X}))$ of $u(\mathbf{X})$, the surrogate output can be defined as:

$$\widetilde{f}(\mathbf{X}) := f(\widetilde{u}(\mathbf{X})) = \sum_{q=1}^{Q} \widetilde{u}_i(\mathbf{X}) f(\zeta_i) \tag{6}$$

whose parameter-independent reals $\ell(\zeta_i)$ $(i = 1, \ldots, n)$ can be calculated and stored during the offline phase, allowing evaluation of $\widetilde{f}(\mathbf{X})$ without explicitly forming $\widetilde{u}(\mathbf{X})$, and leading to a meta-model whose complexity of evaluation depends only on its dimension n (and on Q) – and no more on the dimension of the original model $\dim \mathcal{F}$.

1.2.2. Error bound

An interesting feature of the reduced basis approach is that it comes with a provable error bound $\varepsilon_u(\mathbf{X})$ fully computable with a complexity independent of $\dim \mathcal{F}$ Nguyen et al. (2005). This error bound satisfies

$$\|u(\mathbf{X}) - \widetilde{u}(\mathbf{X})\| \leq \varepsilon_u(\mathbf{X}) \quad \forall \mathbf{X} \in \mathcal{D}$$

for a chosen Hilbert space norm $\|\cdot\|$ on $\mathcal{F}$. To present the error bound, we assume, for simplicity, that the a_q's are symmetric bilinear forms, and that $a(\cdot, \cdot; \mathbf{X})$ is *uniformly coercive*, that is, $\alpha(\mathbf{X})$ defined by:

$$\alpha(\mathbf{X}) = \sup_{v \in \mathcal{F}, \|v\|=1} a(v, v; \mathbf{X})$$

satisfies $\alpha(\mathbf{X}) > 0$ for all $\mathbf{X} \in \mathcal{D}$.

We claim that:

$$\|u(\mathbf{X}) - \widetilde{u}(\mathbf{X})\| \leq \frac{\|r(\mathbf{X})\|_{\mathcal{F}'}}{\alpha(\mathbf{X})} \quad \forall \mathbf{X} \in \mathcal{D}$$

where $r(\mathbf{X})$ is the residual linear form, defined by:

$$r(\mathbf{X})(v) = \psi(v) - a(\widetilde{u}(\mathbf{X}), v; \mathbf{X})$$

and $\|\cdot\|_{\mathcal{F}'}$ is the dual norm on $\mathcal{F}$:

$$\|\ell\|_{\mathcal{F}'} = \sup_{v \in \mathcal{F}, \|v\|=1} \ell(v)$$

Efficient procedures have been developed for efficient offline-online computation of $\|r(\mathbf{X})\|_{\mathcal{F}'}$, and a *lower bound* $\widetilde{\alpha}(\mathbf{X}) < \alpha(\mathbf{X})$, leading to a computable error bound on u:

$$\varepsilon_u(\mathbf{X}) = \frac{\|r(\mathbf{X})\|_{\mathcal{F}'}}{\widetilde{\alpha}(\mathbf{X})}$$

This error bound on u can be used to develop an error bound on the output. For example, if $f(\mathbf{X}) = f(u(\mathbf{X}))$ and $\widetilde{f}(\mathbf{X})$ is the surrogate output defined in (6), one can use

$$\varepsilon(\mathbf{X}) = \|f\|_{\mathcal{F}'} \, \varepsilon_u(\mathbf{X}) \tag{7}$$

which satisfies:

$$\left| f(\mathbf{X}) - \widetilde{f}(\mathbf{X}) \right| \leq \varepsilon(\mathbf{X}) \quad \forall \mathbf{X} \in \mathcal{D} \tag{8}$$

as error bound on the output.

1.2.3. POD-based procedure for reduced basis choice

We now describe a way of selecting a reduced basis $\{\zeta_1, \ldots, \zeta_n\}$.

We randomly choose a finite subset $\Xi = \{\mathbf{X}^1, \ldots, \mathbf{X}^m\} \subset \mathcal{D}$, and compute $u(\mathbf{X})$ for each $\mathbf{X} \in \Xi$. We put the coordinates of $u(\mathbf{X})$ with respect to the basis $\{\phi_1, \ldots, \phi_{\dim \mathcal{F}}\}$ of $\mathcal{F}$ as columns of a *snapshot matrix* S:

$$S = \begin{pmatrix} u(\mathbf{X}^1)_1 & u(\mathbf{X}^2)_1 & \ldots & u(\mathbf{X}^m)_1 \\ u(\mathbf{X}^1)_2 & u(\mathbf{X}^2)_2 & \ldots & u(\mathbf{X}^m)_2 \\ \vdots & \vdots & \ldots & \vdots \\ u(\mathbf{X}^1)_{\dim \mathcal{F}} & u(\mathbf{X}^2)_{\dim \mathcal{F}} & \ldots & u(\mathbf{X}^m)_{\dim \mathcal{F}} \end{pmatrix} \tag{9}$$

We now proceed with the *Proper Orthogonal Decomposition (POD)* of the S matrix: we compute $\{z_1, \ldots, z_n\}$, where z_i is an eigenvector associated with the i^{th} largest eigenvalue of the m-by-m symmetric matrix $S^T \Omega S$ (where Ω is the matrix of the scalar product $<,>$ associated with $\|\cdot\|$, with respect to the $\{\phi_1, \ldots, \phi_{\dim \mathcal{F}}\}$ basis), and define the vectors of the reduced basis to be:

$$\zeta_i = \frac{S z_i}{\|S z_i\|}$$

One can show that the $\{\zeta_1, \ldots, \zeta_n\}$ are solutions of the following optimization program:

$$\text{Minimize } \sum_{\mathbf{X} \in \Xi} \|u(\mathbf{X}) - \pi[u(\mathbf{X})]\|^2, \text{ under the constraints } <\zeta_i, \zeta_j> = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ else.} \end{cases}$$

where π is the orthogonal projector onto $\text{Span}(\zeta_1, \ldots, \zeta_n)$.
Proper orthogonal decomposition (also known as Principal component analysis (PCA), or Singular value decomposition (SVD)) Chatterjee (2000), and variants of POD, are widely used in model reduction without error bounds Bui-Thanh et al. (2007); Bergmann and Iollo (2008). We also showed in Janon et al. (2010, *submitted.*) that POD reduced bases are efficient with respect to the obtained error bounds.

1.3. Estimator on reduced model

Using the reduced model to perform the sensitivity analysis is straightforward: replace every call to the full model f by a call to the reduced one $\widetilde{f}$. This gives rise to an estimator $\widehat{\widetilde{S}_i}$ converging, for $N \to +\infty$, to the true value of the sensitivity index on $\widetilde{f}$:

$$\widetilde{S}_i = \frac{\text{Var}\mathbf{E}(\widetilde{Y}|X_i)}{\text{Var}\widetilde{Y}}$$

Sampling error of this estimator can be assessed using bootstrap Efron et al. (1993), see Algorithm 1.
However, doing so does not take in account the gap between S_i and $\widetilde{S}_i$. If the metamodel is "too far" from the original model, the $(1-\alpha)$–confidence interval estimated using it will not contain the true value of S_i with probability close to $1-\alpha$. On the other hand, a moderate-fidelity metamodel might be well-suited to give a "rough" estimate of sensitivity indices – in some cases such a rough estimate is sufficient – but the user would like be informed that the metamodel he uses gives him a limited-precision estimator; such a limited precision would reflect in the increase in the width of the output confidence interval.

2. 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 the first part 2.1, we review the bootstrap, which we will use for the treatment of sampling error. In the second part 2.2, we show how to use reduced-basis bounds to assess metamodel error.

2.1. 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 Archer et al. (1997).
We use the bias-corrected (BC) percentile method presented in Efron (1981); Efron and Tibshirani (1986). The principle of this method can be summed up the following way: let $\widehat{\theta}(X_1, \ldots, X_n)$ be an

estimator for an unknown parameter θ in a reference population $\mathcal{P}$. To get a point estimate of θ, one takes a random i.i.d. n-sample $\{x_1, \ldots, x_n\}$ from $\mathcal{P}$, and computes $\widehat{\theta}(x_1, \ldots, x_n)$. In (nonparametric) bootstrap we repeatedly, for $b = 1, \ldots, B$, sample $\{x_1[b], \ldots, x_n[b]\}$ with replacement from the original sample $\{x_1, \ldots, x_n\}$ and get a *replication* of $\widehat{\theta}$ by computing $\widehat{\theta}[b] = \widehat{\theta}(x_1[b], \ldots, x_n[b])$. This way we get a sample $\mathcal{R} = \{\widehat{\theta}[1], \ldots, \widehat{\theta}[B]\}$ of replications of $\widehat{\theta}$.
Now see 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 $\widehat{\theta} = \widehat{\theta}(x_1, \ldots, x_n)$, a "bias correction constant" z_0 can be estimated:

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

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 $\mathcal{R}$.
This confidence interval has been justified in Efron (1981) when there exists an increasing transformation g, $z_0 \in \mathbf{R}$ and $\sigma > 0$ such that $g(\widehat{\theta}) \sim \mathcal{N}(\theta - z_0\sigma, \sigma)$ and $g(\widehat{\theta}^*) \sim \mathcal{N}(\widehat{\theta} - z_0\sigma, \sigma)$, where $\widehat{\theta}^*$ is the bootstrapped $\widehat{\theta}$, for fixed sample $\{x_1, \ldots, x_n\}$ and (hence) fixed $\widehat{\theta} = \widehat{\theta}(x_1, \ldots, x_n)$. In practice, due to the complex analytic expressions (20) of the estimators we are going to bootstrap, it seems hard to prove that such a g exists. However, we give in Section 4.4 empirical evidence that, for the two estimators defined at (20), g can approximatively be chosen as identity.
The full computation method of a BC bootstrap confidence interval for S_i is given in Algorithm 1.
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.

2.2. Metamodel error

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

$$\widetilde{y}_k = \widetilde{f}(\mathbf{X}^k), \quad \widetilde{y}'_k = \widetilde{f}(X_1'^k, \ldots, X_{i-1}'^k, X_i^k, X_{i+1}'^k, \ldots, X_p'^k)$$

and:

$$\varepsilon_k = \varepsilon(\mathbf{X}^k), \quad \varepsilon'_k = \varepsilon(X_1'^k, \ldots, X_{i-1}'^k, X_i^k, X_{i+1}'^k, \ldots, X_p'^k)$$

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

$$\widehat{S}_i^m \leq \widehat{S}_i \leq \widehat{S}_i^M \tag{10}$$

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.
Let:

$$R(a; \mathbf{y}, \mu, \mu') = \sum_{k=1}^{N} \left(y'_k - (a(y_k - \mu) + \mu')\right)^2$$

where $\mathbf{y} = (y_1, \ldots, y_N, y'_1, \ldots, y'_N)$ and $\mu, \mu' \in \mathbf{R}$.
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:

$$\widehat{S_i} = \frac{\frac{1}{N}\sum_{k=1}^N y_k y'_k - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)\left(\frac{1}{N}\sum_{k=1}^N y'_k\right)}{\frac{1}{N}\sum_{k=1}^N y_k^2 - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)^2}$$

one easily shows that:

$$\widehat{S}_i = \operatorname*{argmin}_{a \in \mathbf{R}} R(a; \mathbf{y}, \overline{y}, \overline{y}')$$

where: $\overline{y} = \dfrac{1}{N}\sum_{k=1}^N y_k$ and $\overline{y}' = \dfrac{1}{N}\sum_{k=1}^N y'_k$.
In other words, $\widehat{S}_i$ is the slope of the linear least squares regression of the $\{y'_k\}_k$ on the $\{y_k\}$.
Define:

$$R_{inf}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') = \sum_{k=1}^N \left\{ \inf_{z \in [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k], z' \in [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k]} \left(z' - (a(z-\mu) + \mu')\right)^2 \right\} \tag{11}$$

and:

$$R_{sup}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') = \sum_{k=1}^N \left\{ \sup_{z \in [\widetilde{y}_k - \varepsilon_k; \widetilde{y}_k + \varepsilon_k], z' \in [\widetilde{y}'_k - \varepsilon'_k; \widetilde{y}'_k + \varepsilon'_k]} \left(z' - (a(z-\mu) + \mu')\right)^2 \right\} \tag{12}$$

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

$$R_{inf}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') \leq R(a; \mathbf{y}, \mu, \mu') \leq R_{sup}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') \quad \forall a, \mu, \mu' \in \mathbf{R} \tag{13}$$

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:

$$R(a; \mathbf{y}, \mu, \mu') = \alpha a^2 + \beta a + \gamma$$

$$R_{inf}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') = \alpha_{inf} a^2 + \beta_{inf} a + \gamma_{inf} \tag{14}$$

$$R_{sup}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu') = \alpha_{sup} a^2 + \beta_{sup} a + \gamma_{sup} \tag{15}$$

These coefficients depend on μ and μ' [1]. We do not explicitly write this dependence until the last part of our discussion.
Using (13) 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:

$$\gamma_{inf} - \gamma \leq 0 \tag{16}$$

$$(\beta_{inf} - \beta)^2 \leq 4(\alpha_{inf} - \alpha)(\gamma_{inf} - \gamma) \tag{17}$$

As $(\beta_{inf} - \beta)^2 \geq 0$, Equations (16) and (17) above imply that $\alpha_{inf} - \alpha \leq 0$, and that:

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

[1] as well on $\mathbf{y}$ (for α, β, γ) and $\widetilde{\mathbf{y}}$ and ε (for the other coefficients)

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 have $\alpha_{inf} \approx \alpha$ and, as $\alpha > 0$, the hypothesis $\alpha_{inf} > 0$ is realistic.
So, under this supplementary assumption, we have:

$$\underset{a}{\operatorname{argmin}}\, R(a; \mathbf{y}, \mu, \mu') = -\frac{\beta}{2\alpha} \geq -\frac{\beta_{inf} + \delta_{inf}}{2\alpha_{inf}}$$

Now using the second part of (13) and the same reasoning on the non-positive quadratic function of a: $R(a; \mathbf{y}, \mu, \mu') - R_{sup}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu')$, we get that: $\alpha \leq \alpha_{sup}$, and: $\beta_{sup} - \delta_{sup} \leq \beta \leq \beta_{sup} + \delta_{sup}$. Hence,

$$\underset{a}{\operatorname{argmin}}\, R(a; \mathbf{y}, \mu, \mu') \leq -\frac{\beta_{sup} - \delta_{sup}}{2\alpha_{sup}}$$

where $\delta_{sup} = 2\sqrt{(\alpha - \alpha_{sup})(\gamma - \gamma_{sup})}$. This comes without supplementary assumption, because $\alpha_{sup} \geq \alpha$ and $\alpha > 0$, as the minimum of $R(\cdot; \mathbf{y}, \mu, \mu')$ exists.
As we clearly have δ_{inf} and δ_{sup} less than (or equal to) $\widehat{\delta} := 2\sqrt{(\alpha_{inf} - \alpha_{sup})(\gamma_{inf} - \gamma_{sup})}$, we deduce that:

$$-\frac{\beta_{inf}(\mu, \mu') + \widehat{\delta}(\mu, \mu')}{2\alpha_{inf}(\mu, \mu')} \leq \underset{a}{\operatorname{argmin}}\, R(a; \mathbf{y}, \mu, \mu') \leq -\frac{\beta_{sup}(\mu, \mu') - \widehat{\delta}(\mu, \mu')}{2\alpha_{sup}(\mu, \mu')}$$

where we have explicited the dependencies in μ and μ'.
To finish, it is easy to see that we have:

$$\overline{\mathcal{P}} := [\overline{\widetilde{y}} - \overline{\varepsilon}; \overline{\widetilde{y}} + \overline{\varepsilon}] \ni \overline{y} \tag{18}$$

and:

$$\overline{\mathcal{P}}' := [\overline{\widetilde{y}'} - \overline{\varepsilon}'; \overline{\widetilde{y}'} + \overline{\varepsilon}'] \ni \overline{y}' \tag{19}$$

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

$$\min_{\mu \in \overline{\mathcal{P}}, \mu' \in \overline{\mathcal{P}}'} \left(-\frac{\beta_{inf}(\mu, \mu') + \widehat{\delta}(\mu, \mu')}{2\alpha_{inf}(\mu, \mu')} \right) \leq \widehat{S}_i = \underset{a}{\operatorname{argmin}}\, R(a; \mathbf{y}, \mu, \mu') \leq \max_{\mu \in \overline{\mathcal{P}}, \mu' \in \overline{\mathcal{P}}'} \left(-\frac{\beta_{sup}(\mu, \mu') - \widehat{\delta}(\mu, \mu')}{2\alpha_{sup}(\mu, \mu')} \right)$$

Hence, (10) is verified with:

$$\widehat{S}_i^m = \min_{\mu \in \overline{\mathcal{P}}, \mu' \in \overline{\mathcal{P}}'} \left(-\frac{\beta_{inf}(\mu, \mu') + \widehat{\delta}(\mu, \mu')}{2\alpha_{inf}(\mu, \mu')} \right), \quad \widehat{S}_i^M = \max_{\mu \in \overline{\mathcal{P}}, \mu' \in \overline{\mathcal{P}}'} \left(-\frac{\beta_{sup}(\mu, \mu') - \widehat{\delta}(\mu, \mu')}{2\alpha_{sup}(\mu, \mu')} \right) \tag{20}$$

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 $\Xi \subset \overline{\mathcal{P}} \times \overline{\mathcal{P}'}$. See Algorithm 2 for a summary of the entire computation procedure.

3. Combined confidence intervals and parameters choice

3.1. Combined confidence intervals

In the last section, we have seen how to separately assess sampling error and metamodel error. To take both error into account simultaneously, we propose using bootstrap confidence intervals (see

Section 2.1) 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 $\{1, \ldots, N\}$.
The BC bootstrap confidence interval procedure (see 2.1) can then be used to produce a $1-\alpha$-level confidence interval $[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^m_{i,1-\alpha/2}]$ for S_i^m, and a confidence interval $[\widehat{S}^M_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}]$ 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.

ε sampling
Optionally, we can introduce a postulated uncertainty on the error bounds through what we call *ε sampling*. In ε sampling, the b^{th} bootstrap replicates for $\widehat{S_i^m}$ and $\widehat{S_i^M}$ are computed using $(\varepsilon^*_k)_{k\in L_b}, \left(\varepsilon'^*_k\right)_{k\in L_b}$ as error bounds, where ε^*_k and ε'^*_k are sampled independently from a uniform distribution in $[\eta_k\varepsilon_k; \varepsilon_k]$ and $[\eta'_k\varepsilon'_k; \varepsilon'_k]$, where $\eta_k, \eta'_k \in [0; 1]$ are alleged *effectivities* of our error bound, that is, an indicator of the ratio between the true errors $|\widetilde{y}_k - y_k|$ and ε_k (and between $|\widetilde{y}'_k - y'_k|$ and ε'_k). Setting effectiveness close to zero narrows confidence intervals, putting more trust in the reduced model than in the error bound, which is considered too pessimistic; on the contrary, effectiveness close to one means that error bound does not overestimate true error too much and that the error can not be considered too smaller than it.
The procedure for obtaining confidence intervals is summed up in Algorithm 3.

3.2. Choice of reduced basis size and Monte-Carlo sample size

When doing a Monte Carlo estimation of sensitivity indices using a reduced basis metamodel, by means of confidence intervals computed with the strategy described above, one has to choose two important parameters : the sample size (N) and the number of elements in the reduced basis (n). Increasing N and/or n will increase the overall time for computation (because of a larger number of surrogate simulations to perform if N is increased, or, if n is increased, each surrogate simulation taking more time to complete due to a larger linear system to solve). However, increase in these parameters will also improve the precision of the calculation (thanks to reduction in sampling error for increased N, or reduction in metamodel error for increased 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. We wish to choose the best, that is, the one who gives the smallest computation time.
The aim of this section is to describe a simple computational model that helps us in making a good choice of sample size and reduced basis size to produce a confidence interval of a desired precision.

Formulation as a constrained optimization problem
On the one hand, we evaluate computation time: an analysis of the reduced basis method shows that the most costly operation made during an online evaluation (see Section 1.2) is the resolution of a linear system of n equations; this resolution can be done (e.g., by using Gauss' algorithm) with $O(n^3)$ operations. This has to be multiplied by the required number of online evaluations, i.e. the sample size N. Hence, we may assume that computation time is proportional to $N \times n^3$.
On the other hand, the mean length of the $(1-\alpha)$-level confidence intervals for $S_1, \ldots, S_p$ can be written as the sum of two terms. The first, depending on N, accounts for sampling error and can be modelled as

$$\frac{Z_\alpha}{\sqrt{N}}$$

for a constant $Z_\alpha > 0$. The assumption of $1/\sqrt{N}$ decay is heuristically deduced from central limit theorem.
The second term, which accounts for metamodel error, is assumed to be of exponential decay when n increases: C/a^n, where $C > 0$ and $a > 1$ are constants. This assumption is backed up by numerical experiments as well as theoretical works Buffa et al. (2009).
Once this analysis has been done, we translate our problem into the following constrained minimization program:

$$\text{Find } (N^*, n^*) = \underset{(n,N)\in\mathbf{R}^+\times\mathbf{R}^+}{\operatorname{argmin}} \; n^3 \times N \text{ so that } \frac{2q_\alpha\sigma}{\sqrt{N}} + \frac{C}{a^n} = P \tag{21}$$

where P is the desired average precision for the confidence intervals.
Note that we converted the discrete design variables N and n to continuous positive variables so as to use the standard tools of continuous optimization; once optimum of the continuous problem have been found, we just round it to the nearest integer couple to recover a near-optimal integer solution.

Resolution of the optimization problem
The constraint in (21) is equivalent to the conjunction of the following two equations:

$$N = \left(\frac{Z_\alpha}{P - \frac{C}{a^n}}\right)^2 \tag{22}$$

$$P - \frac{C}{a^n} \geq 0 \tag{23}$$

so that the function to minimize over $I =]n_c; +\infty[$ (where $n_c = \ln(C/P)/\ln(a)$ has been chosed so as to satisfy (23)) is:

$$\phi(n) = (Z_\alpha)^2 \frac{n^3}{\left(P - \frac{C}{a^n}\right)^2}$$

This function is differentiable on I and tends to $+\infty$ as $n \to n_c$ and $n \to +\infty$; hence it has a minimizer $n^* \in I$ that satisfies $\phi'(n^*) = 0$, which is equivalent to:

$$\frac{n^*}{Pa^{n^*} - C} - \frac{3}{2C\ln a} = 0 \tag{24}$$

On I, (24) is equivalent to:

$$n^* - \frac{3P}{2C\ln a}a^{n^*} = -\frac{3}{2C\ln a}$$

Now let ψ be the function defined on $\mathbf{R}$ by $\psi(x) = x - \frac{3P}{2C\ln a}a^x$. By remarking that $\psi'(x) = 1 - \frac{3P}{2C}a^x$ is continuous and nonzero on $\left]\ln\left(\frac{2C}{3P}\right)/\ln a; +\infty\right[\supset I$, one has that ψ is injective on I, and so (24) has at most one solution in I. Thus (24) has exactly one solution in I, of which an approximate value can be found by using bisection method Press et al. (1992) on $[n_c + \varepsilon; L]$ where $\varepsilon > 0$ is small enough and L is a sufficiently large. Once n^* has been found, we can find the optimal N^* by setting $n = n^*$ in (22).

Estimation of the parameters

The last question that remains to address is the estimation of the Z_α, a and C constants. The Z_α parameter is estimated by running the estimation procedure on the metamodel, for fixed N and n, estimating combined BC bootstrap confidence intervals (Section 2.1) and taking:

$$\widehat{Z}_\alpha = \sqrt{N}\left(\widehat{S}^M_{i,1-\alpha/2} - \widehat{S}^M_i + \widehat{S}^m_{i,\alpha/2} - \widehat{S}^m_i\right)$$

where the factor multiplying $\sqrt{N}$ is the estimated "Monte-Carlo part" of the error.

The a and C parameters are estimated by running an estimation procedure, for a single N fixed, and different reduced basis sizes $n_1, \ldots, n_K$, and measuring, for each reduced basis size, the average metamodel error $e(n_k) = \widehat{S}^M_i - \widehat{S}^m_i$, for $k = 1, \ldots, K$. The $\{(n_1, e(n_1)); \ldots; (n_K, e(n_K))\}$ pairs are then used to fit the exponential regression model $e(n) = C/a^n$.

If one wants to estimate the sensitivity indices with respect to all variables $i = 1, \ldots, p$ for a single value of N and n, one can use bootstrap procedure to estimate Monte-Carlo errors $\widehat{E}_1, \ldots, \widehat{E}_p$ for each of the p sensitivity indices estimators:

$$\widehat{E}_i = \widehat{S}^M_{i,1-\alpha/2} - \widehat{S}^M_i + \widehat{S}^m_{i,\alpha/2} - \widehat{S}^m_i$$

and then to take:

$$\widehat{Z}_\alpha = \frac{\sqrt{N}}{p}\sum_{i=1}^{p} \widehat{E}_i$$

4. Numerical results and discussion

In this section, we test our combined confidence interval procedure described earlier, and compare it with Monte-Carlo estimation on the full model (with bootstrap to assess sampling error), and with the procedure described in Storlie et al. (2009) and implemented in the *CompModSA* R package. Our criteria of comparison are the CPU time needed to compute the intervals and the lengths of these intervals (the smaller the better).

In all our tests we take $\alpha = .05$ and $B = 2000$ bootstrap replications.

4.1. 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*:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}(u^2) - \nu\frac{\partial^2 u}{\partial x^2} = \psi \tag{25}$$

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 $u_0 \in H^1(]0, 1[)$:

$$u(t = 0, x) = u_0(x) \quad \forall x \in [0; 1] \tag{26}$$

and boundary values $b_0, b_1 \in C^0([0, T])$:

$$\begin{cases} u(t, x = 0) = b_0(t) \\ u(t, x = 1) = b_1(t) \end{cases} \qquad \forall t \in [0; T] \tag{27}$$

Where b_0, b_1 and u_0 are given functions, supposed to satisfy *compatibility conditions*:

$$u_0(0) = b_0(0) \quad \text{and} \quad u_0(1) = b_1(0) \tag{28}$$

The initial u_0 and boundary values b_0 and b_1 are parametrized the following way:

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

$$f(t,x) = f_m + \sum_{l=1}^{n_T(f)} \sum_{p=1}^{n_S(f)} A_{lp}^f \sin(\omega_l^{fT} t) \sin(\omega_p^{fS} x) \qquad u_0(x) = (u_{0m})^2 + \sum_{l=1}^{n(u_0)} A_l^{u_0} \sin(\omega_l^{u_0} x)$$

The values of the angular frequencies $\omega_l^{b_0}$, $\omega_l^{b_1}$, ω_l^{fT}, ω_p^{fS} and $\omega_l^{u_0}$, as well as their cardinalities $n(b_0)$, $n(b_1)$, $n_T(f)$, $n_S(f)$ and $n(u_0)$ are fixed (known), while our uncertain parameters, namely: viscosity ν, coefficients b_{0m}, b_{1m}, f_m and u_{0m}, and amplitudes $\left(A_l^{b_0}\right)_{l=1,\dots,n(b_0)}$, $\left(A_l^{b_1}\right)_{l=1,\dots,n(b_1)}$, $\left(A_{lp}^f\right)_{l=1,\dots,n_T(f);p=1,\dots,n_S(f)}$ and $(A_l^{u_0})_{l=1,\dots,n(u_0)}$ live in some Cartesian product of intervals $\mathcal{P}'$, subset of $\mathbf{R}^{1+4+n(b_0)+n(b_1)+n_T(f)n_S(f)+n(u_0)}$.

However, the compatibility condition (28) constraints b_{0m} and b_{1m} as functions of the other parameters:

$$b_{0m} = (u_{0m})^2 \quad \text{and} \quad b_{1m} = (u_{0m})^2 + \sum_{l=1}^{n(u_0)} A_l^{u_0} \sin(\omega_l^{u_0}) \tag{29}$$

so that the "compliant" uncertain parameters actually belong to $\mathcal{P}$ defined by:

$$\mathcal{P} = \Big\{ \mathbf{X} = \big(\nu, b_{0m}, A_1^{b_0}, \dots, A_{n(b_0)}^{b_0}, b_{1m}, A_1^{b_1}, \dots, A_{n(b_1)}^{b_0}, f_m, A_{11}^f, A_{12}^f, \dots, A_{1,n_S(f)}^f, A_{2,1}^f, \dots, A_{2,n_S(f)}^f, \dots, A_{n_T(f),n_S(f)}^f, u_{0m}, A_1^{u_0}, \dots, A_{n(u_0)}^{u_0}\big) \in \mathcal{P}' \text{ satisfying (29)} \Big\} \tag{30}$$

In Janon et al. (2010, *submitted.*), we gave an example with many more parameters. To illustrate our sensitivity analysis methodology without overloading the text, we choose an example with a reduced number of parameters.

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 (25), (26), (27), 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 (25), (26), (27), as well as an error bound ε_u on u. Due to non-linearity and time-dependence of (25), as well as parametrization of the boundary values, the reduced basis methodology is not as simple as the one presented in Section 1.2 of the present paper. The reader can refer to Janon et al. (2010, *submitted.*) for full details on discretization and reduction of this model. Error bound on output ε is obtained by following (7).

In our numerical experiments, we take $\mathcal{N} = 60$, $\Delta t = .01$, $T = .05$, $n_S(f) = n_T(f) = n(b_0) = n(b_1) = 0$, $n(u_0) = 1$, $\omega_1^{u_0} = 0.5$, $A_1^{u_0} = 5$ and $f_m = 1$.

Reduced basis are found using POD-based procedure with $\#\Xi = 30$.

The two input parameters are assumed independent and uniformly distributed. 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:

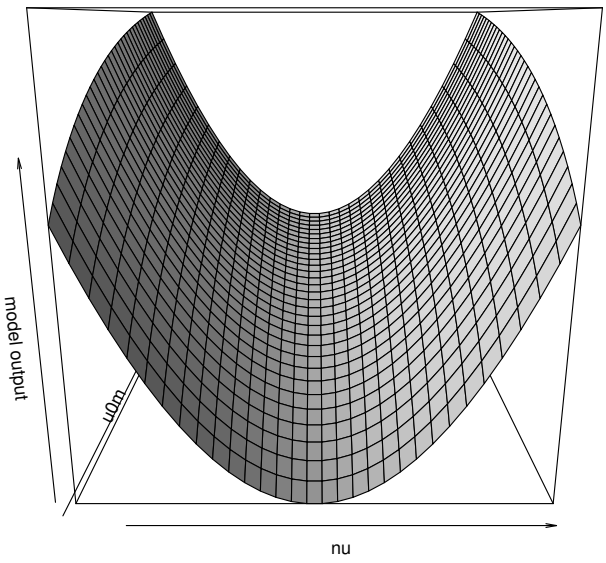

Figure 1: Output f of the Burgers' model, plotted as a function of ν and u_{0m}.

Parameter	Range	95% confidence interval for sensitivity index
ν	[1 ; 20]	[0.0815;0.0832]
u_{0m}	[-0.3 ; 0.3]	[0.9175;0.9182]

The output, as a function of the two uncertain parameters ν and u_{0m} is plotted at Figure 1; as one can see it is nonlinear with respect to the input parameters.

4.2. Convergence benchmark

Figure 2 shows the lower $\widehat{S^m}$ and upper $\widehat{S^M}$ bound (defined in Section 2.2) for different reduced basis sizes (hence different metamodel precision) but fixed sample of size $N = 300$, as well as the bootstrap confidence intervals computed using the procedure presented in Section 3.1. 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 (gaps between confidence interval upper bound and upper bound, and between confidence interval lower bound and lower bound) remains constant, as sample size stays the same.

4.3. Choice of n and N

We now discuss the numerical results obtained when using the parameter tuning procedure (Section 3.2). We have done "pre-runs" for $N = 300$, and different reduced basis sizes $\{7, 8, \ldots, 12\}$ of the combined confidence interval procedure, to yield the following estimates:

$$\widehat{C} = 197.69 \quad \widehat{a} = 2.789 \quad \widehat{Z}_{.05} = 2.6407$$

To assess validity of this estimation, and to check our modelisation of the bound precision and the execution time, we plot the cube root of the CPU time (Figure (3)) and the precision of the bound defined in Section 2.2 (Figure (4)). We can check that these hypotheses are reasonably satisfied.

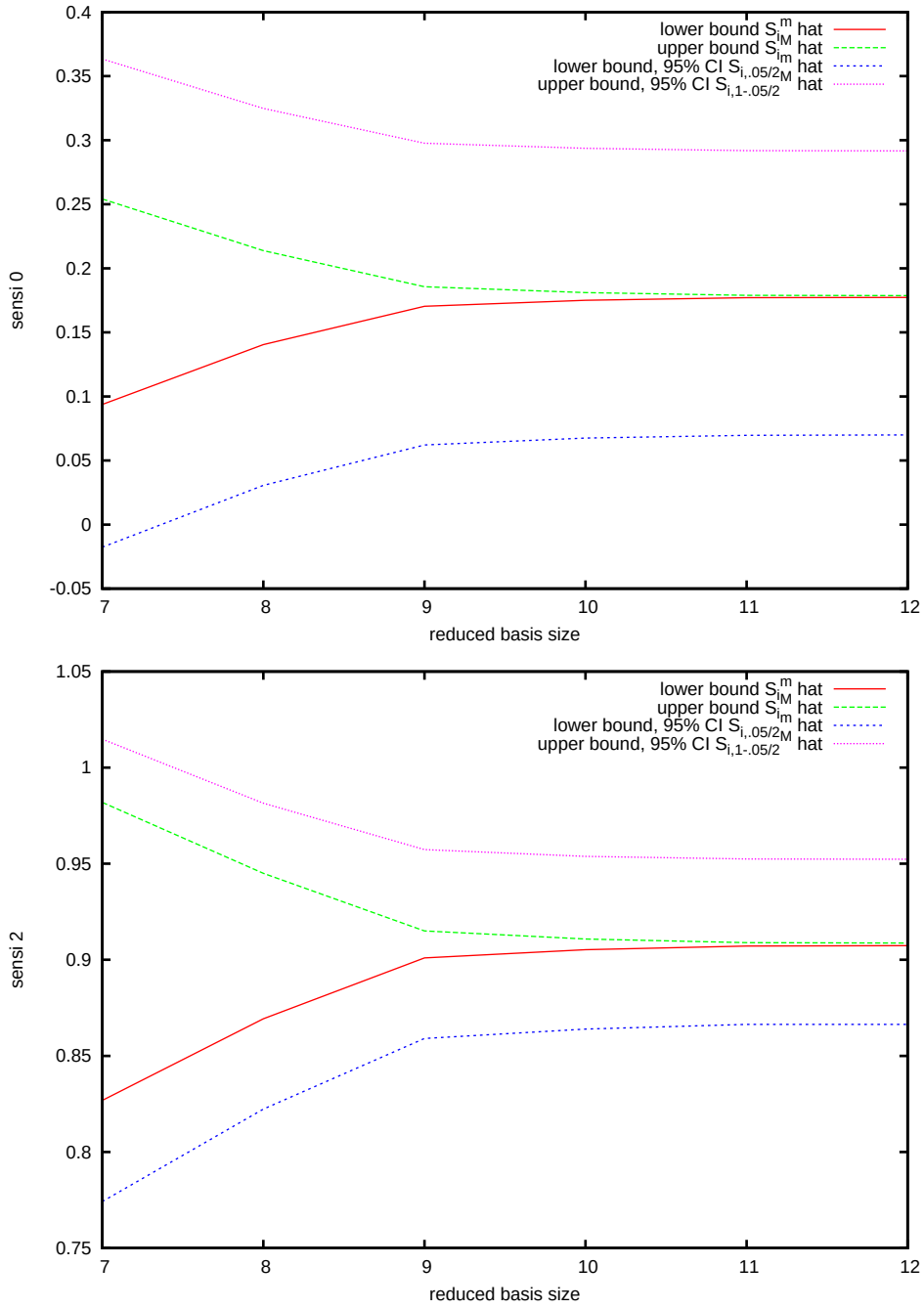

Figure 2: Convergence benchmark for sensitivity indices estimation in the Burgers' model, top: variable ν, bottom: variable u_{0m}. We plotted, for a fixed sample size $N = 300$, estimator bounds $\widehat{S}^m$ and $\widehat{S}^M$ defined in (2.2), and endpoints $\widehat{S}^m_{i,.025}$ and $\widehat{S}^M_{i,1-.025}$ of the 95% confidence interval, for different reduced basis sizes.

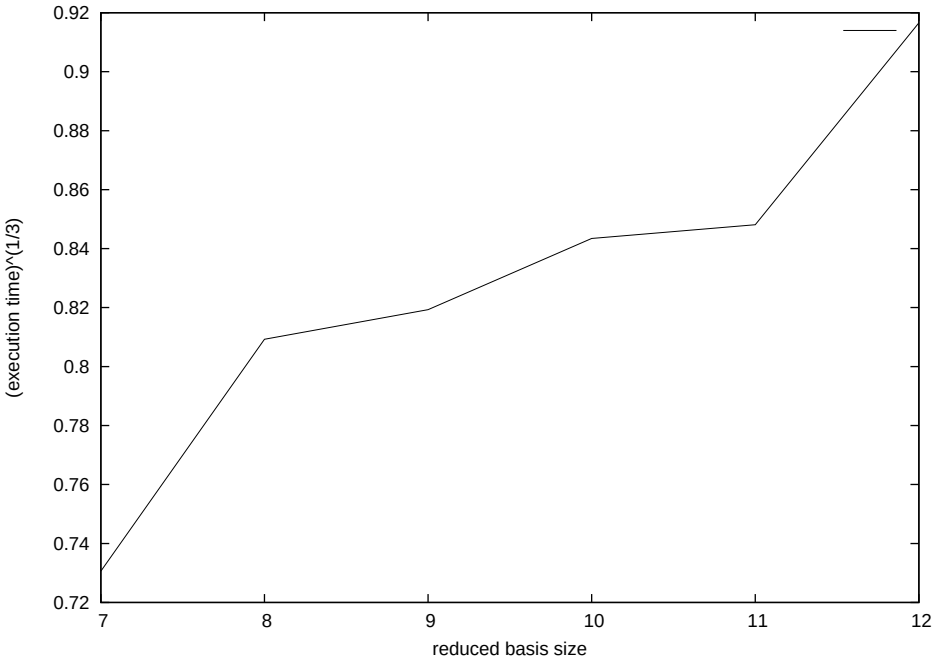

Figure 3: Cube root of the CPU time necessary to do estimations, as a function of the reduced basis size of n. Section 3.2 assumes this function is linear.

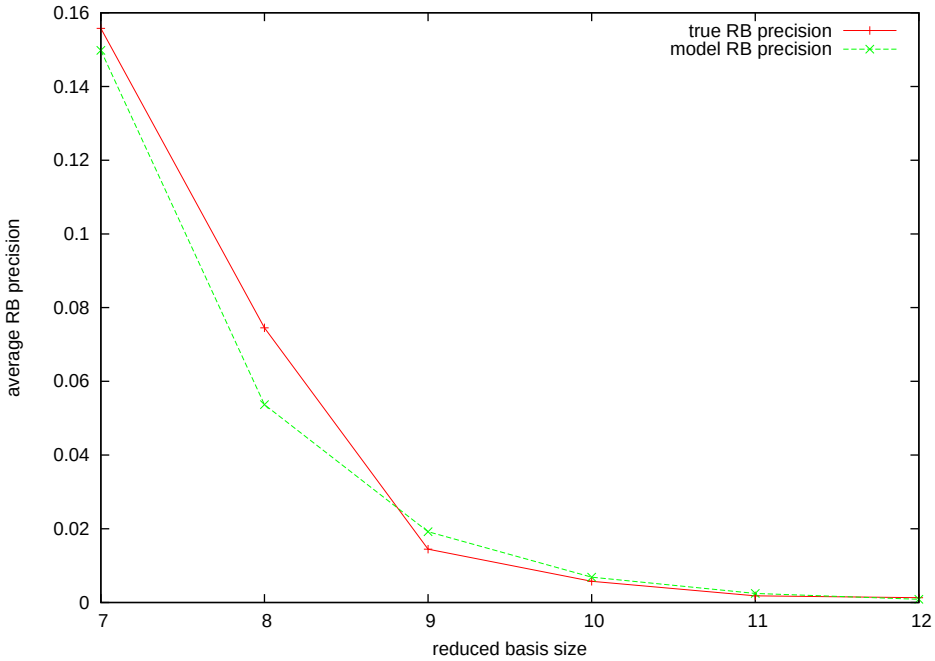

Figure 4: Line: reduced basis "precision", i.e. mean $\widehat{S}_i^M - \widehat{S}_i^m$, as a function of reduced basis size; dashes: result of the fit of an exponential regression model: $\widehat{C}/\widehat{a}^n$.

Precision p	Reduced basis size n^*	Sample size N^*
0.005	12.4437	354491
0.02	11.1095	22057.6
0.05	10.0501	3698.95
0.08	9.59689	1442.7
0.09	9.48332	1139.47

Figure 5: Optimal reduced basis and sample sizes calculated using the strategy described in Section 3.2.

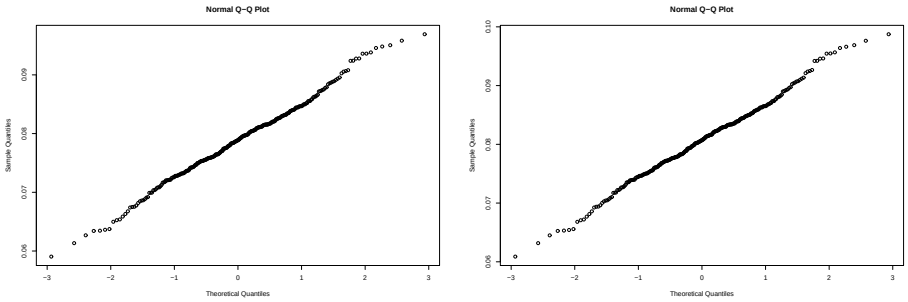

Figure 6: Normal empirical quantile-quantile plots of the distributions of S_ν^m (left) and S_ν^M (right).

One can find in Figure 5 the computed optimal reduced basis sizes n^* and sample sizes N^* using resolution of the optimization problem (21), for various precisions p.
All these values have been computed in 5.77 s CPU time, including the time necessary to estimate C, a and σ.
To check for the efficiency of this parameter tuning strategy, we choose a target precision of $p = .02$. In the table 5, we read that we should take $n \approx 11$ and $N \approx 22000$.
Conducting the combined confidence interval estimation with these parameters give intervals $[0.0659997; 0.0937285]$ for sensitivity index for ν, and $[0.914266; 0.926452]$ for sensitivity with respect to u_{0m}. These confidence intervals have mean length:

$$\frac{1}{2}\left(0.0937285 - 0.0659997 + 0.926452 - 0.914266\right) = 0.0199575 \approx 0.02$$

as desired.
This computation took 52 s of CPU time to complete (including a metamodel offline phase of 1 s).

4.4. Normality of the bootstrap distributions

We give in Figure 6 the empirical normal quantile-quantile plots of the bootstrap replicates $\{\widehat{S}_i^m[1], \ldots, \widehat{S}_i^m[B]\}$ and $\{\widehat{S}_i^M[1], \ldots, \widehat{S}_i^M[B]\}$.
As these plots are close to a line, the bootstrap distributions are approximately normal.

4.5. Optimality of our metamodel error bound

We checked for near optimality of the metamodel error bound 2.2 by comparing it with the values of the optimization problems: $\min_{\mathcal{Y}} \psi$ and $\max_{\mathcal{Y}} \psi$ where:

$$\psi(\mathbf{y}) = \frac{\frac{1}{N}\sum_{k=1}^N y_k y_k' - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)\left(\frac{1}{N}\sum_{k=1}^N y_k'\right)}{\frac{1}{N}\sum_{k=1}^N (y_k)^2 - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)^2}$$

Surface response (metamodel)	Mean confidence interval length	R^2	CPU time
qreg: quadratic regression	0.081	0.996799	143.59 s
mars: multivariate adaptive regression splines	0.075	0.9998506	218.716 s
our approach	0.019	N/A	52 s

Figure 7: Results of CompModSA's sensitivity function on our model, for two fitted response surfaces. R^2 is an indicator of the metamodel fit (values close to unity suggest good fit). The last line recalls the results of the experiment in Section 4.3.

and:

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

These problems, of large dimension $2N$, give the optimal values of $\widehat{S}_i^m$ and $\widehat{S}_i^M$ satisfying (10). They can be solved with simulated annealing Kirkpatrick et al. (1983); Pardalos and Romeijn (2002). Our bound gave results very close to the optimal ones, for a smaller computational cost than using simulated annealing.

4.6. Comparison with estimation on the full model

To obtain a result of the same precision, we carry a simulation for $N = 21000$ (sample size can be chosen smaller than before, as there will be no metamodel error) on the *full* model; we get the bootstrap confidence interval with mean length of ≈ 0.0193.
This computation takes 308 s of CPU time to complete. Hence, on this example, using a reduced-basis surrogate model roughly divides overall computation time by a factor of 5.9, without any sacrifice on the precision and the rigorousness (as our metamodel error quantification procedure is fully proven and certified) of the confidence interval. We expect higher time savings with more complex (for example, two- or three-dimensional in space) models.

4.7. Comparison with CompModSA

We compared our results with the ones obtained using the R *CompModSA* version 1.2 package downloaded at compmodsa. This package implements the method described in Storlie et al. (2009) for assessing sampling error as well as metamodel error. It does not make use of the reduced basis output error bound, but uses a non-intrusive method to fit a metamodel using a reasonable number of full model evaluations.
We fed into CompModSA procedure 50 such full model outputs (which took 0.22 s CPU to compute). We then tried various non-intrusive metamodels (*surface responses*), and reported the results into Figure 7. We used as parameters: n.mc.T=0 (we do not want any total index computation), n.mc.S=23000 (sample size), n.samples=1 (one run), n.CI=300 (generate confidence intervals using 300 bootstrap replications). We contributed [2] the option CI.S, which is set to TRUE to compute bootstrap confidence intervals for the main effect index.
By looking at results in Figure 5, we can see that in this case, our approach is clearly superior, both in terms of precision and computation time. To achieve this result, we took advantage of the particular formulation of the original model which allows the reduced basis methodology to be efficiently applied; CompModSA, due to its non-intrusive nature, is easier to use on a generic "black box" model.

[2] patch available at `http://ljk.imag.fr/membres/Alexandre.Janon/compmodsa.php`

Conclusion and perspectives

We presented a methodology to make a rigorous quantification of the impact of the sampling error and the metamodel error on the sensitivity indices computation, when a reduced-basis metamodel is used. Sampling error is handled by a classic bootstrap procedure, while metamodel error is managed using a new bound on the sensitivity index estimator. Quantification of those two types of errors permits not only a certification on the performed estimation, but also gives a way to tune the optimal parameters (reduced basis and optimal sample sizes), for a given desired precision on the indices. We have shown on a concrete example the superiority of this method when compared to the use of the full model, or non-intrusive (quadratic regression, MARS) metamodels. Our method can be applied to other Monte Carlo (or quasi Monte Carlo) estimators, and to other metamodels which provide an error bound similar to the one provided by the reduced basis framework.

Acknowledgements. We wish to thank Jean-Claude Fort for a suggestion which we exploited to perform our computation of the metamodel-induced error. We also thank Anestis Antoniadis and Ingrid Van Keilegom for advice on bootstrap methodology. – This work has been partially supported by the French National Research Agency (ANR) through COSINUS program (project COSTA-BRAVA n°ANR-09-COSI-015).

A. Algorithms

Algorithm 1:

1. Draw from $\mathbf{X}$ distribution two independent samples of size N: $\{\mathbf{X}^k\}$ and $\{\mathbf{X}'^k\}$.
2. Tabulate necessary model evaluations: for $k = 1, \ldots, N$:
   (a) set $\mathbf{X} \leftarrow \mathbf{X}^k$;
   (b) compute $y_k = f(\mathbf{X})$;
   (c) swap X_i and $X_i'^k$;
   (d) compute $y'_k = f(\mathbf{X})$;
   (e) swap X_i and $X_i'^k$ back.
3. Compute $\widehat{S}_i$:
$$\widehat{S}_i = \frac{\frac{1}{N}\sum_{k=1}^N y_k y'_k - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)\left(\frac{1}{N}\sum_{k=1}^N y'_k\right)}{\frac{1}{N}\sum_{k=1}^N y_k^2 - \left(\frac{1}{N}\sum_{k=1}^N y_k\right)^2}$$
4. Repeat, for $b = 1, \ldots, B$:
   (a) Draw at random a list L of length N, with replacement from $\{1, \ldots, N\}$.
   (b) Compute replication $\widehat{S}_i[b]$:
$$\widehat{S}_i[b] = \frac{\frac{1}{N}\sum_{k\in L} y_k y'_k - \left(\frac{1}{N}\sum_{k\in L} y_k\right)\left(\frac{1}{N}\sum_{k\in L} y'_k\right)}{\frac{1}{N}\sum_{k\in L} y_k^2 - \left(\frac{1}{N}\sum_{k\in L} y_k\right)^2}$$
5. Compute $\widehat{z}_0$:
$$\widehat{z_0} = \Phi^{-1}\left(\frac{\#\{b \in \{1, \ldots, B\} \text{ s.t. } \widehat{S}_i[b] \leq \widehat{S}_i\}}{B}\right)$$
where $\Phi(z) = \dfrac{1}{\sqrt{2\pi}}\displaystyle\int_{-\infty}^{z} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t.$

6. Look up for $z_{\alpha/2}$ so that:
$$\Phi(z_{\alpha/2}) = \alpha/2$$
and take $z_{1-\alpha/2} = -z_{\alpha/2}$, satisfying: $\Phi(z_{1-\alpha/2}) = 1 - \alpha/2$.
7. Compute $\widehat{q}(\alpha/2)$ and $\widehat{q}(1-\alpha/2)$:
$$\widehat{q}(\alpha/2) = \Phi(2\widehat{z_0} + z_{\alpha/2}), \qquad \widehat{q}(1-\alpha/2) = \Phi(2\widehat{z_0} + z_{1-\alpha/2})$$
8. Compute $\widehat{S}_{i,\alpha/2}$ and $\widehat{S}_{i,1-\alpha/2}$, the $\widehat{q}(\alpha/2)$ and $\widehat{q}(1-\alpha/2)$ quantiles of $\{S_i[1], \ldots, S_i[B]\}$.
9. Output $\left[\widehat{S}_{i,\alpha/2}; \widehat{S}_{i,1-\alpha/2}\right]$ as confidence interval for S_i of level $1-\alpha$.

Algorithm 2:

1. Draw from $\mathbf{X}$ distribution two independent samples of size N: $\{\mathbf{X}^k\}$ and $\{\mathbf{X}'^k\}$.
2. Tabulate necessary model evaluations: for $k = 1, \ldots, N$:
   (a) set $\mathbf{X} \leftarrow \mathbf{X}^k$;
   (b) compute $\widetilde{y}_k = \widetilde{f}(\mathbf{X})$ and $\varepsilon_k = \varepsilon(\mathbf{X})$;
   (c) swap X_i and $X_i'^k$;
   (d) compute $\widetilde{y}'_k = \widetilde{f}(\mathbf{X})$ and $\varepsilon'_k = \varepsilon(\mathbf{X})$;
   (e) swap X_i and $X_i'^k$ back.
3. Compute $\overline{\widetilde{y}}, \overline{\widetilde{y}'}, \overline{\varepsilon}$ and $\overline{\varepsilon'}$, the respective means of $\{\widetilde{y}_k\}, \{\widetilde{y}'_k\}, \{\varepsilon_k\}$ and $\{\varepsilon'_k\}$.
4. Choose a finite subset Ξ of the set $\overline{\mathcal{P}} \times \overline{\mathcal{P}'}$ where $\overline{\mathcal{P}}$ and $\overline{\mathcal{P}'}$ are defined by (18) and (19).
5. Repeat, for $(\mu, \mu') \in \Xi$:
   (a) By using (11) and (12), compute $R_{inf}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu')$ and $R_{sup}(a; \widetilde{\mathbf{y}}, \varepsilon, \mu, \mu')$ for three different values of a;
   (b) deduce $\alpha_{inf}, \beta_{inf}, \gamma_{inf}, \alpha_{sup}, \beta_{sup}$ and γ_{sup} satisfying (14) and (15);
   (c) if $\alpha_{inf} \leq 0$, exit with failure;
   (d) compute $\widehat{\delta} = 2\sqrt{(\alpha_{inf} - \alpha_{sup})(\gamma_{inf} - \gamma_{sup})}$;
   (e) compute:
$$\widehat{S}_i^m(\mu, \mu') = -\frac{\beta_{inf} + \widehat{\delta}}{2\alpha_{inf}} \qquad \widehat{S}_i^M(\mu, \mu') = -\frac{\beta_{sup} - \widehat{\delta}}{2\alpha_{sup}}$$
6. Output:
$$\widehat{S}_i^m = \min_{(\mu,\mu') \in \Xi} \widehat{S}_i^m(\mu, \mu') \qquad \widehat{S}_i^M = \max_{(\mu,\mu') \in \Xi} \widehat{S}_i^M(\mu, \mu')$$

Algorithm 3:

1. Follow steps 1. and 2. of Algorithm 2.
2. Compute bounds $\widehat{S}_i^m$ and $\widehat{S}_i^M$ using steps 3.–6. of Algorithm 2.
3. Repeat, for $b = 1, \ldots, B$:
   (a) Draw at random a list L of length N, with replacement from $\{1, \ldots, N\}$.
   (b) If using ε-sampling: for $k \in L$, sample ε_k^* uniformly in $[\eta_k \varepsilon_k; \varepsilon_k]$ and ε'^*_k uniformly in $[\eta'_k \varepsilon'_k; \varepsilon'_k]$.

(c) Else: take, for $k \in L$, $\varepsilon_k^* = \varepsilon_k$ and $\varepsilon'^*_k = \varepsilon'_k$.

(d) Compute bounds $\widehat{S}_i^m[b]$ and $\widehat{S}_i^M[b]$ using steps 3.–6. of Algorithm 2, with $(\widetilde{y}_k)_{k\in L}$ instead of $(\widetilde{y}_k)_{k=1,\dots,N}$, $(\widetilde{y}'_k)_{k\in L}$ instead of $(\widetilde{y}'_k)_{k=1,\dots,N}$ as sample data, and $(\varepsilon_k^*)_{k\in L}$ instead of $(\varepsilon_k)_{k=1,\dots,N}$, and $\left(\varepsilon'^*_k\right)_{k\in L}$ instead of $(\varepsilon'_k)_{k=1,\dots,N}$ as error bounds.

4. Compute, for $w \in \{m, M\}$, the two bias correction constants:

$$\widehat{z_0^w} = \Phi^{-1}\left(\frac{\#\{b \in \{1,\dots,B\} \text{ s.t. } \widehat{S}_i^w[b] \leq \widehat{S}_i^w\}}{B}\right)$$

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

5. Look up for $z_{\alpha/2}$ so that:

$$\Phi(z_{\alpha/2}) = \alpha/2$$

and take $z_{1-\alpha/2} = -z_{\alpha/2}$, satisfying: $\Phi(z_{1-\alpha/2}) = 1 - \alpha/2$.

6. Compute $\widehat{q}^m(\alpha/2)$ and $\widehat{q}^M(1-\alpha/2)$:

$$\widehat{q}^m(\alpha/2) = \Phi(2\widehat{z_0^m} + z_{\alpha/2}), \qquad \widehat{q}^M(1-\alpha/2) = \Phi(2\widehat{z_0^M} + z_{1-\alpha/2})$$

7. Compute $\widehat{S}^m_{i,\alpha/2}$ and $\widehat{S}^M_{i,1-\alpha/2}$, the $\widehat{q}^m(\alpha/2)$ and $\widehat{q}^M(1-\alpha/2)$ quantiles of $\{S_i^m[1],\dots,S_i^m[B]\}$ and $\{S_i^M[1],\dots,S_i^M[B]\}$, respectively.

8. Output $\left[\widehat{S}^m_{i,\alpha/2}; \widehat{S}^M_{i,1-\alpha/2}\right]$ as combined confidence interval for S_i of level $1-\alpha$.

References

Archer, G., Saltelli, A., Sobol, I., 1997. Sensitivity measures, ANOVA-like techniques and the use of bootstrap. Journal of Statistical Computation and Simulation 58, 99–120.

Bergmann, J., Iollo, A., 2008. Numerical methods for low-order modeling of fluid flows based on POD .

Boyaval, S., Bris, C., Maday, Y., Nguyen, N., Patera, A., 2009. 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, 3187–3206.

Buffa, A., Maday, Y., Patera, A., Prud'homme, C., G., T., 2009. A priori convergence of the greedy algorithm for the parametrized reduced basis. Mathematical Modelling and Numerical Analysis .

Bui-Thanh, T., Willcox, K., Ghattas, O., van Bloemen Waanders, B., 2007. Goal-oriented, model-constrained optimization for reduction of large-scale systems. Journal of Computational Physics 224, 880–896.

Chatterjee, A., 2000. An introduction to the proper orthogonal decomposition. Current Science 78, 808–817.

compmodsa, . CompModSA. http://www.stat.unm.edu/~storlie/CompModSA/.

Efron, B., 1981. Nonparametric standard errors and confidence intervals. Canadian Journal of Statistics 9, 139–158.

Efron, B., Tibshirani, R., 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical science 1, 54–75.

Efron, B., Tibshirani, R., Tibshirani, R., 1993. An introduction to the bootstrap. Chapman & Hall/CRC.

Grepl, M., Maday, Y., Nguyen, N., Patera, A., 2007. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. Mathematical Modelling and Numerical Analysis 41, 575–605.

Grepl, M., Patera, A., 2005. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. Mathematical Modelling and Numerical Analysis 39, 157–181.

Helton, J., Johnson, J., Sallaberry, C., Storlie, C., 2006. Survey of sampling-based methods for uncertainty and sensitivity analysis. Reliability Engineering & System Safety 91, 1175–1209.

Homma, T., Saltelli, A., 1996. Importance measures in global sensitivity analysis of nonlinear models. Reliability Engineering & System Safety 52, 1–17.

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

Kirkpatrick, S., Gelatt Jr, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220, 671.

Marrel, A., Iooss, B., Laurent, B., Roustant, O., 2009. Calculations of sobol indices for the gaussian process metamodel. Reliability Engineering & System Safety 94, 742–751.

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

Pardalos, P., Romeijn, H., 2002. Handbook of global optimization. Volume 2. Kluwer.

Press, W., Flannery, B., Teukolsky, S., Vetterling, W., 1992. Numerical recipes in C: the art of scientific programming. Cambridge U. Press, Cambridge, England .

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

Saltelli, A., Chan, K., Scott, E., . Sensitivity analysis. 2000.

Sobol, I., 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation 55, 271–280.

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

Veroy, K., Patera, A., 2005. 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, 773–788.